UPPSALA
UNIVERSITET

UPTEC STS 19016
Examensarbete 30 hp
Maj 2019

Do people actually listen to ads in podcasts?

A study about how machine learning can be used to gain insight in listening behaviour

Madeleine Angergård
Sara Hane

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Do people actually listen to ads in podcasts?

Madeleine Angergård, Sara Hane

Today, listening to podcasts is a common way of consuming media and it has been proven that listeners are much more recipient to advertisement when being addressed in a podcast, rather than through radio. This study has been performed at Acast, an audio-on-demand and podcast platform that hosts, monetizes, and distributes podcasts globally. With the use of machine learning, the goal of this study has been to obtain a credible estimate of how listeners outside the application tend to respond when exposed to ads in podcasts. The study includes a number of different machine learning models, such as Random Forest, Logistic Regression, Neural Networks and kNN. It was shown that machine learning could be applied to obtain a credible estimate of how ads are received outside the Acast application, based on data collected from the application. Additionally, out of the models included in the study, Random Forest was proven being the best performing model for this problem. Please note that the results presented in the report are based on a mix of real and simulated data.

Handledare: Ioana Havsfrid
Ämnesgranskare: Michael Ashcroft
Examinator: Elísabet Andrésdóttir
ISSN: 1650-8319, UPTEC STS 19016

1 Populärvetenskaplig Sammanfattning

Podcasts är ett relativt nytt och populärt mediaformat vilket har öppnat upp för en ny marknadsföringskanal. Acast, ett svenskt bolag grundat 2014, var en av de första aktörerna på podcastmarknaden som såg möjligheten för att sälja annonsplatser i podcasts. Acast är idag en av de dominerande aktörerna inom deras område både i Sverige och globalt. Det har visat sig att podcastformatet lämpar sig bra för marknadsföring då den generella uppfattningen är att det är ett intimt och personligt lyssnarformat vilket gör lyssnare mer mottagliga. Undersökningar har visat på att lyssnare är två gånger mer positivt inställda till reklam i en podcast, jämfört med reklam som levereras i radio.

Acast främsta intäktskälla är försäljning av reklam i podcasts men utöver det erbjuder Acast även en mobilapplikation där de tillhandahåller en podcastspelare. Endast 10% av Acasts totala lyssningar sker via deras egna applikation medan de resterande sker genom ett flertal andra plattformar. Problematiken i nuläget är att Acast endast har möjlighet att samla in data rörande lyssnarbeteendet på de lyssningar som sker inom appen.

Inom de flesta plattformarna finns det en funktion som tillåter lyssnare att skippa reklam genom att spola fram i intervaller om 15 sekunder. Samma funktion började erbjudas inom Acasts applikation under första kvartalet av 2019 och i samband med att funktionen lanserades började Acast även samla in data på hur långt in i en reklam lyssningar skedde, i form av kvartiler.

Inom reklambranschen så debiteras oftast de som köper annonsplatser för antalet intryck som genereras. Problematiken i att endast 10% av lyssningarna sker inom Acast egen applikation är att de inte kan samla data över hur en lyssning ser ut utanför appen och är därför inte medvetna om till vilken grad en reklam konsumeras. Med tanke på att Acast debiterar företag baserat på faktiska annonsintryck men bara har information gällande detta för en minoritet av de totala lyssningarna så skulle en modell som kan bidra med sådan insyn för resterande del av lyssningarna vara av stort värde.

Syftet den denna studie har varit att ta fram en modell som förutspår lyssnarbeteendet inom reklam för utomstående lyssningar med hjälp av data samlat via Acasts egen applikation. Problemformuleringen för studien var huruvida det går att ta fram en trovärdig uppskattning om hur utomstående lyssnare svarar på reklam inom podcasts med hjälp av maskininlärning. Studien visade på att Random Forest var den modell som genererade klassificeringar av data med högst precision samt att det går att applicera maskininlärning till problemet med ett gynnsamt utfall. Observera att resultaten presenterade i rapporten är baserade på en mix av riktig samt simulerad data.

2 Acknowledgements

This study was performed during the time period between November 2018 and May 2019. It was the master thesis project of the programme in Sociotechnical Systems Engineering at Uppsala University. The study was carried out as a collaboration with the Swedish company Acast.

We wish to acknowledge our supervisor at Uppsala University, Michael Ashcroft, for his valuable knowledge and guidance throughout the project. Beyond this acknowledgement we also would like to direct a thank you to everyone at Acast who has been of assistance during our time there. Specifically, we would like to thank our supervisor at Acast, Ioana Havsfrid, for supporting us.

Madeleine Angergård and Sara Hane
Uppsala, May 2019

Contents

3 Introduction

Today, listening to podcasts is one of the most common ways of consuming media. In recent studies it has been proven that listeners are much more recipient to advertisement when being addressed in a podcast, rather than through radio. Advertisement in podcast was shown to be twice as efficient than in radio, which indicates the potential in podcast as a marketing channel. [53]

Acast, the company at which the study has been performed, was one of the first actors seeing the potential in podcast marketing. Acasts main revenue stream comes from selling advertisement space in their monetized and hosted podcasts. Acast is a curated platform for podcasts, offering a full-service solution connecting podcast content creators with advertisers and listeners. One of their products apart from selling ad space, is their own podcast player, called the Acast app. Only about 10 % of the listens in Sweden are done through the app, but the advantage of hosting their own player is that listening behaviour can be tracked and valuable data can be collected. For the remaining part, 90 % of the listens, that are mainly done through other platforms, a vague estimation of the listening behaviour within ads is done.

With the use of machine learning our goal has been to obtain a credible estimate of how listeners through RSS tend to respond when exposed to ads in podcasts. This has been studied using data collected through the Acast app and the study includes a number of different machine learning models, such as random forest, logistic regression, neural networks and kNN. Please note that the results presented in the report are based on a mix of real and simulated data.

4 Framing of Question

The purpose of this master thesis project has been to utilize machine learning to gain insight in what the ad listening behaviour in podcasts looks like. It has implied investigating and comparing a number of machine learning models in order to obtain a credible estimate.

4.1 Problem Formulation

Can machine learning be applied to obtain a credible estimate of how listeners through RSS tend to respond when exposed to ads in podcasts, using data from the Acast app?

4.2 Delimitation

The study performed in this master thesis project was limited to only include podcasts created in Sweden and with Swedish as main language. Additionally, only podcast monetized or hosted by Acast i.e. podcast containing ads, has been included in the study. The downloading of podcasts can be executed in three different ways; via RSS, within the Acast app or through embed listens. Only downloads done via RSS and in the Acast app have been included in the study. Additionally, there are several types of ads sold by Acast. The most common ones are airtime ads and sponsorship ads, hence only these two were included in the study. In the beginning of the study, new data was requested and tracked in order to carry out the investigation. The required data was not available until February 2019, hence the study is limited to only include data from then and forward.

5 Glossary

- Content creators: The people creating the content and talking in a podcast.
- Show: A podcast channel
- Podcast: An episode of a podcast
- Ads: Advertisement. Can either be an integrated ad or an airtime ad.
- Integrated ads: Also referred to as sponsorships where the content creators speaks warmly of a brand, not a distinct ad.
- Airtime ads: Dynamically inserted ads that comes directly from the advertiser. The content creators are not involved or mentioned.
- Impression: An impression is a term used in marketing and can be compared to when an ad is viewed, or in this case listened to. It is a metric used to measure how many people that are exposed to a certain ad.
- CPI: Cost per impression
- CPM: Cost per mille or cost per thousand impressions
- On boarding: launching of a new podcast at Acast, assigning category and sales verticals, done before monetizing.
- Downloaded podcasts: A download of a full podcast, avaibale to be played "offline" at a later date and time.
- Live stitching: Adding ads to the podcast content file upon a request being done. Resulting in an audio file, referred to as a podcast.
- Hosted shows: The shows created by Acast content creates and includes ads sold by Acast.
- Monetized shows: The shows not created by Acast content creators, but where Acast provides ad spots in the shows

6 Background

6.1 Acast

Acast is, as mentioned before, an audio-on-demand and podcast platform that hosts, monetizes, and distributes audio content globally. As a full-service solution they connect podcast content creators with advertisers and listeners. [1]

Acast was founded in 2014 and today the company has 115 employees (February, 2018) in total. In 2017, Acast launched their fourth market in Australia, after already having access to markets in Europe, UK and the US. Acast has its headquarter in Stockholm, and has offices in London, New York City, Los Angeles, and Sydney. [2]

The revenue stream at Acast depends mainly on the ad sales. This fact makes the metrics and the knowledge about number of listens, and especially the number of impressions, of great importance in order to charge their customers correctly which indicates the importance of this study.

6.2 Podcasts

A podcast is defined as a digital audio file made available online for downloading. The download can be done either to a computer, a mobile device or any hardware with access to internet connection and with capability of reproducing audio, such as home devices, cars and more. A podcast is usually an episode as part of a series which a show publishes continuously with a certain time intensity. Shows can be subscribed to in order for listeners to be notified when a new podcast is published. The term podcast origins from the device "iPod" and the word "broadcast". [42]

A podcast can apart from the content created by the content creators also include advertisement. The podcasts used in this research and referred to in this report, will all be containing ads. The structure of how a podcast is composed at Acast is presented in Figure 1. There are three ad sections that can consist of a combination of different types of ads. The three ad sections are referred to as pre-, mid- and post-roll, and are distributed as presented in Figure 1. [35]

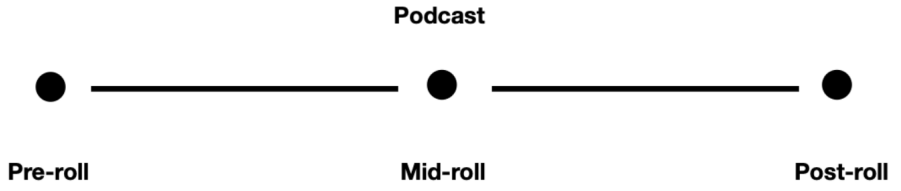

Figure 1: Podcast Model

6.3 Ad Types & Ad Quartiles

There are several types of ads that can be included in an ad section. The two dominating ad types and the ones included in our study are described below.

- Airtime Ads: Dynamically inserted ads that comes directly from the advertiser. The content creators are not involved in the creation of the ads or mentioned in the ad.
- Integrated Ads: These type of ads are also referred to as sponsorships. What characterizes these ads is that the content creators produce the ads and it involves the content creators speaking warmly of a brand.

All ads consists of four ad quartiles, as can be seen in Figure 2. Listening to 25% of an ad means that the first quartile is finished and an ad is listened to until completion at 100 %. How many quartiles a listener has reached is of interest for the advertisers as this indicates to what degree an ad impression is gained.

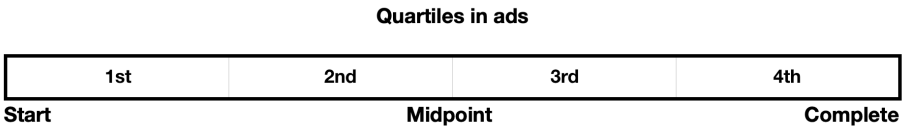

Figure 2: Ad Quartiles

6.4 Podcast Listeners & Listening Behaviour

In surveys done during 2017 it became evident that Acasts strongest listening segment was among people between ages 18-45 years, which represented 70% of the total amount of listeners. It was also shown that 55% were men and 45% were women and that a common attribute among listeners was that they were often highly educated city dwellers, and 35% of the audience had an income of more than 35.000 SEK/month. Additionally, it was concluded that most listeners could be described as unattached, curious and strong-willed people. Podcasts are commonly consumed when alone, on route and mostly during non-weekend mornings. 88% of the podcast requests, and hence the listens, comes from mobile devices.[52]

The distribution of listens varies depending on seasons, day of the week and time of the day. Figure 3 4 and 5 are based on the number of downloads during 2018. By studying Figure 3, presenting the monthly listening behaviour, it becomes evident that there are peaks during spring and fall and downs during the middle of the summer and the winter season. Figure 4, shows that the number of listens decrease during weekend. Figure 5 presents how the listening varies throughout the day and there seems to be a peak during commuting rush hours and before bed-time.

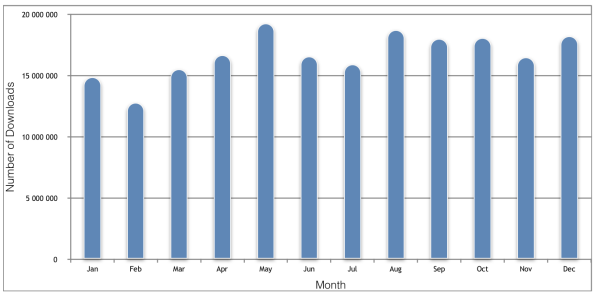

Figure 3: Monthly Listening Behaviour

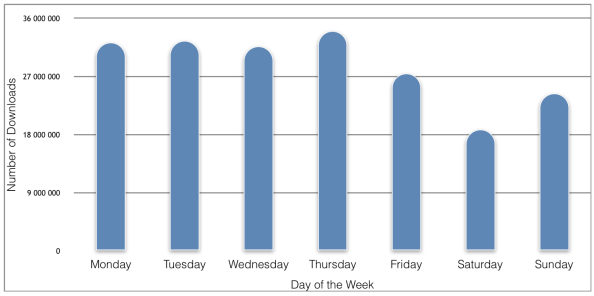

Figure 4: Weekly Listening Behaviour

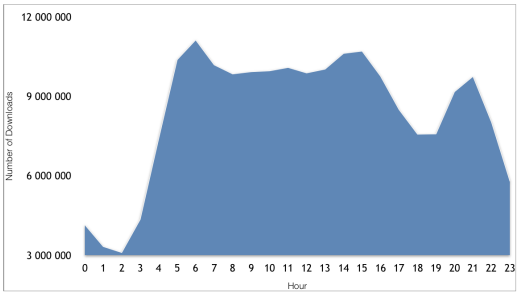

Figure 5: Daily Listening Behaviour

6.5 Advertising in Podcasts

Due to the characteristics of the majority of podcast listeners, the audience can be seen as a good target for advertisers. The intimate feeling of listening to a podcast is shown to be a successful factor as listeners are much more recipient when being addressed directly.[54]

In Table 1, the mean of the attitude towards ads is presented. The mean is based on the answers from listeners to ten different shows monetized by Acast. This shows that the overall attitude seems to be negative. Worth mentioning is that the attitude varies quite vigorously across the different categories. As mentioned earlier, the attitude towards ads in podcast is twice as good as for radio, but can still be seen as over all negative. This attitude makes the situation vulnerable and sensitive to how ads are presented in the podcast. Understanding how listeners behave when encountering ads is therefore of great importance.

Attitude	%
Very Bad	26.96 %
Bad	23.71 %
No Opinion	37.27 %
Good	9.80 %
Very Good	2.77 %

Table 1: Mean of Attitute towards Ads

6.5.1 Components of an Ad

The payment model for ads is based on consumption of ads and impressions gained. At Acast an ad is divided into quartiles and each consumed quartile is reported back to the ad agency in order to get payed for the impressions. Quartile reporting is commonly used in digital media marketing and it helps calculate the engagement rates of the users or as in this case, listeners. [3], [41]

6.5.2 The Ad Currency

The currency in online marketing is cost-per-impression, referred to as CPI. Usually, it is counted in thousands, generating the abbreviated CPM, which stands for cost-per-mille, or cost-per-thousand-impression. The price setting of a CPM is based on who is being targeted in the advertisement and how specific it is. In other words, there is a direct correlation between the specificity of the targeting and the cost of each impression. If targeting an affluent group the return on investment will probably be better. [78], [9]

6.6 Agreements on the Podcast Market

Due to the importance in reporting back the correct number of impressions, the ways of measuring number of impressions has been discussed. On a competitive market it is vital that all actors generate their metrics based on the same grounds and up until agreements were made, the actors struggled with comparing metrics based on different measurements. In order for this study to be carried out, the market agreements will be described and used as foundation for the study. The

correct definitions of metrics is of importance when deciding what events to track and features to be included in the classification model.

Agreements have been produced both domestically, by Poddindex, and globally by IAB.[36], [51] In most ways the two agreements correspond metrics wise, but due to some differences in definitions, both agreements are taken into account by Acast as they compete on markets world wide. A definition of what is classified as a valid listen that both agreements convey is defined as following, "if a podcast is played for at least 60 seconds, this is classified as a listen" and "if a podcast is fully downloaded (100%) it is also classified as a listen". What must be taken into consideration is that a download can be done for immediate, delayed or non-accomplished consumption. If all actors on the podcast market follow the same procedures, stated in the agreements, the market can remain competitive. [36], [51]

6.7 Client Types

Requests for podcasts can be done from different client types, including the Acast app, via an RSS-request or through an embed player. The distribution between the different client types is presented in Figure 6 below.

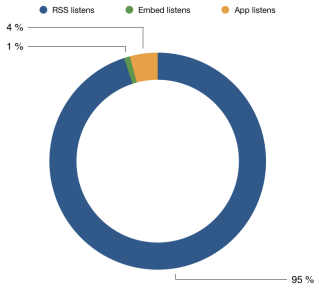

Figure 6: Distribution of listens

6.7.1 Acast App

The Acast app can be utilized as a registered user, a guest user or an Acast+ user. A user can be registered either with their Facebook account, Google account or an email address. Table 2 and Table 3 shows the distribution over the different ways to register in the Acast application, both for iOS and Android. Both tables show that the majority of the users are non-registered users.

- Android

Account type	%
Google	4.00 %
Email	4.06 %

Facebook	4.60 %
None	87.3 %

Table 2: Distribution over user registration for Android

- iOS

Account type	%
Google	5.04 %
Email	6.45 %
Facebook	15.6 %
None	72.9 %

Table 3: Distribution over User Registration for iOS

6.7.2 Embed

Embed listens are done via links to podcasts at websites for instance. Only a minority of the listens are done through this "channel" and as stated in the delimitation section, these listens has been excluded in the study.

6.7.3 RSS

RSS is abbreviation for Rich Site Summary, a web feed which allows users and applications to get access to content available online. As can be seen in Table 6, the majority of the requests are done via RSS. It is a standardized format, XML, allowing anyone who creates web content, such as podcasts, to use it as a method for distribution. Using the tool allows content producers to only having to upload their content to one hosting service, and the distribution is all done through the RSS feed. The majority of the RSS requests are done through apps such as Spotify and Apple Podcaster. [68], [81] Depending on the client type which requests a podcast, the possible information to retain about the user varies.

7 Related Work

As of today, we have not been able to find a similar research project about podcasts. As the podcast business is relatively new and insights in how podcast listeners consume advertisement is of value for competing companies, these kind of results are probably not published but kept in-house if investigated. However, articles regarding general behaviour in skipping ads was found, as well as studies using machine learning to investigate behaviour in media.

7.1 Behavioural Analysis & Machine Learning

Lately, behavioural analysis has become a big part of information technology, something that is frequently investigated using for instance machine learning. Behavioural analysis is of great importance when it comes to recommender systems which are used in streaming services for music, movies and podcasts. The goal is to create customer value by adapting the content and delivering personalized suggestions. On a competitive market, customer satisfaction is crucial in order to avoid churn [1]. The behaviour of certain users and listeners is shown to be effectively studied using machine learning, which suggests that it may be applicable on our study.[65], [72], [30]

7.2 Meta Data

One way of analyzing behaviour, is by using meta data. Meta data is defined as "data about the data" and is said to give information about the data so that it can be used in a more efficient and simple way. Examples of meta data that can be the designation or indexing term of an object, such as "title", "description", "language", "year" etc. [28] In the article "Image Labeling on a Network: Using Social-Network meta data for Image Classification" meta data is harnessed in image labelling and the usefulness of meta data is clearly conveyed. The study which is presented in the article examines what kind of meta data can be used for predicting image labels, tags, and groups of images on Flickr. The data set used for the study included the uploaded photo itself, photo meta data such as time of upload, user information and photo tags. With the 'time of upload' it was possible to determine if several photos were taken by the same user, or from the same location which could be helpful in the labelling. In the study it is also examined what type of meta data that is useful in the labelling and classification, being either a stronger or weaker predictor. [26]

7.3 Predicting Using Neural Networks

In the article "Predicting Movie Success Using Neural Network" a multi-classification problem is presented. The goal was to classify movies into one of the success classes [flop, hit, super hit] based on historical data including movie features and

[1] Churn can be defined as a measure of the number of individuals moving out of a collective group over a specific time period

success outcome of previously released movies. The data set was subjected to a neural network with *Levenberg–Marquardt back propagation* as the learning algorithm. The approach showed to be very accurate in the predictions and could therefore be seen as very efficient. In the article, methods on how to increase the accuracy of the predictions are presented, including data normalization, ratio between train and test set and adjustment of number of network layers. [29], [59]

7.4 Skippable Ads in Online Marketing

Skippable ads is a format commonly used in online marketing and it is often added to media platforms such as online games, videos and podcasts. On YouTube for instance, users are allowed to skip ads after watching a few seconds of the content. In an article published in the *Business Insider*, a study regarding millennials behaviour when encountering online ads in videos is presented. It was shown that 59 % chose to skip ads when possible, while 29 % of the millennials indicated that they watched ads until completion. The advantage of skippable ads is that it is more user friendly, allowing uninterested viewers or listeners to avoid ads. Another advantage is that the advertisers only pay fully for the ads that are actually being viewed. [33] One of the outcomes of skippable ads is that platforms offering the option to skip ads, in general have more users and more advertisers than platforms not allowing such ad format. This, and several more insights on skippable ads are presented in the article "Interactive Advertising: The Case of Skippable Ads". The authors have created a statistical model in order to investigate the efficiency of skippable ads. One conclusion drawn and presented in the article is that the introduction of skippable ads on a platform increases the number of visitors. Second, they present that the skippable ads changes how likely a viewer is to become a consumer and make a purchase, a so called conversion. In case a skippable ad is consumed until completion, the likelihood of conversion is higher, than for traditional ads. [6]

In an investigation done by U.S. Patent in 2013, presented in an article, the different factors affecting the effectiveness of ads are discussed. They mention geographic location, audience, time of download and season as some of the multiple factors that may have an impact on the viewers behaviour. They illustrate the problem in defining how efficient an ad is when viewers are allowed to skip, as well as how problematic it is to charge the advertisers when ads can be skipped. [49]

8 Method

8.1 Defining the Problem

The purpose of this study has been to enable Acast to get insight in the listening behaviour of advertisement in their monetized podcasts. In order to gain such insight, a machine learning approach has been applied. In machine learning, models can be trained into detecting patterns of input data, mapping it into an output variable. The models can later be used to make predictions on new, unseen data. In this study the input variables are represented by date and time of a downloaded podcast, information about ads in podcasts and meta data connected to the podcast of which the ad is presented in. The output of the learning problem studied is how far into an ad, in terms of quartiles, a listener reaches. As described in Section 6.3, there are four quartiles of an ad and therefore, in this study the events of initializing an ad listen and reaching each quartile is represented by a class label each. We therefore have used five classes in our learning problem, described below.

- **Class label = meaning of class label**
- 0 = listening to the ad but not reaching first quartile
- 0.25 = listened to at least the first quartile
- 0.50 = listened to at least the second quartile
- 0.75 = listened to at least the third quartile
- 1= finished the entire ad

For the listens made through the Acast app both the input and output values are available, while for the listens done via RSS, the output values are missing and hence the trained machine learning model is supposed to be used for predicting such values.

8.2 Tools & Libraries

- **Scikit learn:** A free machine learning library compatible with Python used for data mining and data analysis. The library is built on NumPy, SciPy and matplotlib. Used in this study to perform pre-processing of data, prediction and evaluation of models for instance.
- **MySQL Pro for MSSQL:** An SQL server manager compatible with macOS used to extract data from Acast's Azure database referred to as the Warehouse.
- **Keras:** Keras is a high-level API written in Python, used to create and run neural networks. The API is capable of running on top of Tensorflow. It offers a deep learning library enabling the building of neural networks to be simple and easy to adjust.

- **Pandas:**

  Pandas is an open source Python data analysis library. In this study Pandas is used to structure data, using data frames, in order to carry out the data analysis in a simple and efficient way.

- **NumPy:** A package used for scientific computing, compatible with Python. NumPy uses much less memory to store data compared to lists in Python and many vector or matrix operations can be executed more efficiently then when done manually.

- **TensorFlow:**

  An open source library which can be used to develop and train machine learning models. Compatible with Python and works together with Keras.

- **Python:** An interpreted, object-oriented, high-level programming language. Python supports compatibility with many libraries and packages which may be used in order to make programming tasks more efficiently executable.

- **iPython:** iPython is an open source architechture which provides an interactive shell, working as a Python kernel for Jupyter Notebook.

- **Jupyter Notebook:** Jupyter Notebook is an open source web application that can be used to create and share documents containing code for instance. The notebook offers several programming languages including Python which is used in this study.

8.3 Gathering Data

After being given access to the database and being informed of the available tables and its content, a first approach was selecting a number of tables to be considered in the data selection. Thereafter, a more thoroughly feature selection was carried out, which is described in detail in Section 10.

8.4 Data Preparation

In order to use the machine learning process as smooth and efficient as possible, pre-processing of data is carried out. Different approaches are required and suitable depending on what algorithm to expose the data to. The data preparation done in the study is described in detail in Section 10.

8.5 Choosing a Model

Several models has been taken into consideration. After optimizing and tuning the models, they were trained and evaluated in order to find the most suitable model based on a few metrics.

9 Theory

9.1 Statistical Modelling

Machine learning is defined as a set of methods that automatically detects different patterns in data, which enables making predictions on future unseen data. A learning problem is made up by a few components; the input data x, the unknown target function $f : X \longrightarrow Y$, X which is the input space including all possible inputs and Y which is the set of all possible outputs.

There is a data set D of input and output observations $(x_1, y_1)...(x_n, y_n)$ where $y_n = f(x_n)$ for $n = 1...N$. The learning algorithm uses the data set to identify $\hat{f}$ which is an attempt of approximating $f : X \longrightarrow Y$. The algorithm chooses $\hat{f}$ from a set of candidate formulas which is called the hypothesis set H. The algorithm chooses the $\hat{f}$ which best matches f on the training examples of the data set.

Predictions are made using $\hat{f}$, as this is the approximation of the unknown function f. This is done under the assumption that $\hat{f}$ replicates f to an extent which is acceptable.[31], [82]

9.1.1 Types of Learning

The process of learning from data is based on using a set of observations to understand and discover an underlying process and correlation, as mentioned above. There are different types of learning, depending on what information that is available. Below three different types, supervised, unsupervised and reinforcement learning are described. [83]

- Supervised learning is when the training data contains correct outputs for all the given inputs. It is called supervised as the model can be trained under "supervision" due to the knowledge of output labels for the training cases.[83] Given a labeled training data set $D = \{(x_i, y_i)\}_{i=1}^{N}$ the goal is to learn how to predict the output y given the input x. The input x can be a D-dimensional vector of different numbers that represent features or attributes or in the more complex case x can be represented by a image or a graph. The output y can vary a lot but in the majority of the cases y takes on categorical variables or real-valued scalars. An example of when supervised learning is efficiently applied is in classification. [31]

- In unsupervised learning the training data $D = \{(x_i\}_{i=1}^{N}$ does not contain any output values, the only information provided is the input. The goal when performing unsupervised learning is to find patterns in the data that are especially interesting. The lack of output does not mean that we can not learn from the available data, the result may be clusters, just as in supervised learning, but lacking a label for each cluster. One could argue that unsupervised learning is a way of creating a higher level of representation of the data.[83], [31]

- Reinforcement learning is not as commonly used as the two learning types mentioned above. It is a process whereby a control system learns to make decisions that aim to maximize long term expected utility based on environmental feedback.[31]

In this master thesis supervised learning has been used since the output data, regarding reach of quarterlies, was available. Within the field of supervised machine learning there are two types of problems - regression and classification problems, described below.

9.1.2 Types of Problems

Depending on the type of the output variables the problem can be classified as either a classification or a regression problem. A variable can be characterized as either quantitative or qualitative, where quantitative variables include numerical values and qualitative variables include categories or different classes. Categorical variables have at least two different values but can also have several values or categories. These different values or categories can be either nominal or ordinal.

Problems having a quantitative response or output are referred to as regression problems while those having a qualitative response or output are often referred to as classification problems. When deciding what model to use it is common to select model based on the type of the response.[19]

The predicted response for the observation in this study is qualitative and therefore the prediction can be referred to as classifying the observation. When classifying an observation the observation is assigned to a class or a category. Commonly used classifiers are logistic regression, liner discriminant analysis and K-nearest neighbors. Further classification methods that are widely used is random forest, boosting, neural networks and trees.

9.2 Metrics

It is difficult to know what model that will perform the best before trying the models on specific data sets. Therefore it is common to try several models and evaluate them in order to figure out which one suits the current data set the best. Selecting a model can be challenging and using a common metric can help make such a decision. In this section metrics are described that are used in order to measure the performance of a model in terms of correctly made classifications. These metrics common for all models and are not model specific. [20] Other model specific metrics are used during the training of the different models, these will be described briefly when introducing the different algorithms included in the study.

9.2.1 Accuracy

When evaluating a method's performance on a specific data set it is crucial to measure how well the predicted responses matches the observations. In other

words, the number of $\hat{y}$ that corresponds to the actual y in relation to the total number of predictions. One way of doing this is to calculate the error rate, as seen in Equation 1, where $\hat{y}_i$ is the predicted class label of $\hat{f}$ for the *ith* observation.[20]

$$\frac{1}{n}\sum_{i=1}^{n} I(y_i = \hat{y}_i) \tag{1}$$

Here, $I(y_i \neq \hat{y}_i)$ is an indication variable which is equal to 1 if $y_i \neq \hat{y}_i$ and equal to 0 if $y_i = \hat{y}_i$. The error rate will increase with the number of wrongly classified observations. If obtaining an error rate of zero we know that all classified observations are correctly classified.

Equation 1 has been used when calculating the error rate. When applied on the training data it generates the training error rate. i.e the In-sample error. The in-sample error, E_{in}, reviles how well a model performs on the training data. It does not reflect on how well the predictor works in practice and how well it will perform when introduced to new data. Therefore the measure can be interpreted as an optimistic estimation of the actual error.

Using Equation 1 together with the test data, the out-of-sample error, E_{out}, can be calculated and an estimate of the performance of the model can be provided. It reviles how well the training on the training data, has generalized to data which is unknown to the model. It indicates the performance of the predictor in general and it is usually a good estimate of how well the model performs in real life.[20], [83]

9.2.2 Confusion Matrix

One disadvantage when only using the accuracy rate to evaluate the performance of a model is that the accuracy rate does not make any distinction about the different types of errors. For example when predicting if a patient has a certain disease or not, it is more devastating to declare a sick patient as healthy than to falsely declare a healthy patient sick. In such situations, where the cost of errors varies, other rates can be calculated as seen in Equation 2- Equation 9. [43]

For Acast, a miss-classification is not as devastating as in the example stated above. Classifying an ad impression as reached to a certain quartile when not reached in practice, might seem unethical as that impression will be chargeable even though it should not. The opposite, not charging an advertiser for an ad impression, even though it has been delivered, is from Acasts perspective bad for the revenue stream. Both cases are undesirable, and affects different actors, but it is hard to decide which outcome would have the worst effect. One could argue that in this case, we can assume that the errors are balanced and that the effect of the errors on both Acast and the advertisers cancel out, which is not the case in the sick patient example. Nevertheless, it is of interest for Acast to be aware of these rates and gain insight in how distinctive the miss-classification problem is. [25], [50], [14]

A confusion matrix can be computed in order to describe the performance of a classification method. It is represented by a matrix with dimensions n x n, where n is the number of classes in the learning problem. The columns represent the predicted classes, while the rows represent the actual classes, as seen in Figure 7. The diagonal values of the matrix corresponds to the predictions which corresponds to the actual class and hence are correctly predicted. The confusion matrix reports the number of false positives, false negatives, true positives and true negatives for each class. [43]

		Predicted Class				
		Class 1	Class 2	Class 3	Class 4	Class 5
Actual Class	Class 1	TP_1	E_{12}	E_{13}	E_{14}	E_{15}
	Class 2	E_{21}	TP_2	E_{23}	E_{24}	E_{25}
	Class 3	E_{31}	E_{32}	TP_3	E_{34}	E_{35}
	Class 4	E_{41}	E_{42}	E_{43}	TP_4	E_{45}
	Class 5	E_{51}	E_{52}	E_{53}	E_{54}	TP_5

Figure 7: Confusion Matrix

Based on the values that can be obtained from the confusion matrix it is possible to calculate different rates that can be used to get an indication of how well a model is performing. The accuracy is the most common one, see Equation 2. There are several other rates that may be of value, all listed below, together with an example given for Class 1.

$$Overall\ Accuracy = All\ TP/All\ values\ in\ the\ matrix^2 \tag{2}$$

$$Sensitivity = TP\ Rate = TP/(TP + FN)^3 \tag{3}$$

$$Ex:\ Sensitivity\ for\ Class\ 1 = TP_1/(TP_1 + E_{12} + E_{13} + E_{14} + E_{15}) \tag{4}$$

$$Specificity = TN\ Rate = TN^4/(TN + FP)^5 \tag{5}$$

$$Ex:\ Specificity\ for\ Class\ 1 = TN_1/(TN_1 + E_{21} + E_{31} + E_{41} + E_{51}) \tag{6}$$

$$Precision = TP/(TP + FP) \tag{7}$$

[2]TP = The correctly predicted values for each class, i.e. the diagonal values of the confusion matrix

[3]FN = Sum of all the values in the corresponding row, excluding TP

[4]TN = Sum of all the values in all rows and columns, excluding the row and column for the class

[5]TN = Sum of all the values in all rows and columns, excluding the row and column for the class, FP = Sum of all the values in the corresponding column, excluding TP

$$Ex: \ Precision \ for \ Class \ 1 = TP_1/(TP_1 + E_{21} + E_{31} + E_{41} + E_{51}) \quad (8)$$

$$FP \ Rate = 1 - Specificity = FP/(FP + TN) \quad (9)$$

The sensitivity or the true positive rate is the rate that the event is correctly predicted for all samples having the event, i.e it measures the accuracy for the events of interest while the specificity or the true negative rate is the rate that nonevent samples are predicted as nonevents. It is common that there is a trade-off between the sensitivity and specificity, an increased sensitivity of a model means a decreased specificity. [43] In the example described earlier, regarding the classifying of a patient, it is more likely that it is preferred to have a high sensitivity than specificity; it is fine to classify a healthy patient as sick as long as all sick patients are classified as sick.[43]

9.3 Bias and Variance

An expected error of a model consists of an irreducible error and a reducible error, where the latter is a combination of the bias and the variance. The goal is to minimize both the variance and the bias in order to decrease the total error.

The variance refers to how much the estimation of $\hat{y}_i$ varies when using a different training data set. Different training data sets result in a different $\hat{y}_i$ as the set is used to fit the statistical learning method. Ideally the different estimations of $\hat{y}_i$ should not vary, but if a method has high variance, small changes of the data set will result in large changes of $\hat{y}_i$. High variance may cause overfitting, meaning that random noise in the data set is modelled by the algorithm

Bias refers to the part of the error which appears when trying to approximating a real-life problem. Often the correlation between the input values and the target values cannot be modelled using a simple model, due to its complexity, which leads to bias. High bias implies that the model is too simple and unable to model the complexity of the problem.

The bias-variance trade-off describes the problem when increasing the complexity of a model, one will encounter an increase in variance and a decrease in bias. Finding the optimal combination, where both components of the error are minimized without significantly increasing the other component is the goal. Unfortunately, neither the bias nor the variance can be calculated in practice since they depend on the target function which is unknown. [34], [66], [84], [24]

$$E[(y - \hat{f}(x))^2] = (Bias[\hat{f}(x)])^2 + Var[\hat{f}(x)]) + \sigma^2 \quad (10)$$

where

$$Bias[\hat{f}(x)] = E[\hat{f}(x)] - f(x) \quad (11)$$

and

$$Var[\hat{f}(x)] = E[\hat{f}(x)^2] - E[\hat{f}(x)]^2 \quad (12)$$

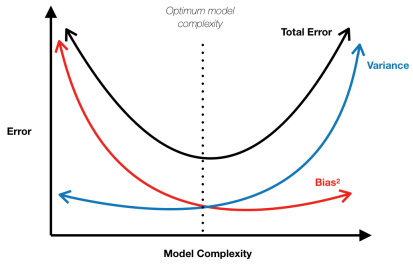

Figure 8: The Bias-Variance Trade-Off

9.4 Over- and Underfitting

Overfitting occurs when the model fits the training data "too" much, see Figure 11. The training error will in the case of overfitting be much smaller than the test error. This is a result of the model working too hard to adjust to the patters found in the training data. The model might memorize irrelevant patterns from the training data that in fact should be ignored. Ideally, the training error should not be smaller than test error, but it will be whenever overfitting is present. Overfitted models should not be used in the real world since it is not able to correctly predict the outcome for new data. [55]

The opposite of overfitting is underfitting and it occurs when a model is unable to identify and adjust to certain trends or patterns of the data, i.e the model does not fit the data sufficiently, as seen in Figure 9. Unlike overfitting, underfitting is a result of a too simple model. [55]

One way to tackle overfitting or underfitting is to perform cross validation, which is further explained in Section 9.7.2. Cross validation provides an indication of how well the model will perform on unseen data and helps evaluate the robustness of the predictive model. Another way to prevent overfitting from occurring is to train the model with more data. It can also be successful to remove irrelevant features by performing feature selection. [64]

Finally, regularization is a further approach to prevent overfitting. Regularization is especially preferred when working with a large number of input features and it is a technique which tries to prevent learning a more complex or flexible model and hence minimize the risk of the model overfitting the data. Regularization is done by penalizing the coefficient estimates, by minimizing it towards zero. There are two commonly used regularization methods; Lasso regression, referred to as L1, and Ridge regression, called L2. What differs the two methods is the penalization term, where L1 uses the absolute value of the magnitude while L2 uses the squared magnitude.

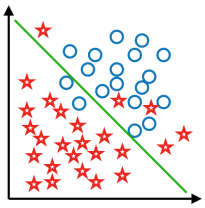

Figure 9: Underfitting

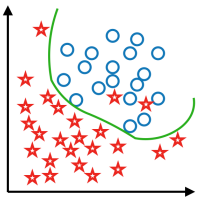

Figure 10: Appropriate Fitting

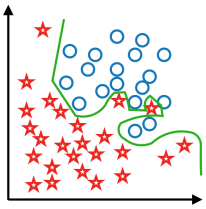

Figure 11: Overfitting

9.5 Statistical Models

9.5.1 Logistic Regression

A logistic regression method can be used when the dependent variable is categorical. Logistic regression does not only predict the value or category of a variable but it can also predict the associated probability i.e the probability that a variable belongs to a certain category. Logistic regression is easy to perform and interpret and it is therefore a commonly used method within machine learning.

The name logistic regression might seem confusing since the model is used for classification problems and not regression. The reason behind the name is that a linear regression model is used but the output is transformed to a value within the interval (0,1) by the logistic function. The value within the interval

is then interpreted as a class probability. When the classification problem is binary the output can take only two possible values which is encoded to either 0 or 1 i.e $Y \in \{0, 1\}$. When having a multi class problem we have K possible classes and the output can take several values, $Y \in \{1, ..., K\}$. The conditional class probabilities for a multi class problem can be defined as seen in Equation 13. [17]

$$q_k(X) = Pr(Y = k|X) \qquad k = 1, ..., K. \tag{13}$$

If the class K is selected as the reference class the model can be defined using the $K - 1$ log-odds between the first $K - 1$ classes and the class K as presented below. [17]

$$\begin{aligned} log\,\tfrac{q_1(X;\theta)}{q_K(X;\theta)} &= \beta_{01} + \textstyle\sum_{j=1}^{p} \beta_{j1} X_j, \\ log\,\tfrac{q_2(X;\theta)}{q_K(X;\theta)} &= \beta_{02} + \textstyle\sum_{j=1}^{p} \beta_{j2} X_j, \\ &\vdots \\ log\,\tfrac{q_{K-1}(X;\theta)}{q_K(X;\theta)} &= \beta_{0(K-1)} + \textstyle\sum_{j=1}^{p} \beta_{j(K-1)} X_j. \end{aligned} \tag{14}$$

When choosing a reference class an equivalent model can be obtained if the log-odds are forced to use one of the K classes as reference. The model has $(K - 1) * (p + 1)$ parameters in total, $\beta_{01}, ..., \beta_{p1}, ..., \beta_{0(K-1)}, ..., \beta_{p(K-1)}$ which are collected in a parameter vector θ. The class probabilities can be computed by inverting Equation 14 and using that $\sum_{k=1}^{K} q_k(X; \theta) = 1$, which will result in Equation 15. [17]

$$\begin{aligned} q_k(X;\theta) &= \frac{e^{(\beta_{0k} + \sum_{j=1}^{p} \beta_{jk} X_j)}}{1 + \sum_{l=1}^{K-1} e^{(\beta_{0l} + \sum_{j=1}^{p} \beta_{jl} X_j)}}, \qquad k = 1, ..., K-1 \\ q_K(X;\theta) &= \frac{1}{1 + \sum_{l=1}^{K-1} e^{(\beta_{0l} + \sum_{j=1}^{p} \beta_{jl} X_j)}}. \end{aligned} \tag{15}$$

Based on Equation 15, an expression for the log-likelihood of the training data can be derived and this can later on be maximized using numerical optimization. The log-likelihood is given by Equation 16. By maximizing the likelihood of the observed training data $D = \{(x_i, y_i)\}_{i=1}^{N}$, for the parameterization of the class probabilities q_k, the model parameters can be found.[17]

$$log\, l(\theta) = \sum_{i=1}^{n} log\, q_{y_i}(x_i; \theta) = \sum_{i=1}^{n} \sum_{k=1}^{K} I(y_i = k) log\, q_k(x_i; \theta). \tag{16}$$

When having a binary problem we can use the simple encoding, $y_i \in (0,1)$ which results in $I(y_i = 1) = y_i$ and makes the log-likelihood equation much more simple. When having a multi-class problem an option in order to make

the log-likelihood equation more simple is to encode the K classes, using one-hot encoding. This simplification allows us to write the log-likelihood function as following, presented in Equation 17, where $q_k(X;\theta)$ is given by Equation 15. [17]

$$log\, l(\theta) = \sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik} log\, q_k(x_i;\theta) \tag{17}$$

9.5.1.1 Implementation of Logistic Regression

The built in function *LogisticRegression* from *sklearn.linear_model* has been used in order to make a prediction using logistic regression. Since the output in this study has more than two classes the *multi_class* hyper parameter is fixed and set to *multinomial*, which implies that the training algorithm uses cross-entropy loss. Cross-entropy loss is used to measure the performance of a classification model when the output is a probability value within the range (0,1). Other hyper parameters that have been considered are *penalty* and *solver*. The penalty can take two values, *L1* or *L2*, and they are used in order to specify the norm used in the penalization. The solver specifies what algorithm that should be used in the optimization problem. Multi-class problems can take four different values; *saga*, *sag*, *newton-cg* and *lbfgs*, where the last three solvers only handle L2 penalty. Further, the hyper parameter C decides the strength of the regularization. By minimizing C a stronger regularization can be applied to the model. The optimal values of the hyper parameters can be found by performing grid search and cross validation. [37], [7]

9.5.2 K-Nearest Neighbor

K-nearest neighbor, kNN, is another commonly used classification method. The idea is to classify by finding the most similar data points in the training data set and base the estimate on the true class values of these k closest points. When using kNN there is no explicit training phase done before the classification and therefore the method belongs to the lazy learning methods. Instead, if the data is generalized this is done during the classification process, which implies that the data can be classified directly without any further investigation. One problem associated with kNN is that it is required that the entire training set is kept in memory unless the data-set is reduced. Another factor making the method expensive is that the algorithm has to go through all data points for each classification in order to find the closest data points among all available data points. Therefore, it is recommended to use kNN on small data-sets with few features. Below is a description of the different steps of kNN. [62], [10]

1. The distance between the data point to be classified and every data point in the training set is calculated.
2. The k nearest data points are picked.

3. The data point is classified as the majority among the k nearest data points in the train set.

9.5.2.1 Implementation of kNN

The built in function *KNeighborsClassifier* from *sklearn.ensemble* has been used in order to make a prediction using kNN. When using kNN it is important to carefully decide what value of k, represented by the parameter *n_neighbors* in the function, that is suitable. The bias and variance, discussed in Section 9.3, and hence the expected error in kNN is effected by the hyper parameter k. The choice of k will decide how many neighbors are included in the majority vote and therefore the classification of a data point. Increasing the value of k may result in a increased bias and a decreased variance. [67]

Furthermore it is important to decide what distance metric that should be used. *Euclidean* and *Manhattan* distance are two commonly used algorithms for distance metrics, presented in Equation 18 and Equation 19.

$$Euclidean\ Distance : d(x, y) = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2} \tag{18}$$

$$Manhattan\ Distance : d(x, y) = \sum_{i=1}^{m}|x_i - y_i| \tag{19}$$

The choice of the distance parameter determines the way of calculating the distance between two data points and the parameter p controls the choice of either Manhattan or Euclidean being used. The algorithm parameter can be set to *auto* which automatically chooses the algorithm most appropriate or to a specific equation such as *ball-tree*, *kd-tree* or *brute*. Depending on the dimension and the size of the training set and what algorithm is chosen, the performance of the model can vary. For a smaller data set with lower dimensions, brute force is preferred. For sparse data the two other algorithms are proved to perform better. Both kd-tree and ball-tree are binary trees with hierarchical structure, hence the query time increases with an increasing number of neighbors. In this study, these possible combinations of hyper parameters have been tested using grid search.

9.5.3 Neural Networks

Neural networks is one method used in machine learning which mimics the human brain and the biological neurons. If a network contains many hidden layers it is referred to as a deep neural network. [87], [60]

Neural networks consists of an input layer, a number of hidden layers and an output layer. The structure of a standard feed-forward fully connected neural network is visible in the image below, see Figure 12. As can be seen, each node of every layer is directly linked to each node of the next layer. The links propagates an activation from one node to another node, given the activation

function, θ. To each link there is a weight which controls the sign of the input and how much the link will affect the node. As a node usually have several inputs, all inputs are summed up, referred to as the "signals in" (s), and the activation function is applied to the summed input, returning an output. This is visualized more in detail in Figure 13. [48]

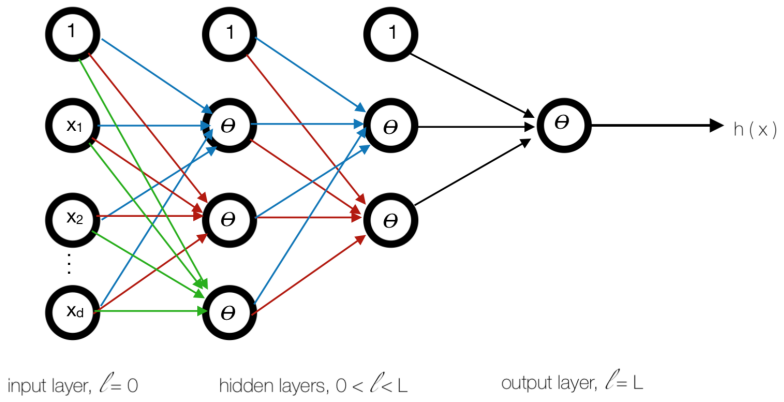

Figure 12: Neural Network

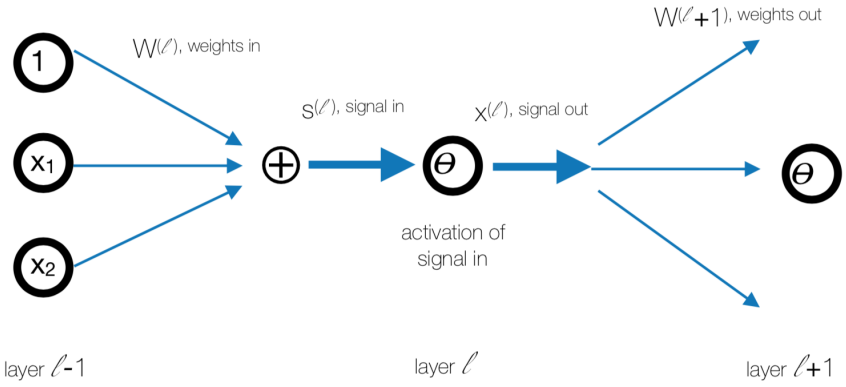

Figure 13: Parameters of layer l of a Neural Network

The output node of a neural network combines the outputs of the hidden layer nodes and delivers the final value. The activation function of the each node can either be a threshold or a mathematical function and there are several

possible activation functions. For instance are *Relu* and *Sigmoid* two commonly used functions. *Relu* returns an output x if it is positive and zero otherwise, which is seen in Equation 20 and Figure 14, while the *Sigmoid* function is non-linear and returns an output within the range (0,1), as an be seen in Equation 21 and Figure 15.

$$Relu : f(x) = max(0, x) \tag{20}$$

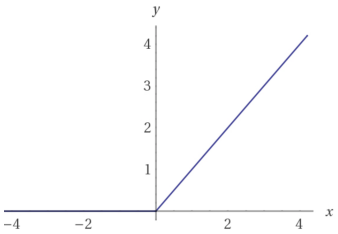

Figure 14: Relu Function

$$Sigmoid : f(x) = \frac{1}{1 + e^{-x}} \tag{21}$$

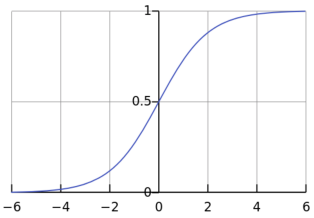

Figure 15: Sigmoid Function

For a multi-class problem, $Softmax$, which is a type of multi-nomial logistic regression, is often preferred as the the activation function for the output layer. Using $Softmax$, the neural network performs a non-linear feature transformation before applying logistic regression and hence turns a numeric score into probabilities with a sum equal to one. The class with the highest probability is the one which the input is most likely to belong to. The optimal case is where one class probability is equal to one while the remaining ones are zero. [87], [76]

There are several advantages with neural networks; they can be very powerful, flexible and they have a lot of potential to efficiently approximate complex target functions. The number of nodes in the hidden layer will affect the models ability to model more complex target functions and an increased amount of nodes will increase such ability. It is not the most complex statistical model but

may, if composed correctly, be very efficient. One weakness with the approach is that a neural network can easily overfit data which is something that must be taken into consideration when deciding on the number of nodes to use for instance. [88]

9.5.3.1 Implementation of Neural Networks

In this study a neural network approach has been applied to a solve multi-class problem. When handling a multi-class problem using neural networks, it means that the given input may belong to any of the possible output classes, in this case being five classes. The output label is defined as a one-hot-encoded vector and the number of nodes of the output layer corresponds to the number of possible output classes. [77]

The use of the library *Keras*, simplifies the process of building neural networks since it offers the necessary "building blocks" and is easy to use. Therefore, Keras was chosen for this study. When adding a layer using Keras, depending on what the hyper parameters are set to, either an input layer, a hidden layer or an output layer is created. Additionally, there are hyper parameters in the compiling and fitting of the model which determines how the network is trained and may have an impact on the networks performance. [11]

To summarize, the hyper parameters of a neural networks can both determine the structure of the network and how the network is being trained. These are vital details which have an effect on how well the neural network will perform and the selection of hyper parameters should be done under consideration. The values of the initial weights must be set and the structure of the network must be specified by deciding the number of hidden layers. The layer-specific parameters are the number of input and output nodes, as well as the choice of activation function. Finally, the use of regularization is also layer-specific and it can either be applied or not at each layer. The two types of regularization methods used in the study are L1 and L2, explained further in Section 9.4.

The hyper parameters affecting the training of the model, which are taken into consideration, includes the learning rate, the number of epochs and the batch size, all being set to numerical values. Additionally, depending on what learning problem one is dealing with, the loss function and the metrics must be decided. Since the output in this study has more than two classes, a number of hyper parameters must be fixed, such as the hyper parameter *metrics* being set to *categorical accuracy* and the *loss function* to *binary cross-entropy*. [13], [12], [89], [70] The use of categorical accuracy requires the target to be specified as a one-hot-encoded vector and the accuracy is returned in the same format. The vector sum will always be one and the most probable class is the one with the highest accuracy. The mean accuracy rate across all predictions for multi-class classification problems is calculated and returned. Binary cross-entropy is also referred to as *Sigmoid cross-entropy* loss and it is independent for each vector component i.e. class. It is preferred when working with a multi-class problem because the loss computed for one vector component is not affected by the other component values. [58]

9.5.4 Random Forest

Random forest is a tree-based algorithm and as the name reveals it creates a forest that is random to a certain degree. It is an ensemble method, using several classifiers and it classifies by for example taking a weighted vote of all trees predictions. [15] The idea is to combine several learning models in order to increase the overall result. Each tree, included in the random forest model, can be trained either using the entire data set or only a sample of the available data. When using only a sample, so called bootstrapping is carried out. The bootstrapping technique is characterized by generating the bootstrapped training data set by sampling with replacement from the full training data. [57]

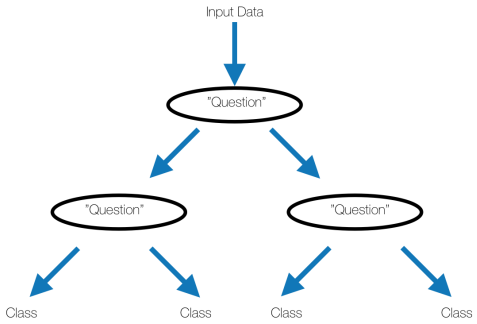

Figure 16: Decision Tree

In Figure 16, the composition of a decision tree is illustrated. The output of a decision tree is usually a probability of an input belonging to a certain class, rather than the class itself. Random forest as an algorithm can be described as a combination of several specialized and specific de-correlated decision trees, where each trees output is collected and combined into one common output. The method is applicable to a wide range of problems and often quite efficient. [15]

In classification, the random forest algorithm obtains an output of what class the input belongs to, from each tree. The final output is labeled as the majority vote of all trees outputs. [73]

When creating a new split in a tree, a random sample of predictors is chosen as candidates to that split, out of the total number of predictors. The split is made using only one of the chosen number of candidates and each time a new split is to be done, a new set of candidates is presented. The de-correlation results in the algorithm not being allowed to even consider the use of the majority of the possible predictors. The advantage of de-correlation is that if there is one very strong predictor, this cannot be used in the top split in all trees. In such way one can avoid having many decision trees resembling each other and hence avoid having highly correlated predictions. [21]

The information gain is one metrics used to control how a decision tree splits the data and how a tree is gorws. It is used to decide which feature that should be used for each split when the tree is built. The goal is to obtain a decision tree with as high information gain as possible. The information gain is based on the entropy which is a measure of randomness, where the lower the entropy, the more concentrated probability and the less randomness. This means that the empirical distribution of cases at a leaf node is less random which will result in hopefully one predominant class. A high entropy indicates a low level of purity and high level of uncertainty in a node. The goal is to decrease the uncertainty and keep splitting, and growing the tree, until a low entropy is reached. The entropy and the information gain are presented in Equation 22 and Equation 23. Here $p_1, p_2, ...p_i$ are fractions representing the percentage of each class present in the child node. [80], [45]

$$Entropy = -\sum_{i=1}^{K} p_i log_2(p_i) \tag{22}$$

$$Information\ gain = Entropy(Parent) - WeightedSum[Entropy(Children)] \tag{23}$$

In Equation 23, the parent is a node, which is divided into sub-nodes, referred to as children. The root node represents the entire population and it is further divided until reaching the terminal nodes, also called the leafs. All nodes in between the root and the leaf has parent nodes and child nodes.

The Gini impurity is the measure of total variance across the K classes and it is applied when working with a multi-class classifier. It measures how often a randomly chosen element is incorrectly classified. The Gini impurity is an alternative to the entropy as both are criterion's typically used to evaluate how good a split in the decision tree is. The lower the Gini impurity, the better the split. The Gini impurity can be computed as shown in Equation 24, where $i \in \{1, 2, ..., K\}$ is a set of items with K classes an p_i is the fraction of items labeled with class i. [47], [22], [63], [56], [57], [79]

$$Gini\ impurity = 1 - \sum_{i=1}^{K} {p_i}^2 \tag{24}$$

Out of the tree-based methods, random forest, is the method that de-correlates the decision trees it consists of. In terms of the bias and variance trade off, described in Section 9.3, random forest tackles the issue of high variance by combining many decision trees. Each individual tree has high variance, but low bias. The high variance is reduced by averaging over the trees. The de-correlation of the trees results in a further decrease of the variance as they are composed using different predictors. [21]

There are several advantages in using trees for decision making. Among these are that they are easy to visualize and display graphically, they are also

easy to explain as they closely resemble human decision making. As the algorithm aggregates many decision trees the predictive performance of trees can be improved, hence random forest can be seen as an efficient model. However, trees are not always very robust, so when introducing new data, the performance of a model might decrease. In other words, a high training accuracy might not always result in a high testing accuracy. [21]

9.5.4.1 Implementation of Random Forest

The built in function *RandomForestClassifier* from *sklearn.ensemble* is used in order to make a prediction using random forest. The hyper parameters that have been considered in this study are *n_estimators*, *max_features*, *max_depth*, *bootstrap* and *criterion*. The number of trees in the forest is described by the hyper parameter n_estimators and can be set to any integer value. The max_features hyper parameter describes the number of features used and can be set to either *sqrt*, *log2* or none. Using *sqrt*, the maximum number of features will be decided by taking the square root of the possible number of features, while for *log2* it is determined by taking the binary logarithm of the possible number of features. If set to none, all available features are used.

Max_depth describes the maximum depth of the tree and can be set to either an integer or none. When set to none the nodes are expanded until all leaves are pure, in terms of the chosen criterion, or until the leaves contain less than minimal number of samples required to split a node, which by default is equal to two.

Another parameter which can be varied is bootstrap, either being set to *True* or *False* and it indicates if bootstrap is used or not when building trees. If not used, the entire data set is used to build each tree. If bootstrap is used, random forest bootstraps the data for each tree and creates a tree that can only use a random sub set of the data set at each split. Hence, multiple different models can be created using the single training data set.

The hyper parameter criterion describes how to measure the quality of a split and can be set to either *gini* representing the measurement Gini Impurity or *entropy* representing the measurement Information Gain. [47], [22], [63], [56], [57]

9.6 Pre-Processing of Data

Pre-processing is the first step before introducing the data to the different models. The pre-processing phase include data cleaning, handling of missing data, converting and scaling data as well as splitting the data set into a training and a test set. Possible methods for pre-processing of data are described ahead.

9.6.1 Data Cleaning

The first part of the data cleaning process consists of removing unwanted observations from the data set including duplicates and irrelevant observations. It

is important to make an adequate decision when defining whether an observation is irrelevant or not. The second part is fixing the structural errors such as misspellings, typos, inconsistent capitalization and mislabeled classes. During this part of the process "confirmed to be wrong" outliers can be removed from the observations. What must be taken under consideration is that an outlier is innocent until proved guilty and an actual outlier can be of importance when training the model.

9.6.2 Missing Data

When dealing with data it is quite common to have observations with missing values. Before deciding how to handle the missing value entries it is important to determine where the data is missing at random (MAR), missing completely at random (MCAR) or missing not at random (MNAR).

One way to deal with missing data is to remove all rows or columns containing missing values. A danger with this approach is losing valuable information. Removing a column, i.e a feature, will result in an unbiased data set, but that might contain less information than the original data set. Removing rows will in the MCAR case result in unbiased data, while it can heavily bias the data in the MAR and MNAR case. Another approach is to fill the missing entries with simple statistics such as the *mean*. This method is commonly used but should preferably be avoided as the model will treat these values as actual values and it might affect the predictions.

A more complicated approach is to fill the missing entries with a estimated value. This is done using different algorithms such as expectation maximization which computes the maximum likelihood estimates. This approach will result in an unbiased data in the MCAR and MAR case but can bias the data in the MNAR case. [74], [44]

9.6.3 Converting Categorical Data to Numerical Data

One required measure sometimes needed to be taken is converting categorical values into numerical representations in order for them to be applicable in the algorithms. There are are several approaches to this, but the two methods used in this study are integer encoding and one-hot-encoding. Below, an example of each conversion method is presented. Both processes are easy to reverse and match with an array of labels in order to get knowledge about what the integer value stands for.

- Integer encoding converts unique category values into integers. An example of this can be presented based on one of the categories available in the data set used for the study. One column "Ad Position" can take on three possible categorical values such as "pre", "mid" or "post". In integer encoding "pre" would be represented by 1, "mid" by 2, "post" by 3. There is a natural order relationship between the integers which can be interpreted by the algorithm. Integer ordering can be problematic

as it implies relationships that are not present in the original ordinal or nominal variable.

- Table 4 and Table 5 describes the one-hot encoding conversion process. The first table shows the position being post, which in the latter table is represented by setting the category named post being true (equal to one) and the other values as false (equal to zero). One-hot-encoding is the most common and basic method of converting a token into a vector as it does not result in an ordinal relationship between the encoded categories. [27], [16]

Position
Post

Table 4: Data structure before one-hot-encoding

Pre	Mid	Post
0	0	1

Table 5: Data structure after one-hot-encoding

Integer encoding works well with algorithms not sensitive to the ordinal relationship between the encoded categories, such as random forest. One-hot-encoding is to prefer for algorithms that learns a weight for each variable. For example if "post" is represented by 3, while "pre" is represented by 1, that categorical value will be weighing stronger in the prediction, due to the value it has been assigned. Hence, one-hot-encoding is a better option for such algorithms, including for instance neural networks. However, one-hot-encoding results in a high dimensional data set. An increase in dimensionality might lead to problems finding a pattern in the data and more data is required in order to find it. This phenomenon is described by the term "the curse of dimensionality". It clearly states the problem of an increasing sparsity of data when increasing the dimensionality of a data set. Also redundancy among the features may occur. [4], [61], [75]

9.6.4 Feature Scaling

Scaling of the features can be done either with normalization or standardization. The advantage of feature scaling is that multivariate data will have a common unit, be transformed into the same range and be given the same importance in the algorithm. It is executed in order to prevent an ill-conditioned model or functions not working properly. Choosing a method which can be applied to all values is preferred. There are two different types of normalization, as seen in Equation 25 and Equation 26 . Normalization of features scales the values into the selected target range, being either [0,1] or [-1,1] depending on the data.

Standardization of values can be done to create a data set containing values with zero mean and unit variance.

- Min-max-normalization:

$$Xnew = \frac{X - Xmin}{Xmax - Xmin} \tag{25}$$

- Mean-normalization:

$$Xnew = \frac{X - \mu}{Xmax - Xmin} \tag{26}$$

- Standardization:

$$Xnew = \frac{X - \mu}{\sigma} \tag{27}$$

9.6.5 Data Split

There are three data sets commonly referred to in machine learning. A training set, a validation set and a test set. The test set should be unseen and not involved until the final testing is carried out.

The training set is used in the optimization algorithm that obtains values for the parameters in the statistical models, in other words, training the model. This is done in order to enable making predictions when introducing a new data set to the model. During the training process, several models are often included. A validation set can be used in order to evaluate the performance of the different models without having to involve the test set. The validation set is used to select the best model and the best hyper parameters. The test set is not used until the final model is to be tested and it can provide an unbiased indication of how well the final model will perform on new data. [85]

If having a data set of a limited size, the division of the data set into train, test and validation must be done in a way that utilizes the total amount of data in the best way possible. As the test set does not contribute to the efficiency of the model as it does not give any feedback to the learning, it might not be efficient to sacrifice too many observations to the test set. Still, in order to get an accurate estimation of the actual error, it has to be sufficient enough. Creating a learning graph and evaluating different ratios of the division of the data set is one way of finding the optimal split.

9.7 Optimizing the Learning Process

In order to optimize the learning process several approaches can be taken into consideration. The process can for example be optimized by finding the optimal hyper parameters and the optimal features to include.

9.7.1 Grid Search

Within the machine learning field there are two types of parameters, hyper parameters and model parameters. The model parameters are parameters in the model equation that are given values to minimize the error metric for the training data. A model parameter could for instance be the weights in a neural network. Hyper parameters can be described as decisions that are needed to be made to fully specify the model equation. An example of a hyper parameter could be the k in kNN.

Models can have several hyper parameters and it can be tricky to find the optimal combination. Grid search is a way of tuning hyper parameters that methodically evaluates every combination of the parameters in a grid. The grid search technique goes through all possible combinations of the hyper parameters and and are therefore very computationally expensive.
In this study the built in function *GridSearchCV* from *sklearn.model_selection* is used. The function delivers the optimal value of each hyper parameter by performing a cross validated grid search over the parameter grid. Cross validation is more thoroughly described in Section 9.7.2. The parameter grid is a dictionary with the names of the hyper parameters and the corresponding possible values they can take. When using the built in function GridSearchCV the hyper parameters *CV* and *estimator* must be defined. CV can be set to any integer value and specifies the number of folds, in a kfold cross validation. Estimator defines what method that is used and a scoring must that be passed.

9.7.2 Cross Validation

Cross validation is one of the most common re-sampling methods used in order to obtain more information about a model. The idea of re-sampling is to fit the same method multiple times using different sub sets of data. One drawback with re-sampling is that it can be very computationally expensive. Cross validation is a way to evaluate the training process and obtain an estimation of how well the model will perform on new data.

The estimated performance of each model is unbiased but as we choose the model based on these estimates it becomes biased for the selected model. When performing cross validation the observations are divided into different sets; a training set and a validation set. This is illustrated in Figure 17.

By using cross validation, the performance of each model included in the study can be measured and thereafter a selection of the best model can be done. The best model can thereafter be evaluated, using an unseen test set, which generates an unbiased estimation of the models performance. [23]

- k-fold Cross Validation: k-fold Cross Validation is one approach of cross validation where the data set is randomly divided into k sub sets, or folds, of the same approximate size. The procedure of fitting the method on the $k-1$ training folds is repeated k times, using different k folds as validation set each run, as can be seen in Figure 17. The k estimated test errors are used to calculate the k-fold cross validation as shown in Equation 9.7.2.

$$CV(k) = \frac{1}{k} \sum_{i=1}^{k} Err_i \tag{28}$$

It is common to set k to either five or ten and it is less computational expensive when choosing a smaller k. When choosing a larger k, the training data for each fold will be greater and the validation data set smaller. This will decrease the bias of the model and hence there is a correlation between the value of k and the bias.

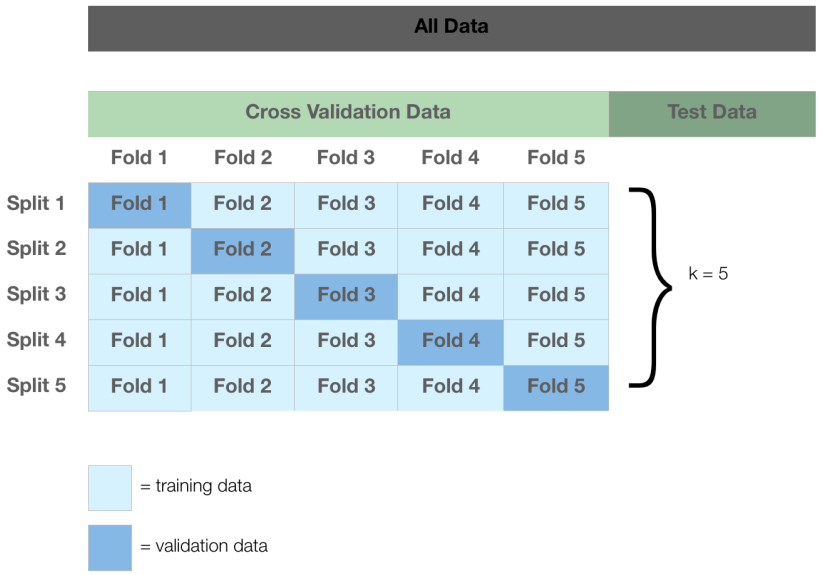

Figure 17: Division of data set for cross validation

In this study, k-fold cross validation is applied to improve the understanding of the models performance during training. The built in cross validation, available through *Scikit Learn* evaluates the estimator performance by using the training set. A separate validation set is therefore not needed. The training set is split into k smaller sets, as described above. The parameters the function takes as input are the estimators, the training set and the the k-value. There are two functions, that may be used, either *cross_val_score* which only returns the accuracy of each model or the more detailed one called *cross_validation* which also provides information about the running time. The mean accuracy of the cross validation performed on each model, as well as the running time, are taken into consideration when deciding upon a final model. [39]

9.7.3 Recursive feature elimination

Recursive feature elimination, RFE, is a way to decrease the number of features by removing the features of low importance. This will reduce the model complexity and help avoiding overfitting. The idea is to remove the weakest features one by one until specified number of features is reached. The features are selected by recursively considering smaller and smaller sets of features and at each step the weakest feature is removed. *Scikit Learn* provides an inbuilt function called *RFE* where the importance of each feature is obtained through either a *coef_ attribute* or a *feature_importance* attribute. The *coef_ attribute* is the coefficient of the features in the decision function for logistic regression while the *feature_importance* attribute returns the feature importance for random forest. [5], [40]

10 Data

For this study data has been retrieved from Acast's Azure SQL Database named "Acast Prod Warehouse" which hosts a large amount of data about podcasts, ads, creators and listens. We were given full access to the database together with an introduction of the tables and its content. First an inventory of the table structure and its connection was carried out and the tables of interest were identified. These choices were based on the general listening behaviour, described in Section 6.4. One could see that the behaviour varied depending on what time of the day and day of the week a podcast was listened to, as well as what category it belonged to.

10.1 Available Data

All features possible to track for downloads via the Acast application and via an RSS request were identified. The features that were available to track both in the Acast application and through RSS were included in the first selection of features. Out of these we selected the eight most informative features, presented in Section 10.3.

10.2 Pre-Processing of Raw Data

In anticipation of uploading data into the database, the raw data has been filtered and formatted by Acast. Different pre-processing methods have been described and some of the methods mentioned are used during the process executed by Acast.[25]

10.2.1 Abnormal Listening Behaviour

Abnormal listening behaviour include requests of podcasts from bots, automatic requests from apps and extreme listening behaviour in terms of a peak. These events are manually monetized and removed from the listening statistics if confirmed to be non-organic. [25]

10.2.2 Non Valid Ad Listens

Some ad campaigns run during a limited time. If a podcast including an ongoing campaign ad is downloaded but the actual listen is postponed until the ad has expired, filtering is done so that the ad listen does count as an impression.[25]

10.2.3 Conversion of Numbers

As statistics is collected in different formats some conversions into common metrics is carried out. Converting currencies, rounding off numbers, simple mathematical calculations are examples of what is done by Acast.[25]

10.2.4 Remove "Floating" Data

There are errors in the data caused by human operations which needs to be removed before added to the database. When content is published, an episode ID is created. If an episode is removed after generating a number of listens, this will results in "floating listens" connected to a non-existing episode ID. The same can appear if a registered user decides to remove its account. Such an action results in "loose" userIDs which needs to be removed from the database. [25]

10.3 Features

In this section the features that we decided to include in our two data sets are listed.

10.3.1 Ad

- Data type: Varchar.
- Describes: Describes the type of ad. Possible values are "sponsorship fixed", "airtime", "compensation" and "sponsorship impression based"
- Pre-processing of data: One-hot-encoding has been applied for all models.
- Justification of feature: The listening behaviour might be affected by the type of ad included in a podcast. Some ad types might generate a higher or lower listening rate.

10.3.2 Podcast

- Data type: Nvarchar.
- Describes: Name of the Podcast channel.
- Pre-processing of data: One-hot-encoding has been applied.
- Justification of feature: The name of the podcast might unveil that a listening segment of a certain podcast have a similar listening behaviour.

10.3.3 Position

- Data type: Varchar.
- Describes: Describes where in the podcast the ad is played. There are three possible values including "pre", "mid" and "post".
- Pre-processing of data: One-hot-encoding has been applied.
- Justification of feature: The position of the ad may have an impact on the listening rate as there is a drop-off rate in podcasts, which in general increases with time. [25]

10.3.4 Hour

- Data type: Tinyint.
- Describes: The hour of when the podcast was downloaded, possible values being 0-24. The feature is interpreted as the "time of the listen" even though there can be an offset in time of download and time of the listen.
- Pre-processing of data: No conversion needed.
- Justification of feature: It is shown that listening behaviour differs depending on time of the day, as seen in Figure 5.

10.3.5 Year

- Data type: Smallint.
- Describes: Describes the year of when the ad impression was consumed.
- Pre-processing of data: No conversion needed.
- Justification of feature: The feature is used to create new features.

10.3.6 Month

- Data type: Tinyint.
- Describes: Describes the month of when the ad impression was consumed.
- Pre-processing of data: No conversion needed.
- Justification of feature: Listening behaviour is shown to differ depending on the month of the year, as seen in Figure 3.

10.3.7 Day

- Data type: Smallint.
- Describes: Describes the day of when the ad impression was consumed.
- Pre-processing of data: No conversion needed.
- Justification of feature: Listening behaviour is shown to differ depending on the day of the week, as seen in Figure 4.

10.3.8 Sales Vertical

- Data type: Nvarchar.
- Describes: Describes the different target groups of a podcast, represented by a segment of listeners. Even though a podcast can have several sales verticals, only the first one listed in the database is retrieved as they are prioritized after relevance. All possible values are listed below.
- Pre-processing of data: One-hot-encoding has been applied. Several missing values identified.
- Justification of feature:

  The sales vertical represents the target audience of the show and should therefore be strongly related to actual listeners grouped by interest. Individuals in such listener groups may have similar behavior.[18]

AFL	History	Parenting & Families
Australian Voices	Hockey	Reality TV
Black Voices	Latino Voices	Sport & Activity
Branded	LGBT	Storytelling
Business & Finance	Lifestyle	Tech
Comedy	Men	Travel
Curious Thinkers	Men 18-24	True Crime
Daily Briefing	Men 25-44	Women
Education & Learning	Men 45+	Women 18-24
Film & Entertainment	Millenials	Women 25-44
Food & Drink	Music	Women 45+
Football	News & Politics	Young Affluents
Health & Wellness	NRL	

10.3.9 Content Category

- Data type: Varchar.
- Describes: Describes the different content categories that a podcast belongs to. Even though a podcast can belong to several categories, only the first one listed in the database is retrieved as they are ordered after relevance. All possible values are listed below.
- Pre-processing of data: One-hot-encoding has been applied.
- Justification of feature: The content category indicates the major themes of the podcast. It is a feature used to simplify the navigation among podcasts for the listeners. Podcasts within the same category might have similar listeners and hence similar behaviour.

Art	Games and Hobbies	Science and Medicine
Business	Health	Society and Culture
Comedy	Kids & Families	Sports & Recreation
Education	Music	Technology
Entertainment	News and Politics	TV and Film
Food and Drink	Religion	

10.3.10 Ad Quartiles

- Data type: Float.
- Describes: Describes how far into an ad a listen has been reached. Possible values being 0, 0.25, 0.5, 0.75 and 1.
- Pre-Processing of data: Integer encoding has been applied.
- Justification of target: This is the target of the machine learning problem in this study.

10.4 Pre-Processing of Data

10.4.1 Removing Sales Vertical

When exploring the data set it became evident that several rows for data were missing a value of the feature *Sales Vertical*. Each podcast should be assigned at least one sales vertical during the on boarding process and it was shown that a sales vertical had not been assigned to some of the podcast by mistake. The missing values of the feature *Sales Vertical* are immediately correlated to the feature *Podcast*, as the missing data was consistent in terms of only appearing together with some of the podcasts. The missing values of the *Sales Vertical* were handled as if it was "missing at random". Therefore, two approaches could be considered; removing the data rows containing missing data or removing the entire column, as discussed in Section 9.6.2. Based on the advantages and disadvantages of the different approaches, it was decided that the column was removed from the data set, hence not including the sales verticals in the study.

What motivates the choice of removing the column *Sales Vertical* is that it leads to less biased data set than if decided to fill in the missing data entries with simple statistics or removing the rows with a missing value. Biasing a data set is when one changes the probability distribution and the sample, i.e. the data set, does no longer have the same distribution as the population, i.e. the reality. When removing an entire feature it may result in loosing valuable information. But in this case, the removal of *Sales Vertical* is compensated by including the feature *Content Category* in the data set. The feature *Content Category* which does not miss any data is similar to *Sales Vertical* in terms of what information it provides. The number of content categories are less than the number of sales verticals, hence it can be seen as an aggregation of the Sales Verticals. In addition, an advantage of removing the feature *Sales Vertical* is

that the one-hot-encoding would generate a great number of new features and therefore the removal can be seen as an advantage as this decreases the total number of features.

10.4.2 Creating New Features

Two new features were created by combining of a number of features included in the original data set. These are presented below.

- Weekday: By combining the features *Year*, *Month* and *Day* the weekday could be identified and set as a new feature. The reason behind creating the new feature, *Weekday*, with this information was to be able to show the difference in listening behavior throughout the week. As can be seen in the Figure 4, the weekday seems to effect the number of listens.
- Time of day: Another general behaviour, shown in Figure 5, reviles that the listening intensity differs depending on the part of the day. By creating evenly distributed time intervals representing the 24 hours of a day, a listen could be assigned to a part of the day based on what hour it was downloaded. The intervals used for the creation of the new feature, *Time of Day*, with the intervals "Night":0-6, "Morning": 6-12, "Day": 12-18 and "Evening": 18-24.

10.5 Data Sets

The information available in the database is represented by several different formats of data including integers, real numbers, strings, date and time. A combination of relevant values was used to create data sets for the study. Two types of data sets of different characteristic has been used throughout the process. Below the difference between the two data sets are described as well as a motivation to why two data sets were included in the study. [71]

10.5.1 Natural Data Set

The natural data set consists of 50 000 rows, where each row contains information about an ad which has been delivered in a podcast. As can be seen in the diagram below, Figure 18, the data is not evenly distributed over the classes. There is one class, class 1 which represents an ad being listened to until completion, which is overly represented. A data set of such characteristic may affect the algorithm to predict according to the predominant class, as this is the most common case in the training set. Due to this, another data set has been created, a more balanced data set.

10.5.2 Balanced Data Set

Due to the fact that Acast launched their skipping function very recently, many users might not be aware of it. Over time, we assume that the listening be-

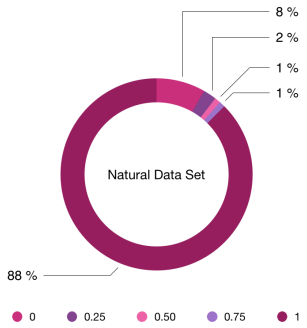

Figure 18: Natural Data Set - Distribution of Classes

haviour in terms of skipping ads might change. As listeners become more familiar with the skipping function, the distribution will most likely become more even, why we decided to create a balanced data set.

The balanced data set contains the same amount of rows as the natural data set but the classes are more evenly distributed. The new distribution of classes is presented in Figure 10.5.2. An advantage with working with a balanced data set is that by increasing the number of observations belonging to the minority classes, one enables the algorithm to have more observations from each class to train on. The majority class is still kept larger than the other classes, but not as extreme as in the natural data set. Keeping the natural distribution, to a certain degree, was done in order for the training set to still represent and resemble the test set, which is used in the evaluation of the model.

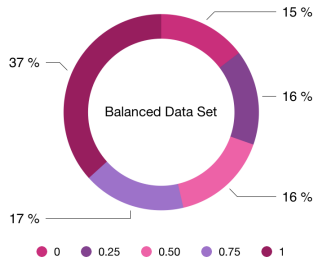

Figure 19: Balanced Data Set - Distribution of Classes

10.5.3 Benchmark

In the study several algorithms have been tested on the two different data sets, the balanced and the natural data set. In order to evaluate how efficient the al-

gorithms are, a benchmark was obtained for each set. The goal was to beat the benchmark, and hence beat "random". Any accuracy higher than the benchmark indicates that the algorithms is sufficient and can be used for predicting. [8] For both sets the benchmark accuracy has been the base rate, i.e. the accuracy of predicting predominant class.

- Benchmark for natural data set: 88%
- Benchmark for balanced data set: 37%

10.5.4 Splitting the Data and Learning Graph

In order to find the optimal split, where both the training and the test accuracy is maximized, an evaluation of the proportion of the samples devoted to the training set has been carried out. Using the built in function from Scikit Learn *sklearn.model_selection.train_test_split* and random forest with its default values as the predictor, a test in order to find the optimal split could be accomplished. The parameter *test_size* was assigned different values between 0-1 in order to identify if there was a difference in performance. [69], [86]

In Figure 20 and 21 two graphs are presented, one for the natural data set and one for the balanced data set. The graphs illustrate how the performance of a model, in terms of test and training accuracy, differs when changing the size ratio between the test and the training data set. By analyzing the trends in the learning graphs an indication of the optimal split can be provided.

Figure 20: Learning Curve for the Balanced Data Set

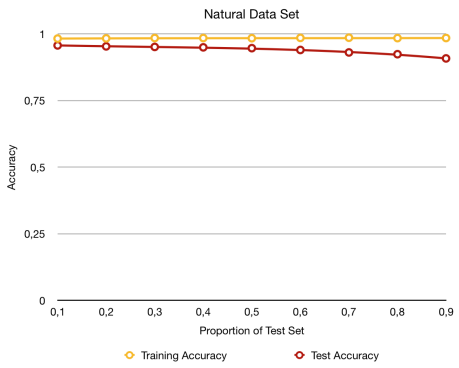

Figure 21: Learning Curve for the Natural Data Set

By observing the graph presented in Figure 21 it can be concluded that the optimal split for the natural data set is right before the test accuracy starts to level of, in this case around 0.3. This means that 30% of the data set should be devoted for testing, and the remaining part used for training of the model. In Figure 20, the learning graph for the balanced data set is presented. As one can clearly see, the training accuracy remains stable for all ratios, while the test accuracy varies vigorously. The unsuspected behaviour may be a result of human error but we noticed during the study that when using the balanced data set, the accuracy has been varying a lot for each run of the models. However, we found that the optimal split for the balanced data set was 0.2. This means that 20% of the data should be used for testing and 80% for training.

Cross validation using the training set has been applied when training and evaluating the different models so there was no need for a split to create a validation set. [46], [32], [38]

11 Experiment

A process carried out, referred to as the experiment, included finding the optimal hyper parameter values for each model and evaluating each models performance.

In an attempt to investigate the importance of each feature, one feature at a time was excluded from the data set and the models were run and the accuracies noted. In the tables below, Table 8, 11 and 14, these accuracies are presented for the natural data set. The accuracies are presented together with a + or a -, representing a positive or negative difference from the accuracy when the original data set including all features was used. This is one way of testing the relevance of a certain feature in a specific model, and is used in order to evaluate and decide whether to include certain features or not.

A number of selected hyper parameters for each model, presented in Table 6, 9, 12 and 15, was evaluated in order to find the optimal values. By performing the grid search the optimal values of hyper parameters could be identified, these are presented in Table 7, 10 and 13. For neural networks the chosen values of the hyper parameters are presented in 16. The training and validation accuracy of running the models with the optimal values is presented together with the accuracy when using the default values in Table 18 and 17.

11.1 Logistic Regression

Hyper Parameters for Logistic Regression	
multi_class	multinomial
solver	newton-cg, lbfgs, sag, saga
C	7 values in range (10^{-3} , 10^{3})
penalty	L1, L2

Table 6: The hyper parameters and values included in the grid search.

Optimal Hyper Parameters - Logistic Regression		
Parameter	Balanced	Natural
multi_class	multinomial	multinomial
solver	newton-cg	newton-cg
C	0.1	0.001
penalty	l2	l2

Table 7: The best combination of optimal hyper parameters.

Accuracy when removing one feature - Logistic Regression	
Excluded feature	Natural (diff)
None	87.33%

Podcast	87.56% (+)
Ad	87.13% (-)
Hour	86.68% (-)
Year	87.64% (+)
Month	87.11% (-)
Day	87.69% (+)
Position	87.65% (+)
Category	87.56% (+)
Time of Day	88.03% (+)
Weekday	87.49% (+)

Table 8: Validation accuracy when removing one feature from the data set.

11.2 kNN

Hyper Parameters - kNN	
n_neighbors	range (5, 30)
algorithm	auto, ball_tree, kd_tree, brute
p	1, 2

Table 9: The hyper parameters and values included in the grid search.

Optimal Hyper Parameters - kNN		
Parameter	Balanced	Natural
n_neighbors	5	10
algorithm	brute	ball_tree
p	1	1

Table 10: The best combination of optimal hyper parameters.

Accuracy when removing one feature - kNN	
Excluded feature	Natural (diff)
None	96.72%
Podcast	96.56% (-)
Ad	96.10% (-)
Hour	95.69% (-)
Year	96.83% (+)
Month	96.13% (-)
Day	96.60% (-)
Position	97.50% (+)
Category	96.55% (-)

Time of Day	97.10% (+)
Weekday	96.46% (-)

Table 11: Validation accuracy when removing one feature from the data set.

11.3 Random Forest

Hyper Parameters - Random Forest	
n_estimators	100, 200, 400, 600
max_features	auto, sqrt, log2, None
max_depth	10, 30, 50, 70, 100, None
criterion	gini, entropy
bootstrap	True, False

Table 12: The hyper parameters and values included in the grid search.

Parameter	Balanced	Natural
Optimal Hyper Parameters - Random Forest		
n_estimators	100	400
max_features	auto	None
max_depth	50	None
criterion	entropy	entropy
bootstrap	False	True

Table 13: The best combination of optimal hyper parameters.

Accuracy when removing one feature - Random Forest	
Excluded feature	Natural (diff)
None	99.40%
Podcast	98.74% (-)
Ad	99.11% (-)
Hour	95.83% (-)
Year	99.60% (+)
Month	99.39% (-)
Day	98.00% (-)
Position	96.38% (-)
Category	99.61% (+)
Time of Day	99.50% (+)
Weekday	99.61% (-)

Table 14: Validation accuracy when removing one feature from the data set.

11.4 Neural Networks

Model Parameters - Neural Networks	
no_layers	any integer value
no_epochs	any integer value
learning rate	value between 0-1
loss function	binary cross_entropy
batch size	any integer value
metrics	categorical_accuracy
Hyper Parameters for each layer	
activation function	relu, sigmoid, softmax, None
units	no. of nodes in the next layers, any integer
kernel_regulizer	L1 or L2
bias_regulizer	L1 or L2

Table 15: The hyper parameters and values included in the grid search.

Due to no change in accuracy when varying the hyper parameters, a table with the best combination of hyper parameter is not included in the report. We are unsure of the reason behind the neural network not being changed in performance when changing the hyper parameters. It is an unexpected behaviour and the network created was tested using an entirely different data set to see if it would change its predicting behaviour, but with no result. Due to the models odd behaviour, we did not carry out the same experiments presented for the models described above.

However, the parameter values that are used in achieving a training and testing accuracy are listed in Table 16.

Chosen Parameters for Neural Networks	
no_layers	2
no_epochs	300
learning rate	0.1
loss function	binary_cross_entropy
batch size	5
metrics	categorical_accuracy
Hyper Parameters for each layer	
activation function	relu used for all layers except for the output layer where softmax is used
units	128, 64, 32, 5
kernel_regulizer	L2
bias_regulizer	L1

Table 16: The chosen hyper parameters.

11.5 Training and Validation Accuracy

The training accuracy of each model, when using the balanced and the natural data sets, is presented in Table 17 and the valiadtion accuracies are presented in Table 18.

Model	Balanced	Natural
Default Values		
Logistic Regression	37.61%	89.37%
kNN	92.04%	92.64%
Random Forest	94.77%	93.12%
Neural Network	39.41%	90.20%
Optimized Values		
Logistic Regression	38.96%	87.63%
kNN	92.13%	98.56%
Random Forest	94.93%	99.82%
Neural Network	39.41%	90.20%

Table 17: Training Accuracy

Model	Balanced	Natural
Default Values		
Logistic Regression	68.54%	89.20%
kNN	47.56%	88.98%
Random Forest	49.98%	94.66%
Neural Network	37.39%	88.12%
Optimized Values		
Logistic Regression	68.56%	87.33%
kNN	49.24%	96.72%
Random Forest	57.46%	99.40%
Neural Network	37.39%	88.12%

Table 18: Validation Accuracy

The validation accuracy obtained for each model was compared to the benchmark for each data set, namely being 37% for the balanced data set and 88% for the natural data set. The validation accuracy obtained using the optimal values were compared model-wise in order to make a decision of which model to use as our final model. After analyzing the accuracies presented in Table 14 it was decided to keep all features in the final model since the accuracies did not differ significantly.

11.6 Choosing the Final Model

Based on the validation accuracy of each model and data set, presented in Figure 18, a final model could be selected. We based our decision on the prediction ability in terms of overall accuracy and a confusion matrix was produced in order to analyze the sensitivity and specificity of each class for our chosen model. The combination generating the highest accuracy was random forest being trained using the natural data set. Even though this is the overall best performing model, it can be of interest to state that for the balanced data set logistic regression was the most appropriate model. When evaluating the logistic regression model trained with the balanced data set, an accuracy of 68.56 % was obtained, which indicates an increase of 85 %, when compared to the benchmark for the balanced set. That increase of accuracy is remarkably higher than the increase generated when using random forest trained on the natural data set but the decision of final model was entirely based on which model that generated the highest overall accuracy and not the highest increase in accuracy.

- **Final Model:** Random forest, trained using the natural data set

11.6.1 Confusion Matrix & Metrics

A confusion matrix for our final model, presented in Figure 22, reveals how the model has predicted for each possible class. In Section 9.2.2, equations for metrics possible to calculate based on the confusion matrix are presented, these can provide additional information regarding the models performance.

		Predicted Class				
		0.0	0.25	0.50	0.75	1.0
Actual Class	0.0	1152	4	0	0	54
	0.25	0	299	0	0	10
	0.50	0	0	139	0	4
	0.75	0	0	0	139	2
	1.0	6	6	1	3	13181

Figure 22: Confusion Matrix for Final Model

Using Equation 3, the sensitivity for each class is calculated. The results are presented in Table 19. As can be seen in the table, our sensitivity rates are high for each class, meaning that our model has a high ability to make correct predictions for each class. Additionally, using Equation 7, the specificity of each class is calculated and presented in Table 20. The specificity is also high, but our majority class is the class with the lowest specificity. The overall high specificity is also confirmed when observing the confusion matrix, where the

diagonal values, marked in green, indicates the correctly made predictions for each class.

Sensitivity for each class	
Class 0.0	0.952
Class 0.25	0.968
Class 0.50	0.972
Class 0.75	0.986
Class 1.0	0.999
Mean Sensitivity	0.975

Table 19: Sensitivity Rates for Final Model

Specificity for each class	
Class 0.0	0.999
Class 0.25	0.999
Class 0.50	0.999
Class 0.75	0.999
Class 1.0	0.961
Mean Specitivity	0.992

Table 20: Specificity Rates for Final Model

As we have mentioned earlier, it is common that there is a trade-off between a high sensitivity and a high specificity. It can be hard to obtain a model with high values for both metrics, but our model returns predictions generating both high sensitivity and specificity for each class. In addition, our high accuracy indicates that the model is suitable to use for predictions.

11.6.2 Testing the Final Model

After choosing the final model, the model was exposed to an unseen test set so that an out-of-sample, i.e. test accuracy could be obtained. The test accuracy indicates the performance of the predictor and reveals how well the model generalizes to unseen data. In other words, the test accuracy tells how the model will perform in practice, and is therefore a measure of how likely our future predictions is to be correct.

- **Test Accuracy of Final Model:** 93.85 %

The chosen model generated a test accuracy, which is an increase of 7 % compared to the benchmark of the natural data set.

12 Discussion

12.1 The Use of Machine Learning

The test accuracy of 93.85 %, which was obtained with our final model, random forest, using the natural data set for training indicates that the use of machine learning to predict listening behaviour is a plausible approach. The accuracy differed widely depending on what model was used and what data sets were used in the training and the testing process. All models, regardless of what data set was used in the training process, had to be tested and evaluated using the natural data set in order to obtain a realistic measure on the models performance and predicting ability of today. This is because the natural data set is the closest one can come to replicate the RSS-data sets which will be applied to the model on wards.

We have worked with a benchmark accuracy for each data set in order to evaluate the value the models can bring in predicting listening behaviour when encountering ads. When using the balanced data set for training, the benchmark accuracy is 37 %, while for the natural data set it is 88 %, both being equal to the proportion of the majority class. The goal was to create a model where the benchmark accuracy could be beaten to as such high degree as possible. If this would not be possible, one conclusion to be drawn would be that machine learning is not the suitable approach. Nevertheless, having an accuracy of 93.85 % when training on the natural data set indicates an increase (+6.65 %) of accuracy compared to the benchmark, which is why machine learning can be seen as a suitable approach.

12.2 Evaluation of Models

A good approach when choosing a model is starting with a linear model such as logistic regression and using that accuracy as baseline. This is what we did and thereafter we tried the non-linear classifier random forest and a selection of other models. When analyzing the performance of the other models used to solve this learning problem, a number of them resulted in a fairly poor prediction ability, which was reflected by its accuracy. A low accuracy may suggest that the correlation between the input values and the target values is not as evident as expected, but as the accuracy of each model differs so widely, and some models perform well, this is probably not be the case. Instead, what might be the case is that our data is not suitable together with some of our chosen models.

We chose to use random forest as our final model, due to its performance in predicting class being much better than the other models. The model generating the highest accuracy was, as mentioned before, random forest being trained using the natural data set. Random forest is a model performing well together with categorical data and it is not as affected by class imbalances as the other models. Therefore we assumed that the model would perform best, at least when using the natural data set. This was shown to be a correct expectation. Additionally, the model is not as sensitive to the pre-processing of the data as

the other models. As our pre-processing was not very pronounced, this might be one of the reasons to why random forest was successful with our data.

During our study we have worked with a number of other models which were deselected in the final evaluation step. There are several possible reasons behind each models non satisfactory performance. Logistic regression is a linear model and might therefore not perform well together with data missing a linear relation. But, if the decision boundary is linear, the model will possibly perform well. The accuracy obtained when training our logistic regression model, with the natural data set indicates that this is probably not the case, given our fairly low accuracy.

Another model included in our study was kNN. When using kNN, strongly overlapping clusters can prevent the algorithm from remembering local groupings, and hence obstruct classifications. The model has low bias and high variance, and the trade-off is affected by the value of K. When training using our natural data set the optimal K-value found in the grid search was ten, while for the balanced data set it was five. Increasing the value of K leads to more neighbors being included in the prediction of an observations class, but with an increasing K, bias also increases. kNN was the model performing second best out of all models tested and the performance may indicate that the class clusters are fairly separated and that machine learning can be applied to this learning problem.

Our worst performing model was neural networks. For neural networks to be efficient, data with a sufficient connection between the input and the output is required. If the random parts of the data exceeds the non-random parts it can be hard for the neural network to make successful predictions. Noise and class imbalances are two other factors which might affect the performance of a neural network negatively. Several compositions of our network was created, where the structure of the network and the training of the network was changed. The number of hidden layers, activation function as well as the use of regularization was altered and the network was evaluated in terms of accuracy. Even though differing the combinations of hyper parameters and network structure, no remarkable change in the accuracy of the predictions was generated. The network predicted according to the same behaviour for both data sets. The predicted class for each input became the same as the predominant class, and the probability of belonging to each class was the same for all input values. This revealed that the network did not take the characteristics of each input into account when predicting what class it belonged to. One conclusion that could be drawn given this result was that using a neural network for predictions together with our available data was not the best option as varying the network structure and the ambient parameters did not change the predicting ability of the model. Anyhow, it is more likely that something went wrong during our work with neural networks, as we managed to achieve a successful result using other models. The maximum accuracy reached for both data sets was almost the same as the benchmark and the proportion of the majority class of each data set. Hence, it was decided, due to the models poor performance, to exclude the model from the possible candidates for our final model.

For the majority of our models, when training with the balanced data set and using the natural data set for validation, the training accuracy is considerably higher than the validation accuracy. This can be interpreted as the model overfitting the training data, and not being able to generalize well when introduced to new data. One reason behind the overfitting can be the size of the data set being too small for such a complex model. Other reasons can be the lack of regularization, the ensemble size (number of trees in the random forest) or that the training and validation sets are very different in terms of class imbalances. Such difference can make it hard for the training set to represent the validation set properly. Another underlying cause may be that the pre-processing of the data is not being done correctly or a lacking correlation between input values and the target variables.

12.3 Are the Results of our Study Reliable?

The study is based on the assumption that the general listening behaviour of podcasts is similar to the specific listener behaviour tracked within the Acast app. This enables the models to be trained using data taken from the Acast app, while being tested using data from the RSS listens. The data used in training of the models resembles the data available from the RSS feed. The majority of the listeners within the app are non-registered users, as presented in Table 2 and Table 2. For Android the unregistered users represent 87 % and for iOS 73 %. This implies that no personal information such as gender and birth year is available, since this information is only requested when registering. This validates that the data from the Acast app is sufficient in predicting the behaviour of the RSS listeners, as we almost have access to the same type of data. The only difference is that the data collected via the Acast app allows us to see the actual listening behaviour as well.

Acast launched the new function allowing users to skip (jump ahead in the audio file) 15 seconds when listening to a podcast in their app in the beginning of 2019. The skip function has been available before but limited from being used when reaching the ad section. As we are studying the listening behaviour within ads, this is of interest as it affects the behaviour and allows listeners to control their listening pattern within the ads. What must be taken into consideration is that the data used in this study is collected during the initial faze after launching this additional skipping function. One can assume that not all users are early adopters, noticing and using the skip function during ads, since this has not worked prior to 2019. Hence, many observations in a natural, non-balanced data set will have "maximum quartile reached" equal to one, representing a full ad listen. There is a chance that this behaviour will change along time, when users identify the possibility of not listening to ads. The distribution over the quartiles reached, and hence the different outcome classes, will possibly also change over time and even out. In order to keep the model realistic and up to date, it is important to continuously extract new data to train the model on in order to ensure that the possibly new listening behaviour can be tracked in the training process and used when predicting.

12.4 Creating Value for Acast

In terms of value, we think that the model will bring insight in what the general listening behaviour within ads looks like. Acast have up until now not been able to present any information regarding the listening behaviour of the RSS-listens, this will probably bring a lot of value, not least for the sales team. With a fairly high accuracy our final model can be used in predicting the listening behaviour in ads for the RSS listens. More specifically, gain insight in how far into an ad the listener listens. This information is of value for the sales team working at Acast, when setting prices for ads. Depending on what knowledge is requested the input data can be varied. By for instance only using listening data from a specific show, one can gain insight in how their listeners tend to behave. Shows can be compared and the price to have an ad played in a specific show can be set based on the predicted listening rate of an ad, instead of the number of downloads. This can be of value for the advertisers as they probably wish to only pay for the ads actually being listened to, in other words not skipped ads, and hence counted as an ad impression. Another example of information that can be gained and used for price setting is information regarding at what time of the day ads tend to be less or more likely to be listened to.

12.5 Challenges

Many of our models required a numerical input, hence one hot encoding was applied to the categorical features. When downloading a data set of a given number of observations it will most likely result in different compositions of observations, therefore also different number of podcasts and content categories for instance. When one hot encoding, each unique categorical value will generate a new column in the data set. Due to the one-hot-encoding of the categorical features, one will end up with uneven dimensions for training set and the test set. This is problematic and the data sets cannot be used in the models due to this. There are two possible approaches, either to compensate the non-matching dimensions by removing columns from the data set with the largest dimension or by adding empty columns to the data set of smallest dimension. The approach chosen for our study was originating from the training set. For columns only existing in the training set, the correspondent columns were added to the test set and filled with zeros. If the test set contained columns non existent in the training set, these were removed from the test set.

To make sure the data sets used in the learning process were sufficient in terms of having enough observations representing all cases we made sure to select a number of podcasts to include in the learning process. Instead of including only one observation of a certain podcast in the training we decided to choose a number of podcasts to train on. After grouping and sorting the podcasts based on occurrences in the data sets, the top 100 shows were included in the study. In such way we could guarantee that there would be a large number of observations for each podcast and hence also each content category, as the two features are directly correlated to each other.

The majority of our models are sensitive to the input data and hence the pre-processing done to the data before it being introduced to the model, but after converting the categorical values into dummies through one-hot-encoding. Therefore we decided to carry out individual feature selection to each model but the results generated showed that the models preferred the use of all features. This result was somewhat astonishing as some features often are redundant and might only confuse the model and not contribute to the learning. Additionally, normalizing and standardization of the quantitative features was carried out but with no success in increasing accuracy of the predictions. When reflecting over these insights and results, we contemplated if the pre-processing was not carried out correctly, or if the data set was optimal in its original version.

13 Future Work

After deciding on a final model, what data set to use for training and the most suitable hyper parameters, the model can be applied to predict the listening behaviour using RSS data. Acast can use the final model to get insights in how the listens over the ad quartiles are distributed outside the Acast app. Future work includes predicting using RSS data, as of now we only have a model that can be used for predictions with an estimated accuracy of the predictions which the model can generate.

Further, "packaging" of the results could be carried out in order for Acast to be able to use them in an efficient way. For instance, a simplification of the results, by presenting segments which are more likely to listen to an ad, and vice versa, can help the key account managers to make the price setting of ads based on target group, type of ad, when an ad is delivered etc. Creating such segments in a clear way is a thing which could bring value to the company and make our results more easy to use in practice. Another future work of interest could be including third party demographics from the other platforms which offer downloads of podcast. Gaining more insight in what type of listeners exists, and how they behave could help improve the algorithm both in training and testing to see if our predictions are accurate. Additionally, working more with the pre-processing of our data could possibly provide more accurate models. Pre-processing can lead to new revelations of relationships in the data, and hence provide valuable information to the model. We decided only to present a confusion matrix for our final model and final data set. It could be of value to analyze the prediction ability of each model more in detail by including each models confusion matrix. To summarize, dedicating more time to the experiments to both the data and the models would be interesting.

Throughout the process we have worked with several models and two data sets. Including more models in the study could have been interesting as it is possible that there is another, more suitable model for this learning problem. Furthermore, circumstances might change and another model might become more appropriate to model a new listening behaviour. Therefore it might require the model selection to be redone in order to make accurate predictions. It became evident that logistic regression was the most suitable model when training on the balanced data set. As mentioned before, we assume that the future listening behaviour will be more evenly distributed across the different classes. Therefore, logistic regression might be a more appropriate model to use for predictions in the future. We have suggested that Acast not only re-train our final model with fresh data, but also performs a new model selection if noticing a change in behaviour.

14 Conclusion

The problem formulation of this study has been "Can machine learning be applied to obtain a credible estimate of how listeners through RSS tend to respond when exposed to ads in podcasts, using data from the Acast app?". Four models and two data sets have been used in order to investigate the described problem.

Acast, the company at which the study has been carried out, lacked knowledge about how the majority of their listeners, responded to the ads played in the podcasts. In other words, how far into an ad, in terms of quartiles, the listeners decided to listen. I February 2019, Acast launched a new feature within their application, enabled their users to skip ads in podcasts, hence allowing them to terminate the ad listen before it being played to completion.

Only 10 % of the total listens are done via the Acast app, the majority of the remaining part is done via RSS. The RSS listens are done through a number of other applications and platforms which have allowed users to skid ads prior to Acast launching the function.

As mentioned before, Acasts main revenue stream comes from the ad sales and hence the number of impressions delivered is of importance. Using machine learning and data collected from the Acast app, we have now created a model, which predicts to what degree an ad is listened to, with an accuracy of approximately 94 %. The model can be exposed to RSS data and predict until what quartile an ad listen is done. Up until now, a random guess has been done, based on what the general listening pattern within the Acast app looks like.

The results partly indicates that using machine learning to obtain an estimate of the ad listening behaviour for RSS can be successful. One of our goals was to beat the benchmark, which was set to the proportion of the majority class. Our final model predicts with an accuracy being 6.65 % higher than the stated benchmark, for the natural data set, which is not such a huge increase.

If one would predict all inputs to belonging to the majority class, an accuracy of 88 % would be obtained. Such an approach would classify all listens as if the ads had been listened to until completion. Such approach would not bring valuable insight to neither Acast or their advertisers. What our model offers is actual predictions to each possible class, according to the behaviour analyzed using the Acast app data, with an accuracy of 93.85 %. Our model provides better indications of how the listens are actually done outside the Acast app and with a higher accuracy than the benchmark. To conclude, even though the improvement of accuracy is not that high, the insight the model brings is of major value. Therefore, we claim that machine learning can be applied to obtain a credible estimate of how ads are received outside the Acast app, based on the behaviour within the app.

References

[1] Acast. *About Us*. URL: https://www.acast.com/about. (accessed: 2018-12-12).

[2] Acast. *Acast Secures $19.5MM USD Series B Funding*. URL: https://www.acast.com/press/releases/acast-secures-19-5mm-usd-series-b-funding. (accessed:2018-12-17).

[3] Know Online Advertising. *Quartile Reporting*. URL: http://www.knowonline.advertising.com/advertisingdictionary/quartile-reporting/. (accessed:2019-03-10).

[4] S. Aksoy and R. Haralick. *Feature Normalization and Likelihood-based Similarity Measures for Image Retrieval*. URL: http://www.cs.bilkent.edu.tr/~saksoy/papers/prletters01_likelihood.pdf. (accessed: 2019-04-10).

[5] Bakharia Aneesha. *Recursive Feature Elimination with Scikit Learn*. URL: https://medium.com/@aneesha/recursive-feature-elimination-with-scikit-learn-3a2cbdf23fb7. (accessed:2019-04-11).

[6] Dukes Anthony, Liu Qihong, and Shuai Jie. *Interactive Advertising: The Case of Skippable Ads*. https://editorialexpress.com/cgi-bin/conference/download.cgi?db_name=IIOC2018&paper_id=359. (accessed:2019-03-24). Apr. 2018.

[7] ML Cheatsheet. *Loss Functions*. URL: https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html. (accessed: 2019-02-19).

[8] Cornell CIS. *Performance Measures for Machine Learning*. URL: https://www.cs.cornell.edu/courses/cs578/2003fa/performance_measures.pdf. (accessed: 2019-05-01).

[9] Codefuel. *Website Monetization: Pay Per Play vs. Pay Per Impression vs. Pay Per Click*. URL: https://www.codefuel.com/blog/website-monetization-pay-per-play-vs-pay-per-impression-vs-pay-per-click. (accessed:2019-03-10).

[10] Soni Devin. *Introduction to k-Nearest-Neighbors*. URL: https://towardsdata.science.com/introduction-to-k-nearest-neighbors-3b534bb11d26. (accessed: 2019-02-17).

[11] Keras Documentation. *Layers*. URL: https://keras.io/layers/about-keras-layers/. (accessed:2018-11-14).

[12] Keras Documentation. *Metrics*. URL: https://keras.io/metrics/. (accessed:2018-11-14).

[13] Keras Documentation. *Usage of activations*. URL: https://keras.io/activations/. (accessed:2018-11-14).

[14] Sadawi Dr. Nourreddin. *Evaluating Classifiers: Confusion Matrix for Multiple Classes*. https://www.youtube.com/watch?v=FAr2GmWNbT0. (accessed:2019-02-24). Aug. 2014.

[15] Chollet François. “Deep Learning with Python”. In: Manning Publications, 2018. Chap. 1.2.

[16] Chollet François. “Deep Learning with Python”. In: Manning Publications, 2018. Chap. 6.1.

[17] Lindsten Fredrik et al. *Statistical Machine Learning - Lecture notes on linear regression, logistic regression, deep learning boosting*. Uppsala University. (accessed:2019-02-15). Feb. 2018.

[18] Ljungstedt Gabriella. *Interview*. 2018-11-30.

[19] James Gareth et al. “An Introduction to Statistical Learning: with Applications in R.” In: Springer:New York, 2013. Chap. 2.1.5.

[20] James Gareth et al. “An Introduction to Statistical Learning: with Applications in R.” In: Springer:New York, 2013. Chap. 2.2.1.

[21] James Gareth et al. “An Introduction to Statistical Learning: with Applications in R.” In: Springer:New York, 2013. Chap. 8.1-8.3.

[22] James Gareth et al. “An Introduction to Statistical Learning: with Applications in R.” In: Springer:New York, 2013. Chap. 8.1.2.

[23] James Gareth et al. “An Introduction to Statistical Learning: with Applications in R.” In: Springer:New York, 2013. Chap. 5.1-5.4.

[24] T Hastie et al. “An Introduction to Statistical Learning: with Applications in R.” In: Springer:New York, 2013. Chap. 2.2.2.

[25] Havsfrid Ioana. *Interview*. 2018-12-11.

[26] McAuley J. and Leskovec J. “Image Labeling on a Network: Using Social-Network Metadata for Image Classification.” In: *Lecture Notes in Computer Science* 7575 (2012). DOI: https://doi.org/10.1007/978-3-642-33765-9_59.

[27] Brownlee Jason. *Why One-Hot Encode Data in Machine Learning?* URL: https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/. (accessed:2019-01-06).

[28] Hare Jason. *What is metadata and why is it as important as the data itself?* URL: https://www.opendatasoft.com/2016/08/25/what-is-metadata-and-why-is-it-important-data/. (accessed:2019-01-11).

[29] Arundeep Kaur and Anant Vidya Nidhi. “Predicting Movie Success Using Neural Network”. In: (2013).

[30] Cherry Kendra. *Behavior Analysis in Psychology*. URL: https://www.verywellmind.com/what-is-behavior-analysis-2794865. (accessed:2019-04-25).

[31] Murphy Kevin P. “achine Learning: A Probabilistic Perspective”. In: Cambridge, 2012. Chap. 1.1.

[32] Dobbin Kevin and Simon Richard. “Optimally splitting cases for training and testing high dimensional classifiers”. In: *BMC Medical Genomics* 4.31 (2011), pp. 2–3.

[33] Gallagher Kevin. *Millennials skip YouTube ads... and that's ok*. URL: https://www.businessinsider.com/millennials-skip-youtube-ads-and-thats-ok-2017-1?r=US&IR=T. (accessed:2019-01-19).

[34] Renu Khandelwal. *Bias and Variance in Machine Learning*. URL: https://medium.com/datadriveninvestor/bias-and-variance-in-machine-learning-51fdd38d1f86. (accessed: 2019-03-10).

[35] Johan Kisro. *Interview*. 2019-12-02.

[36] IAB Tech Lab. "IAB Podcast Measurement Technical Guidlines". In: (2017).

[37] Scikit Learn. *sklearn.linear$_m$odel.LogisticRegression*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html. (accessed: 2019-02-25).

[38] Scikit Learn. *User Guide: sklearn.model$_s$election.train$_t$est$_s$plit*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html. (accessed:2019-04-10).

[39] Scikit Learn. *User Guide:Cross-validation: evaluating estimator performance*. URL: https://scikit-learn.org/stable/modules/cross_validation.html. (accessed:2019-04-09).

[40] Scikit Learn. *User Guide:sklearn.feature$_s$election.RFE*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html. (accessed:2019-04-10).

[41] Google Ad Manager. *Understand Video Solutions report metrics*. URL: https://support.google.com/admanager/answer/2759433?hl=en. (accessed:2019-02-01).

[42] Marketingterms.com. *Podcast*. URL: https://www.marketingterms.com/dictionary/podcast/. (accessed:2019-02-19).

[43] Kuhn Max and Johnson Kjell. "Applied Predictive Modeling". In: Springer:New York, 2016. Chap. 11.2.

[44] Ashcroft Michael. *Missing Data Basics*. URL: https://www.futurelearn.com/courses/advanced-machine-learning/2/steps/472962. (accessed:2019-03-06).

[45] Islam Sujan Nasir. *What is Entropy and why Information gain matter in Decision Trees?* URL: https://medium.com/coinmonks/what-is-entropy-and-why-information-gain-is-matter-4e85d46d2f01. (accessed:2019-03-14).

[46] Ritchi Ng. *Learning Curve*. URL: https://www.ritchieng.com/machinelearning-learning-curve/. (accessed:2019-04-12).

[47] OECD. *Gini Index*. URL: https://stats.oecd.org/glossary/detail.asp?ID=4842. (accessed: 2019-02-10).

[48] Norvig P. and Russell S. J. "Artificial Intelligence: A Modern Approach". In: Pearson Education: New Jersey, 2003. Chap. p. 729.

[49] U.S Patent. *AD SKIP FEATURE FOR CHARACTERIZING ADVERTISEMENT EFFECTIVENESS*. https://patentimages.storage.googleapis.com/ee/36/f3/5fa5b71eb67f39/US8468056.pdf. (accessed:2019-03-25). June 2013.

[50] Mather Paul and Tso Brandt. "Classification Methods for Remotely Sensed Data". In: CRC Press, 2009. Chap. p. 70-71.

[51] Poddindex. *Swedish Podcast Measurement Standards*. URL: https://www.poddindex.se/measurement. (accessed:2018-12-10).

[52] Acast - Power Point. "Acast betydelse i Mediemixen 2018". In: (2018).

[53] Acast - Power Point. "Acast General Information". In: (2018).

[54] Acast - Power Point. "Acast vs Radio". In: (2018).

[55] Shaikh Raheel. *Cross Validation Explained: Evaluating estimator performance*. URL: https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85. (accessed:2019-05-01).

[56] L. Raileanu and K. Stoffel. "Theoretical comparison between the Gini Index and Information Gain criteria". In: *Annals of Mathematics and Artificial Intelligence* 41 (2004), pp. 77–93. DOI: https://www.unine.ch/files/live/sites/imi/files/shared/documents/papers/Gini_index_fulltext.pdf.

[57] Prratek Ramchandani. *Random Forests and the Bias-Variance Tradeoff*. URL: https://towardsdatascience.com/random-forests-and-the-bias-variance-tradeoff-3b77fee339b4. (accessed: 2018-12-18).

[58] Gómez Raúl. *Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss etc*. URL: https://gombru.github.io/2018/05/23/cross_entropy_loss/. (accessed: 2019-04-11).

[59] Travis Rhee and Farhana Zulkernine. "Predicting Movie Box Office Profitability: A Neural Network Approach". In: (Dec. 2016), pp. 665–670. DOI: 10.1109/ICMLA.2016.0117.

[60] Sharma Sagar. *What the Hell is Perceptron?* URL: https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53. (accessed:2019-01-12).

[61] Saitta Sandro. *Standardization vs. normalization*. URL: http://www.dataminingblog.com/standardization-vs-normalization/. (accessed: 2019-04-25).

[62] Patel Savan. *Chapter 4: K Nearest Neighbors Classifier*. URL: https://medium.com/machine-learning-101/k-nearest-neighbors-classifier-1c1ff404d265. (accessed: 2019-02-17).

[63] Rahul Saxena. *How Decision Tree Algorithm Works*. URL: http://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/. (accessed: 2019-01-28).

[64] Elite Data Science. *Overfitting in Machine Learning: What It Is and How to Prevent It.* URL: https://elitedatascience.com/overfitting-in-machine-learning. (accessed: 2019-02-20).

[65] Retention Science. *Customer Churn Rate: Definition, Measuring Churn and Increasing Revenue.* URL: https://www.retentionscience.com/why-measuring-your-customer-churn-rate-increases-revenue/. (accessed:2019-03-15).

[66] Forthmann-Roe Scott. *Understanding the Bias-Variance Tradeoff.* URL: http://scott.fortmann-roe.com/docs/BiasVariance.html. (accessed: 2019-04-16).

[67] Fortmann-Roe Scott. *Understanding the Bias-Variance Tradeoff.* http://scott.fortmann-roe.com/docs/BiasVariance.html. (accessed:2019-01-24). June 2012.

[68] Spreaker. *What it is RSS and How does an RSS Feed Work with Podcasting.* URL: https://help.spreaker.com/kb/rss-feed/what-it-is-rss-and-how-does-an-rss-feed-work-with-podcasting. (accessed:2019-02-22).

[69] Shah Tarang. *About Train, Validation and Test Sets in Machine Learning.* URL: https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7. (accessed: 2019-04-01).

[70] Tensorflow. *Train your first neural network: basic classification.* URL: https://www.tensorflow.org/tutorials/keras/basic_classification. (accessed:2019-01-10).

[71] Tibco. *Data Types.* URL: https://docs.tibco.com/pub/spotfire/6.5.2/doc/html/ncfe/ncfe_data_types.htm. (accessed: 2019-02-23).

[72] Jones Tim. *Recommender systems, Introduction to approaches and algorithms.* URL: https://www.ibm.com/developerworks/library/os-recommender1/index.html. (accessed:2019-03-15).

[73] Hastie Trevor, Tibshirani Robert, and Jerome Friedman. "The Elements of Statistical Learning". In: Springer:New York, 2009. Chap. 15.3.

[74] Hastie Trevor, Tibshirani Robert, and Jerome Friedman. "The Elements of Statistical Learning". In: Springer:New York, 2009. Chap. 9.6-9.7.

[75] Hastie Trevor, Tibshirani Robert, and Jerome Friedman. "The Elements of Statistical Learning". In: Springer:New York, 2009. Chap. 2.5 3.8.

[76] Malik Usman. *Creating a Neural Network from Scratch in Python: Multi-class Classification.* URL: https://stackabuse.com/creating-a-neural-network-from-scratch-in-python-multi-class-classification/. (accessed:2018-12-16).

[77] Malik Usman. *Creating a Neural Network from Scratch in Python: Multi-class Classification.* URL: https://stackabuse.com/creating-a-neural-network-from-scratch-in-python-multi-class-classification/. (accessed:2018-11-12).

[78] Wikipedia. *Cost Per Impression*. URL: https://en.wikipedia.org/wiki/Cost_per_impressio. (accessed:2019-03-10).

[79] Wikipedia. *Decision tree learning*. URL: https://en.wikipedia.org/wiki/Decision_tree_learning. (accessed:2019-02-14).

[80] Wikipedia. *Entropy (information theory*. URL: https://en.wikipedia.org/wiki/Entropy_(information_theory). (accessed:2019-02-14).

[81] Wikipedia. *RSS*. URL: https://en.wikipedia.org/wiki/RSS. (accessed:2019-02-22).

[82] Abu-Mostafa Yaser, Magdon-Ismail Malik, and Lin Hsuan-Tien. "Learning from Data". In: Renesselaer Polytechnic Institute, 2012. Chap. 1.1.

[83] Abu-Mostafa Yaser, Magdon-Ismail Malik, and Lin Hsuan-Tien. "Learning from Data". In: Renesselaer Polytechnic Institute, 2012. Chap. 1.2.

[84] Abu-Mostafa Yaser, Magdon-Ismail Malik, and Lin Hsuan-Tien. "Learning from Data". In: Renesselaer Polytechnic Institute, 2012. Chap. 4.3.

[85] Abu-Mostafa Yaser, Magdon-Ismail Malik, and Lin Hsuan-Tien. "Learning from Data". In: Renesselaer Polytechnic Institute, 2012. Chap. 2.2.

[86] Abu-Mostafa Yaser, Magdon-Ismail Malik, and Lin Hsuan-Tien. "Learning from Data". In: Renesselaer Polytechnic Institute, 2012. Chap. 2.3.

[87] Abu-Mostafa Yaser, Magdon-Ismail Malik, and Lin Hsuan-Tien. "Learning from Data: e-Chapter Neural Networks". In: Renesselaer Polytechnic Institute, 2015. Chap. 7.1-7.4.

[88] Abu-Mostafa Yaser, Magdon-Ismail Malik, and Lin Hsuan-Tien. "Learning from Data: e-Chapter Neural Networks". In: Renesselaer Polytechnic Institute, 2015. Chap. 7.5.

[89] Abu-Mostafa Yaser, Magdon-Ismail Malik, and Lin Hsuan-Tien. "Learning from Data: e-Chapter Neural Networks". In: Renesselaer Polytechnic Institute, 2015. Chap. 7.6.

15 Appendix A - Interviews at Acast

15.1 Interview with Ioana Havsfrid, 2018-11-28

Interview about how podcasts are delivered, users, the Acast application and how data is collected.

- What information about the listeners does Acast have via the app?
- Is the app the only way of delivering podcast? If not, what other ways are there?
- Are all customers registered or can you listen via the app be done without an account?
- What is the ratio between registered and non-registered listeners?
- What meta data is available when a podcast is downloaded by a user (non-registered and registered)?

15.2 Interview with Gabriella Ljungstedt, 2018-11-30

Interview about how the on boarding of new podcast and the selling of ads in a podcast is handled.

- What does the on boarding process of new podcasts look like?
- What "features" of a podcast would you say are the most important ones when selling ads in a new podcast? (target group, theme etc)
- How are ads sold? Do you sell them to specifically fit into a podcast or to a number of podcast of a certain theme or target group? Explain.

15.3 Interview with Johan Kisro, 2018-12-02

Interview about ads. General questions about how ads are reported, what kind of ads are sold via Acast etc.

- What kind of ads are sold via Acast?
- Are there different types of ads?
- Do you collaborate with any other companies in the selling of the ads?
- What metrics does Acast use in the reporting of ads?
- How are the ad impressions reported back?
- Where in a podcast can ads be inserted?
- What metrics does Acast use in the reporting of ads?
- How long can an ad section be?

15.4 Interview with Ioana Havsfrid, 2018-12-11

Interview about the database, what features are relevant for the project and what pre-processing has been done before data is added to the database by Acast.

- What data is available in your database, referred to as 'the Warehouse'?
- What tables might be relevant for our project?
- Has any pre-processing of the data been executed prior to being added to the database? For what columns and how?
- Have you done any similar projects before using the data in the database to predict behaviour of your listeners?
- Are you aware of any columns with a lot of missing data in the database?