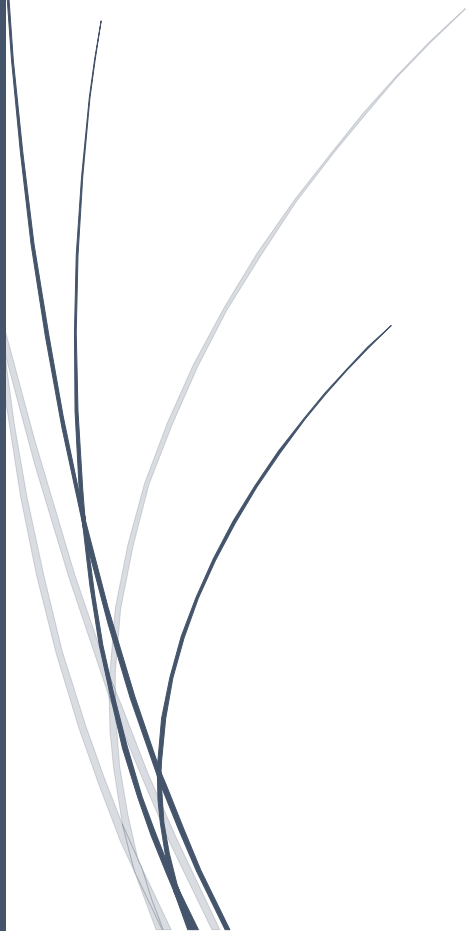
2020/2021

CALL CENTER DATA ANALYTICS

INDUSTRIAL DATA ANALYTICS PROJECT

Maria Migrova, ID: 20121291



National University of Ireland, Galway

Table of Contents

LIST OF FIGURES 2
1. INTRODUCTION 4
1.1. DESCRIPTION OF BUSINESS PROBLEM/ NEED 4
1.2. RESEARCH ON POTENTIAL ANALYTICS AND VISUALISATION TOOLS USED 4
LATENT DIRICHLET ALLOCATION 4
SENTIMENT ANALYSIS 6
NAÏVE BAYES 7
DECISION TREE 8
1.3. PROJECT OBJECTIVES 9
2. THE BUSINESS PROBLEM/ NEED 10
2.1. DESCRIPTION OF CONTEXT 10
2.2. DATA SOURCES 13
2.3. DATA PREPARATION 15
2.4. PRELIMINARY ANALYSIS 16
3. 17
4. THE SOLUTION 18
4.1. REFINED PROJECT OBJECTIVES 18
SENTIMENT ANALYSIS 22
NAÏVE BAYES 25
DECISION TREES 27
4.2. ANALYSIS AND VISUALISATION METHODS EMPLOYED 29
4.3. WORK BREAKDOWN AND PLANNING 30
5. EVALUATION OF THE SOLUTION 30
5.1. PRESENTATION OF THE SOLUTION 30
6. CONCLUSIONS 31
6.1. EVALUATION OF THE PROJECT AS A WHOLE 31
6.2. LESSONS LEARNT 32
6.3. POTENTIAL FUTURE WORK 32
BIBLIOGRAPHY 33

LIST OF FIGURES

1. Latent Dirichlet Method 4

2. Sentiment Analysis 5

3. Naïve Bayes 7

4. Decision Trees 8

5. 8x8 screen 1 9

6. 8x8 screen 2 9

7. Data 14

8. Corpus 15

9. Most common words 16

10. Number of topics 17

11. Agents numbers 18

12. Result dataset 18

13. Splitted dataset 18

14. Topic 1 20

15. Topic 2 20

16. Topic 3 20

17. Topic 4 21

18. Topic 5 21

19. Topic 6 21

20. Most common words 22

21. Wordcloud 22

22. Sentiment 23

23. Sentiment 2 23

24. Naïve Bayes 1 25

25. Naïve Bayes 2 26

26. Naïve Bayes 3 26

27. Naïve Bayes 4 26

28. Decision Trees 1 27

29. Decision Trees 2 27

30. Decision Trees 3 28

31. Decision Trees 4 28

1. INTRODUCTION

1.1. DESCRIPTION OF BUSINESS PROBLEM/ NEED

We are living in the age of data, where the data means the most powerful companies' asset. Businesses that use their big data hold a big competitive advantage over those that don't. Call centres are one of the biggest users of data analytics platforms. By receiving thousands of calls, emails, texts, and chat messages daily, they produce an enormous amount of data about their customers and agents. Understanding this data helps call centres in improving their operations, making faster business decisions, providing personalised customer service, generating more sales, reducing call volume, decreasing average handling time, etc… In this project, we will be cooperating with company called Kaptec, which provides flexible and scalable managed IT services, cloud communications, telephony, and contact centre outsourcing. Our project will be directed to analyse data from their contact centre activities which are provided for customers like Westnet, Vodafone, Eir, Bio-Medical, Cisco. Just recently Kaptec made a decision to move their PBX phone functionality to a cloud-based system provided by 8x8 which is powered by AI innovation.

1.2. RESEARCH ON POTENTIAL ANALYTICS AND VISUALISATION TOOLS USED

For analysing our data, we will be using R Studio [1], which is an open-source software designed specifically for data science teams. R is a language and environment for statistical computing and graphics. R is also extremely flexible and easy to use when it comes to creating visualisations. One of its capabilities is to produce good quality plots with minimum codes. R Programming lets us to visualise data by using a set of inbuilt functions and libraries to build graphs and present data.

LATENT DIRICHLET ALLOCATION

The biggest part of our project will be to create a topic modeling algorithm using LDA – Latent Dirichlet allocation method [2,3]. Which is a probabilistic generative model that extracts the thematic structure in a big document collection. The model assumes that every topic is a distribution of words in the vocabulary, and every document (described over the

same vocabulary) is a distribution of a small subset of these topics. The result of our topic modeling algorithm will be a number of topics which contains the list of most common words connected to that topic. For such a model as LDA, the most important parameter to define is a number of topics – K. If K is too small, the collection is devided into a few very general topics. If it's too large – the collection is devided into too many topics, which some may overlap, and others are hardly interpretable. LDA is a mathematical method for estimating both of these at the same time: finding the mixture of words that is associated with each topic, while also determining the mixture of topics that describe each document. There are a number of existing implementations of this algorithm.

The typical preprocessing steps before performing LDA are:

1) Tokenisation – process of splitting words into tokens.

2) Removing of special characters

3) Removing of stopwords

4) Lemmatisation- process of grouping forms of word so they can be analysed as a single item.

Using this algorithm, we will then find the most common topics discussed in the call center. In the figure below we can see how the Latent Dirichlet allocation method works. [4]

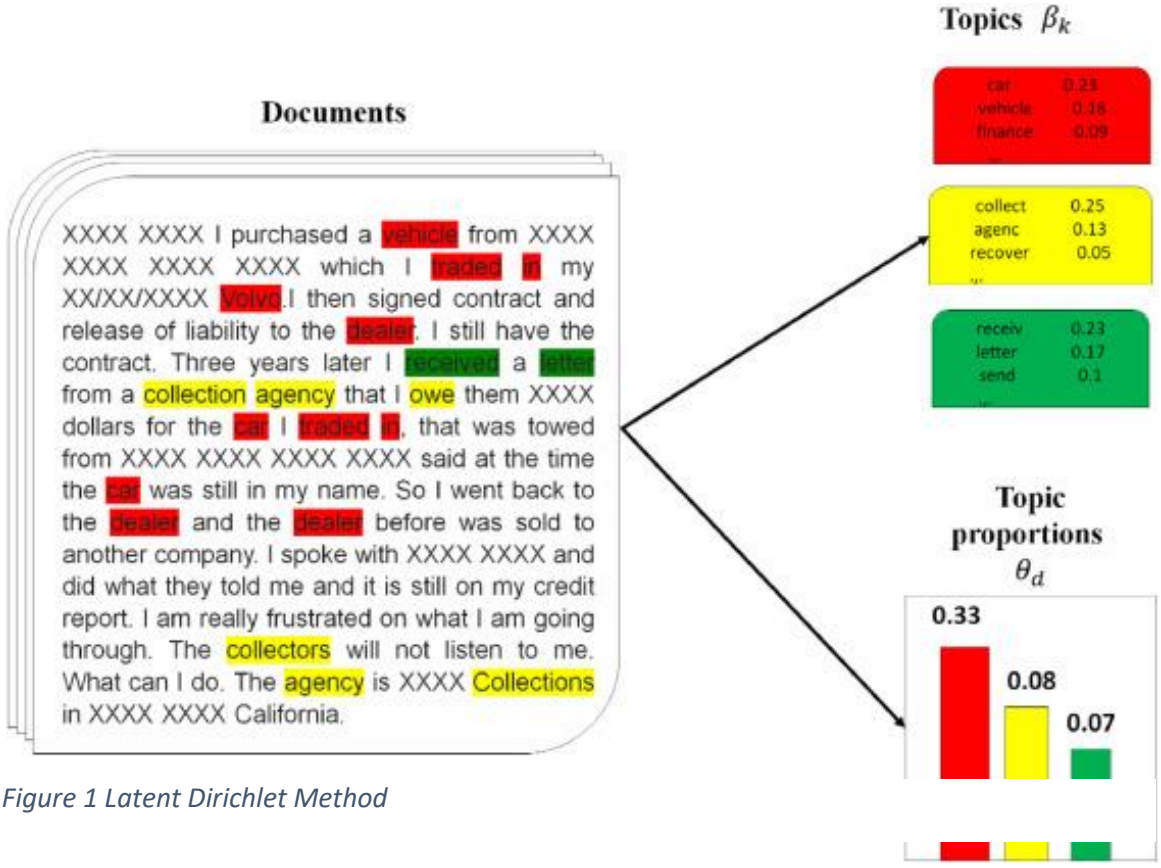


Figure 1 Latent Dirichlet Method

SENTIMENT ANALYSIS

When we split our conversations into different topics, we can then perform a sentiment analysis. From different words used in our conversations we can infer whether a part of text is positive or negative, or even characterise whether emotion is surprise or disgust. One way of analysing the sentiment of a text is to consider the text as a combination of its individual words and the sentiment content of the whole text as the sum of the sentiment content of the individual words. This is an often-used approach, and an approach that naturally takes advantage of the tidy tool ecosystem.

The three most common general-purpose lexicons for sentiment analysis in R are:

- AFINN
- bing
- nrc

All three of these lexicons are based on unigrams (single words). These lexicons contain many English words, and the words are assigned scores for positive/negative sentiment, and also emotions like joy, anger, sadness, etc… The nrc lexicon categorises words in a binary way (yes/no) into categories of positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise and trust. The bing lexicon categorises words in a binary way into positive and negative categories. The AFINN lexicon assigns words with a score that runs between -5 and 5, with negative scores indicating negative sentiment and positive scores indicating positive sentiment. [5] For our project we will be using the ncr lexicon.

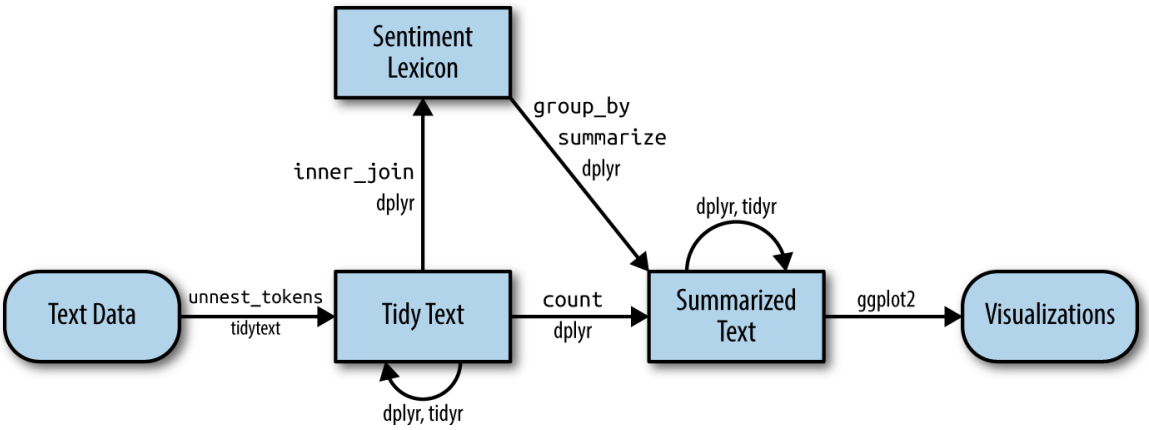


Figure 2 Sentiment Analysis

The next part of our project will be analysing a performance of each agent based on the different conversation topic. We will be able to find the topics in which each agent excels or on the other side needs more training to gain better results. The data will be labelled by using sentiment score, which will make it easier for us to evaluate each conversation into succesfull or unsuccesfull. We can then count a score for each agent for each topic to find out which topic is most suitable for each agent.

The last part of our project will be predicting sentiment of each topic using different predicting classification. We will be using queueNumber, confidence and duration of the call.

NAÏVE BAYES

Naive Bayes classifiers [6] are a collection of classification algorithms based on Bayes' Theorem**.** It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

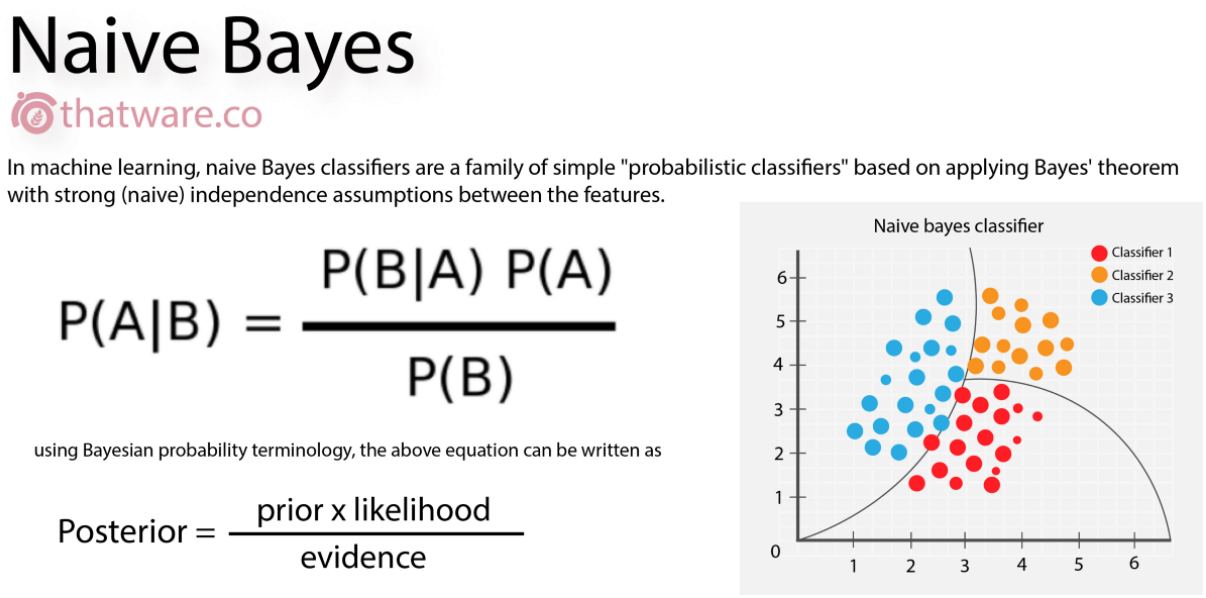


Figure 3 Naive Bayes

DECISION TREE

Decision Trees [7] are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

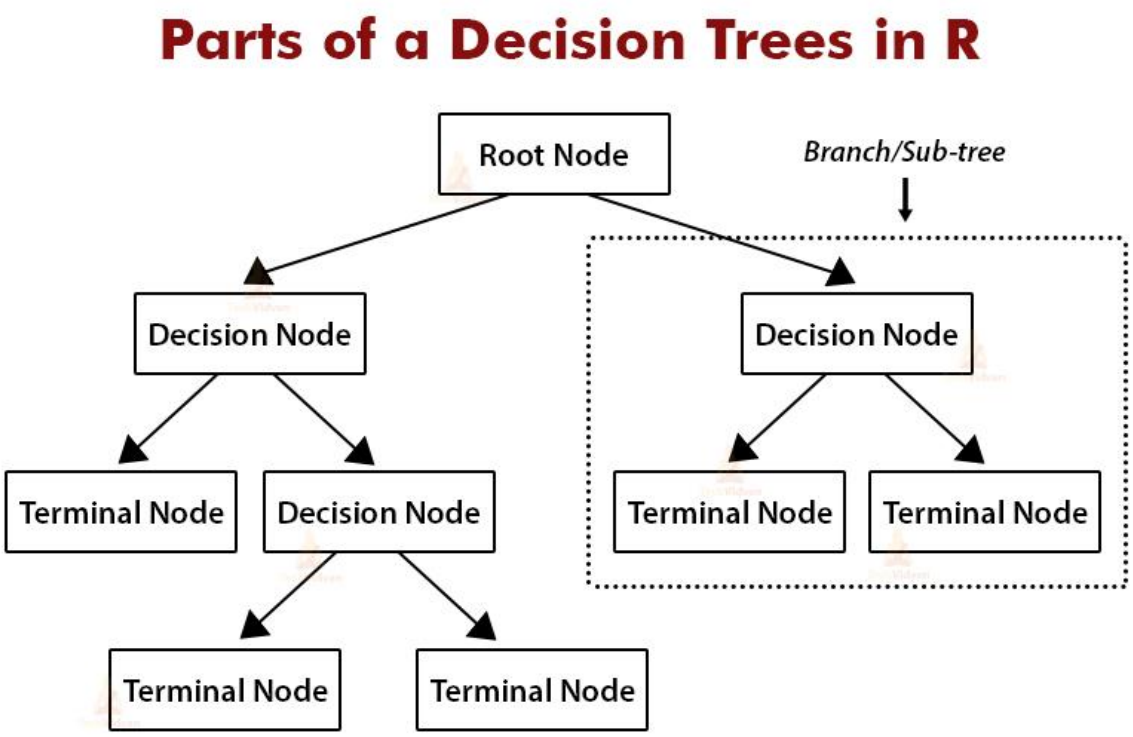


Figure 4 Decision Trees

With the gained results we will produce numerous graphics that will explain our results. For example: visualising the most common conversation topics over a time, 3 the performance of agents, satisfaction of clients over time, number of calls related to a different topic, average time to solve an issue related to different topic and more.

1.3. PROJECT OBJECTIVES

The main goal of our project is to:

- Analyse agents' efficiency from Kaptec company over time by analysing company's KPIs.

- Finding the most common topics occuring across all the transcripted conversations between agents and their customers by analysing them using topic modeling algorithm.

- Using supervisor's evaluation score, or by finding patterns between used phrases describing successful call (e.g. using Thank you for your help phrase, etc..),call duration time or if the call was transferred or not, we can then evaluate each call as successful or not.

- Calculate agent's succesfull rate for each topic and analyse their performance related to different topics.

- Visualise our results

2. THE BUSINESS PROBLEM/ NEED

2.1. DESCRIPTION OF CONTEXT

Kaptec [8] is a customer experience solutions company, where they developed a fantastic reputation as a solutions provider which allows them to work with the best companies and provide the most suitable technologies. This company is dynamic, focused and works closely with their customers. Kaptec's mission is to enable their customers to meet their business objectives through delivering the highest quality of service, working to industry best practice, innovating their services and product range to meet their customer's needs and maintaining market leadership. One of their main activities is a call center outsorcing. Call center outsourcing is the strategic business decision for managing customer's call center and customer service operations to leverage a 3rd party organisation. A contact centre manages all customer contact through a variety of media such as telephone, fax, letter, email, live chat and more. Software for call centers is designed to help clients reach customer service business. Call center software, designed mainly for telephone support, enables customers to communicate with call center customer service across multiple channels.

Kaptec has developed a fantastic reputation as a managed IT service provider. In early days their business was built on the expertise delivered via their contact center. What they have today is a dynamic, focused company that delivers the right results together with their highly trained and motivated consultants and leading technologies. They also offer 24/7 customer relationship management support to their Irish and international customers. Kaptec's contact centre solutions are designed to maximise the productivity of their organisation's workforce. Kaptec hosts contact center in the cloud, or its components. They provide multi-channel services on their network.

Kaptec provides services including:

- Telephone answering – their telephone answering call centre service is available 365 days of the year. Kaptec is an ISO accredited call center with full call recording and call reporting features.
- Email handling service – with reply in less than 1 hour to all emails.

- Live chat – customers are able to chat with on of Kaptec's agents in real time with their Live Chat Support Contact Center service.

Just recently Kaptec made a decision to move their PBX phone functionality to a cloud-based system provided by 8x8 [9] which is powered by AI innovation. Their innovative services include:

- ACD – skills-based routing direct customers to the best-qualified agent.
- IVR – delivers consistent and efficient support while reserving valuable agent resources for more complex, high-payoff interactions.
- Digital Channels – helps to stay ahead of changing customer expectations and meeting them in their channel of choice.
- Predictive Dialer – a fully integrated outbound dialing system which improves connection rates to increase the volume of completed calls, maximizing potential revenue opportunities.
- Quality Management – empowers agents and provides the consistency and confidence to reduce risk and meet compliance objectives.
- Workforce Management – accurately forecasts volume across interaction channels for optimum staffing and agent utilization. It generates schedules quickly and keep agents happy by aligning expectations with performance goals.
- Customer Surveys – using customer feedback dashboards.
- Call and Screen Recording – makes it easy to search, playback and retrieve and download recording.
- Customer Experience Analytics – delivers an unprecedented level of visibility into every aspect of customer interaction by visualisating of the full customer journey.
- IVR Journey Maps – showing which menus are effective, and where the adjustments are needed.
- Speech Analytics – allows every company to drive measurable improvements in omnichannel customer experience and agent performance.
- Native CRM – consolidates the contact center applications and customer data into a single unified interface.
- CRM Integrations – gives the choice and flexibility to personalise the customer experience and maximise agent productivity.

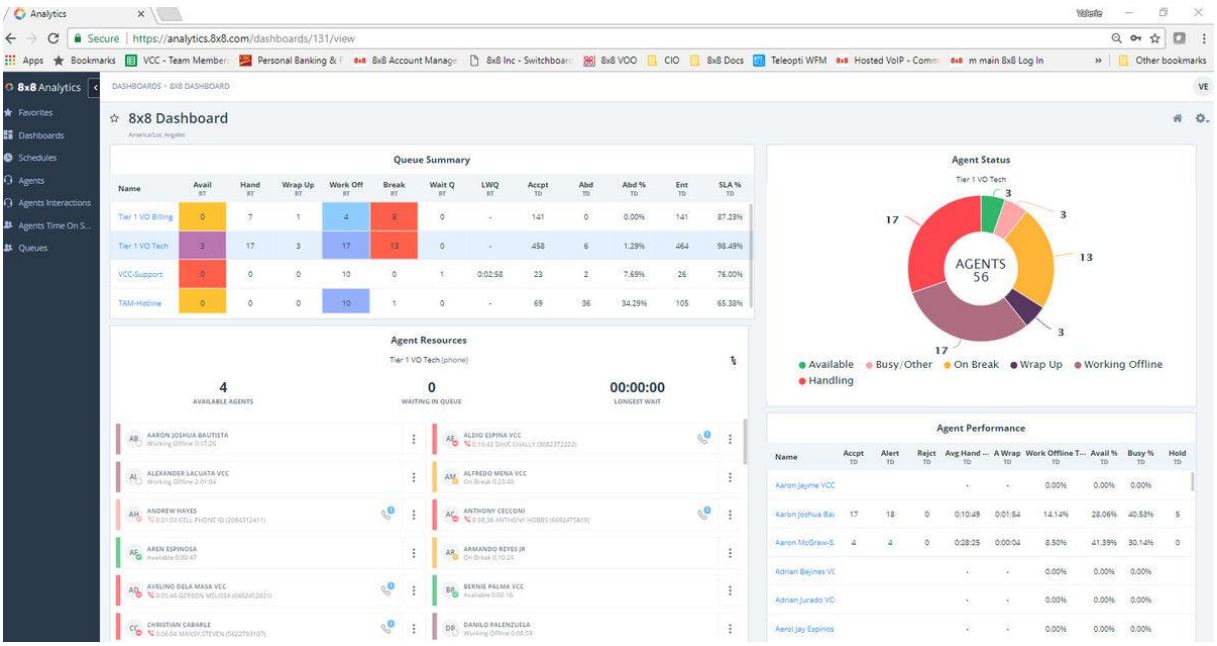


Figure 5 8x8 screen 1

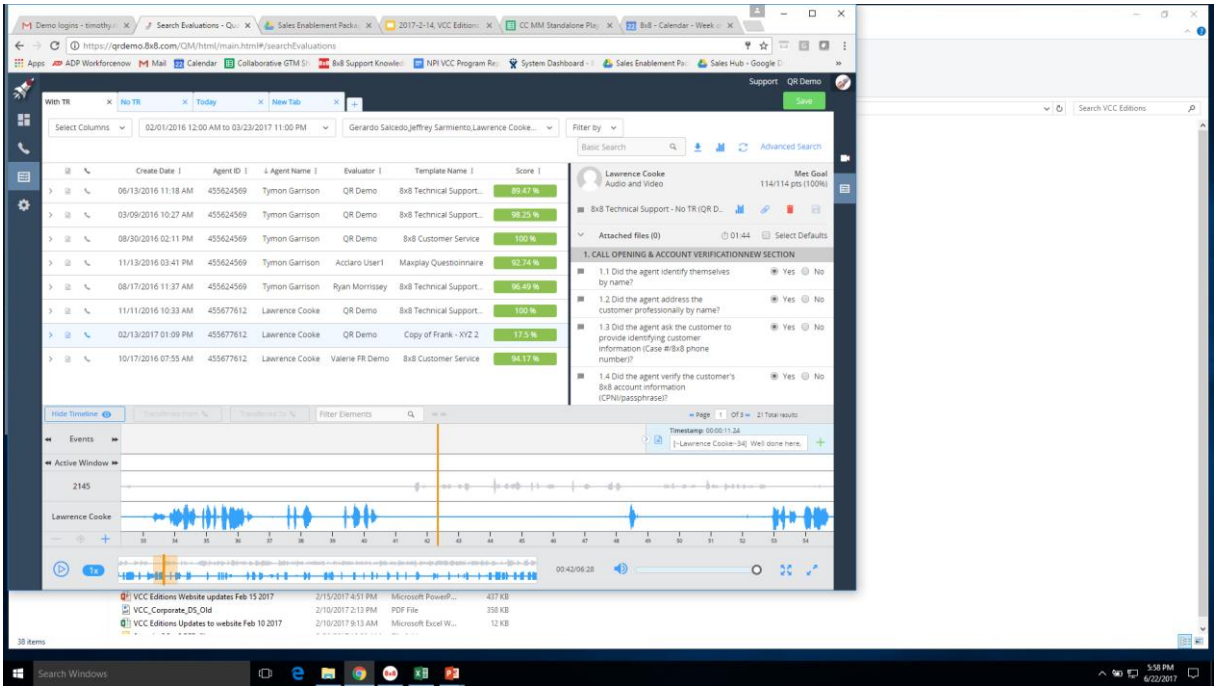


Figure 6 8x8 screen 2

Figure 5 and 6 showsan example of how 8x8 system looks like in practice.

We can see th 8x8's contact services are reallypowerfull. However even though they provide so many services, we can't see any services oriented on agents' conversations and finding a way for agents to get better. This is a part where our project could be helpful not only for us but also for Kaptec to see where their agents need trainig.

2.2. DATA SOURCES

Our data will be retrieved directly from 8x8 system or using API which will require creating an account first, and then retrieve the data using specialised software like Python or R Studio. We can access and download these data objects for the further analysis and use the available data from 8x8 system are as following [6]:

• 8x8's Cloud Storage Service (CSS) offers us a single point of access for Virtual Office PBX telephone, Virtual Contact Center call recordings, Virtual Contact Center screen recordings and Virtual Meetings.

• The Quality Management and Speech Analytics (QMSA) API enables us to access raw resources such as interaction metadata, evaluation results, and users.

• 8x8 Virtual Contact Center Analytics (VCCA) Historic Metrics API offers an entire suite of historical reports for agent interactions, agent status, and queue interactions.

• The Virtual Contact Center Analytics (VCCA) Real-time Metrics API enables you to obtain the latest, real-time statistical data on queues and agents queue interactions.

• Contact Center Customer Experience (CE) enables you to obtain historical call interaction data that you can use for further insights and solution development.

• A Virtual Contact Center Post Call Survey (PCS) is a questionnaire presented to the caller at the end of a call, typically to collect feedback about the quality of service offered by the company. Companies use a survey to gather feedback from customers and then analyze the collected data to help improve their services. For example, you can get feedback on overall customer satisfaction, how well your service agent performed, and how many calls it took to resolve issues.

Here we can see an example of metadata we will have access to.


```
"key": "callId",
"value": "int-17a826d24f2-XEwvo7NA7jjyqvXBilH1c1IL7-phone-00-kaptecnfr01"
},
{
"key": "ipbxid",
"value": "kaptecnfr"
},
{
"key": "result",
"value": "ok"
},
{
"key": "address",
"value": "+12157945422"

"value": [redacted]
},
{
"key": "direction",
"value": "INBOUND"
},
{
"key": "queueName",
"value": "Innovo_US_IB"
},
{
"key": "startTime",
"value": "1625685839597"
},
{
"key": "calleeName",
"value": "[redacted]
},
{
"key": "callerName",
"value": "+12157945422"
},
{
"key": "channelName",
"value": "353949051442"
},
{
"key": "queueNumber",
"value": "229"
},
{
"key": "holdDuration",
"value": "0"
},
{
"key": "callSnippetId",
"value": ""
},
{
"key": "transactionId",
"value": "477350"

{
"key": "agentId",
"value": "agg3NK7ZzfTzOxCuDDhe5xWw"
},
{
"key": "branchId",
"value": "e0TSN_SiTjGoOS5myQNaPA"
},
{
"key": "calleeId",
"value": "353949051442"
},
{
"key": "callerId",
"value": "2157945422"
},
{
"key": "duration",
"value": "219"
},
{
"key": "language",
"value": "en-GB"
},
{
"key": "mediaUrl",
"value": "R202107071930570002.wav"
},
{
"key": "provider",
"value": "voci"
},
{
"key": "sourceId",
"value": "258c194a-fa15-454c-bd23-b3a251162278"
},
{
"key": "tenantId",
"value": "kaptecnfr01"
},
{
"key": "agentName",

{
"key": "extensionNumber",
"value": "60002"
},
{
"key": "sourceObjectType",
"value": "callcenterrecording"
},
{
"key": "billingTelephoneNumber",
"value": "353949288194"
}
],
"shared": false
},
{

],
"channels": 2,
"provider": "voci",
"language": "en-GB",
"tags": {
"sentiment": "Mostly Positive",
"emotion": null,
"confidence": "0.86",
"recvtz": "UTC,0",
"model": "eng3:callcenter",
"donedate": "2021-07-07 19:34:54.583659",
"recvdate": "2021-07-07 19:34:50.831849"
},
"duration": 219
}
```


Figure 7 Data

2.3. DATA PREPARATION

Our dataset consists of 189 rows of data. These variables are: createdTime, updatedTime, objectState, result, agentId, branchId, callerId, duration, language, agentName, direction, queueName, queueNumber, sentiment, emotion, confidence, donedate and transcription.

From these data we will delete agentName because of GDPR. And we will be only using the agentID as a distinction. From the rest of the data, we will use only: createdTime, agentID, callerID, duration, direction, queueName, sentiment, emotion, confidence, transcription.

Then, we will continue by changing the transcriptions into corpus.

```r
```{r}
#Changing to paragraphs
corp = corpus_reshape(transcript_df1, to ="paragraphs")
#Deleting words with one or two letters
corp <- rm_nchar_words(corp, "1,2")
#Changing to low letters
corp <- tolower(corp)
#Removing numbers
corp <- removeNumbers(corp)
#Removing punctuation
corp <- removePunctuation(corp)
#Removing words
corp <- removeWords(corp, c("yeah","can","now","ill","like","ive","okay","dont","one","just","will","thank","ge
t","bye","know","thats","give","let","youre","put","see","back","much","say","see","yes","please","said","right
","fine","think","well","five","sure","sorry","mean","want","dot","cant","theres","gonna","actually","even","so
mething","perfect","thanks","kind","great","good","really","take","try","got","use","theyre","call","calling","
email","name","address","number","whats","bit","alright","thing","come","make","cause","able","little"))
#Stemming words
#Removing english stopwords
dfm = dfm(corp, remove_punct = T, remove=stopwords_en, stem=T)
#Using only words which occur at least 5 times
dfm = dfm_trim(dfm,min_docfreq = 5)
dfm
```
```

```
Document-feature matrix of: 189 documents, 814 features (91.8% sparse).
       features
docs    leav messag finish press hang though system run want place
  text1    1      1      1     1    1      1      1   1    1     1
  text2    0      1      1     1    1      0      1   0    0     0
  text3    0      0      0     0    0      0      0   0    0     1
  text4    0      0      0     0    0      0      0   0    0     0
  text5    0      0      0     0    0      0      0   0    0     0
  text6    0      2      0     0    0      0      1   0    0     0
[ reached max_ndoc ... 183 more documents, reached max_nfeat ... 804 more features ]
```

Figure 8 Corpus

The next step was removing words which has less than 2 letters, changing all words to lower case, removing numbers from the text, removing punctuation and removing stop words. In addition to english stop words, we also deleted some words which are not usable for our conversations. The last part was stemming and also using only words which occur at least 5 times.

2.4. PRELIMINARY ANALYSIS

After using LDA, we decided to split the transcripts to 6 different topics. Here we can see the most common words for each topic. From these graphs we can already see what a content of each topic is. For example, in topic 1 the customer is probably talking about problem with broadline in his house, in topic 2 the agent is probably trying to contact the customer and is asking about the time when the customer is available. In topic 3 they are talking about an order which was sent to the customer. In topic 4 there is an issue with telecom account and the bill. In topic 5 there is some problem with phone account which has to be paid. In topic 6 it is clear this conversation is about starting a new job.

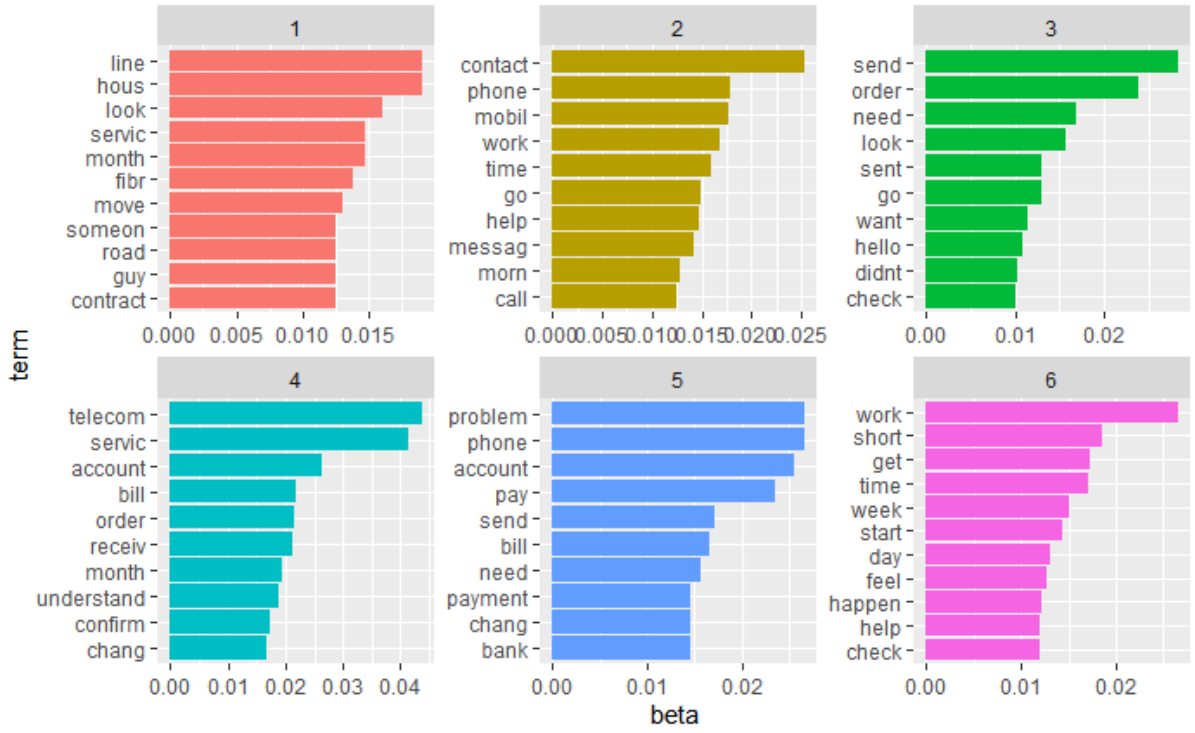


Figure 9 Most common words

Here we can see which topics are most associated with which transcription and how much.

TOPIC 1:

```
  text177   text179    text87    text67    text86   text167
0.7702991 0.6645480 0.6061198 0.5937363 0.5588235 0.5385675
'data.frame':   182 obs. of  1 variable:
 $ topic.docs: num  0.77 0.665 0.606 0.594 0.559 ...
```

TOPIC 2:

```
  text105   text153   text121   text135    text74    text45
0.9841772 0.9609375 0.9578652 0.9264706 0.9060150 0.8863636
'data.frame':   182 obs. of  1 variable:
 $ topic.docs: num  0.984 0.961 0.958 0.926 0.906 ...
```

TOPIC 3:

```
   text136     text8   text133    text47    text32    text31
 0.9647887 0.9539877 0.8197115 0.8076923 0.7844828 0.7813688
 'data.frame':   182 obs. of  1 variable:
  $ topic.docs: num  0.965 0.954 0.82 0.808 0.784 ...
```

TOPIC 4:

```
    text97   text129    text91    text33     text3   text151
 0.9871531 0.9826389 0.9626538 0.9582671 0.9481737 0.9210240
 'data.frame':   182 obs. of  1 variable:
  $ topic.docs: num  0.987 0.983 0.963 0.958 0.948 ...
```

TOPIC 5:

```
    text51    text13   text118   text110   text134   text117
 0.7067594 0.6859606 0.6133094 0.5802239 0.5793269 0.5777027
 'data.frame':   182 obs. of  1 variable:
  $ topic.docs: num  0.707 0.686 0.613 0.58 0.579 ...
```

TOPIC 6:

```
  text132    text21   text116   text119   text162   text141
0.9825175 0.8056995 0.7979972 0.7794118 0.7579225 0.7373394
'data.frame':   182 obs. of  1 variable:
 $ topic.docs: num  0.983 0.806 0.798 0.779 0.758 ...
```

Here we created a new dataframe, which contains numbers of topics for each conversation.

transcriptID <chr>	Topic <int>	createdTime <chr>	agentID <chr>	callerID <chr>	duration <int>
text1	3	2021-08-11T14:16:26	agvd2avJDPQCiBQ5k6Ure9Xw	3.54E+11	54
text2	3	2021-08-27T16:23:23	agvMZOeLXxSim3Mu1ElsQPwQ	4.48E+11	40
text3	4	2021-07-16T10:23:21	ageou4J6UdQra9K3RmEHsO8Q	35316855060	445
text4	2	2021-07-31T15:23:32	agT0A9fcYdTZeqbrthTv7uug	35317116000	180
text5	2	2021-08-10T14:34:31	ageou4J6UdQra9K3RmEHsO8Q	3.54E+11	132
text6	2	2021-07-13T09:06:27	ag61C_7yTaQNaKDRnAXxt3vw	3.54E+11	333
text7	2	2021-08-06T15:06:24	ageou4J6UdQra9K3RmEHsO8Q	3.54E+11	51
text8	3	2021-07-30T12:43:58	agfUU69IIcTlCOHKBcPJ5qUA	4.41E+11	219
text9	2	2021-08-24T10:13:28	ageou4J6UdQra9K3RmEHsO8Q	4.48E+11	91
text10	2	2021-08-22T11:54:36	agP3MWo1GpQUKLaFW0NeILBg	4.48E+11	234

1-10 of 182 rows | 1-6 of 12 columns Previous 1 2 3 4 5 6 ... 19 Next

Figure 10 Numbers of topic

4. THE SOLUTION

4.1. REFINED PROJECT OBJECTIVES

Then we decided to evaluate agents' performance by comparing their confidence level for each topic. So, we calculated mean performance for each agent in each topic. We also changed agentID to Agent1, Agent2, to better plot the results.

```r
data2.df <-data.df %>%
  group_by(agentID,Topic)%>%
  summarise(n= n(),confidence)
data2.df$confidence <- as.double(data2.df$confidence)
data2.df$Topic <- as.factor(data2.df$Topic)
data2.df

data2.df <-data2.df %>%
  group_by(agentID, Topic)%>%
  summarise(mean_confidence = mean(confidence))

data2.df$mean_confidence <- round(data2.df$mean_confidence,2)
data2.df$agentID[data2.df$agentID == "ag61C_7yTaQNaKDRnAXxt3vw"] <- "Agent1"
data2.df$agentID[data2.df$agentID == "agAuZjTmR_QvOZ6OMC5U8YhQ"] <- "Agent2"
data2.df$agentID[data2.df$agentID == "agBev5uNJDR924_gTsbOccQA"] <- "Agent3"
data2.df$agentID[data2.df$agentID == "agBjCh9b2sSyKT2dfEW_Boxw"] <- "Agent4"
data2.df$agentID[data2.df$agentID == "agDhT5X7vISoWBc1ZHcVbLiw"] <- "Agent5"
data2.df$agentID[data2.df$agentID == "ageou4J6UdQra9K3RmEHsO8Q"] <- "Agent6"
data2.df$agentID[data2.df$agentID == "agfauaSKCEQYqoTR1e59GNfg"] <- "Agent7"
data2.df$agentID[data2.df$agentID == "agfUU69IIcTlGOHKBcPJ5qUA"] <- "Agent8"
data2.df$agentID[data2.df$agentID == "agg3NK7ZzfTzOxCuDDhe5xWw"] <- "Agent9"
data2.df$agentID[data2.df$agentID == "agiEJ2aFsuRtOwgxCxMY7i5Q"] <- "Agent10"
data2.df$agentID[data2.df$agentID == "agnD1EfPfzQ46hsngZ4yeOOQ"] <- "Agent11"
data2.df$agentID[data2.df$agentID == "agNL1__IFCQdKOZ4Jsn25bgw"] <- "Agent12"
data2.df$agentID[data2.df$agentID == "agP3MWo1GpQUKLaFWONeILBg"] <- "Agent13"
data2.df$agentID[data2.df$agentID == "agqGZnfD_dQeS2PyoT193rOw"] <- "Agent14"
data2.df$agentID[data2.df$agentID == "agTOA9fcYdTZeqbrthTv7uug"] <- "Agent15"
data2.df$agentID[data2.df$agentID == "aguylEguURQ7OP2mA1GTp5qg"] <- "Agent16"
data2.df$agentID[data2.df$agentID == "agvd2avJDPQCiBQ5k6Ure9XW"] <- "Agent17"
data2.df$agentID[data2.df$agentID == "agvMZOeLXxSim3Mu1ElsQPwQ"] <- "Agent18"
data2.df$agentID[data2.df$agentID == "agVUInCdLRR4WX4gOrjAEuvw"] <- "Agent19"
```

Figure 11 Agents Numbers

agentID <chr>	Topic <fctr>	mean_confidence <dbl>
Agent1	1	0.87
Agent1	2	0.86
Agent1	4	0.86
Agent1	5	0.84
Agent1	6	0.89
Agent2	2	0.92
Agent2	3	0.82
Agent2	6	0.87
Agent3	2	0.92
Agent3	3	0.90

Figure 12 Result dataset

Here we can see how our result dataset looks like.

To plot the results, we decided to split the dataset into 6 different ones, one for each topic. Here is an example for topic 1.

```r
data3.df <- data.frame(data2.df$agentID, data2.df$Topic, data2.df$mean_confidence)
data3.df <- data3.df %>%
  rename(agentID = data2.df.agentID, Topic = data2.df.Topic, mean_confidence=data2.df.mean_confidence)
data3.df <- filter(data3.df, Topic == "1")
data3.df
```

agentID <chr>	Topic <fctr>	mean_confidence <dbl>
Agent1	1	0.87
Agent10	1	0.85
Agent11	1	0.86
Agent13	1	0.86
Agent14	1	0.85
Agent15	1	0.87
Agent16	1	0.86

Figure 13 Splitted dataset

For getting better visualisation results we changed agents' IDs to Agent1 – Agent19 nicknames:

ag61C_7yTaQNaKDRnAXxt3vw<- "Agent1"

agAuZjTmR_QvOZ6OMC5U8YhQ<- "Agent2"

agBev5uNJDR924_gTsbOccQA<- "Agent3"

agBjCh9b2sSyKT2dfEW_Boxw<- "Agent4"

agDhT5X7VISoWBc1ZHcVbLiw<- "Agent5"

ageou4J6UdQra9K3RmEHsO8Q<- "Agent6"

agfauaSKCEQYqoTR1e59GNfg <- "Agent7"

agfUU69IIcTlGOHKBcPJ5qUA<- "Agent8"

agg3NK7ZzfTzOxCuDDhe5xWw<- "Agent9"

agiEJ2aFsuRtOwgxCxMY7i5Q<- "Agent10"

agnD1EfPfzQ46hsngZ4yeO0Q<- "Agent11"

agNL1__IFCQdKOZ4Jsn25bgw<- "Agent12"

agP3MWo1GpQUKLaFW0NeILBg <- "Agent13"

agqGZnfD_dQeS2PyoT193rOw <- "Agent14"

agT0A9fcYdTZeqbrthTv7uug <- "Agent15"

agUylEguURQ7OP2mA1GTp5qg <- "Agent16"

agvd2avJDPQCiBQ5k6Ure9Xw <- "Agent17"

agvMZOeLXxSim3Mu1ElsQPwQ <- "Agent18"

agVUInCdLRR4WX4g0rjAEuvw <- "Agent19"

Here are our results:

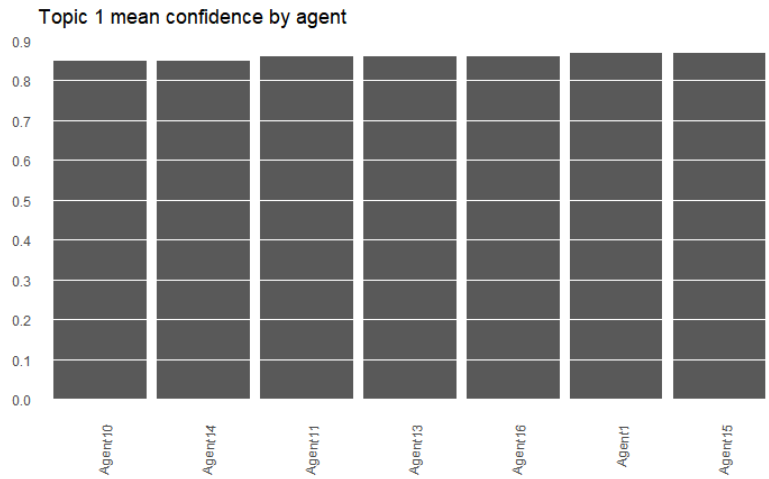


Figure 14 Topic 1

From looking at this first graph, we can see that Agent 15 has the highest confidence score in Topic n.1. But the scores are very close to each other without any big differencies.

For topic 2 the best performing agent is Agent14 with the confidence score nearly 90%.

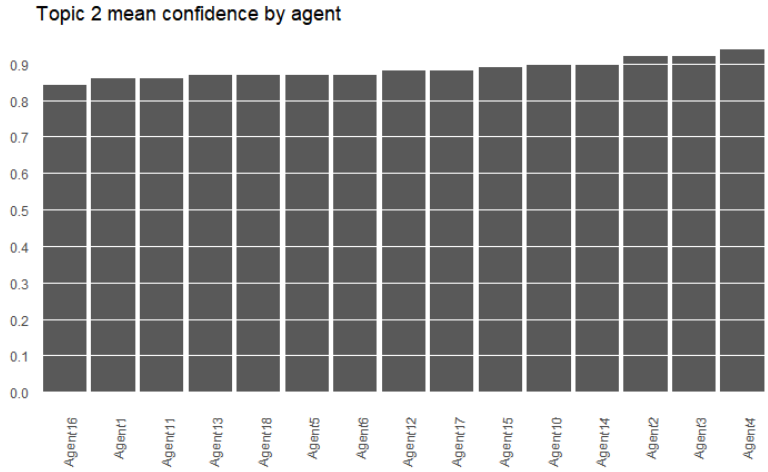


Figure 15 Topic 2

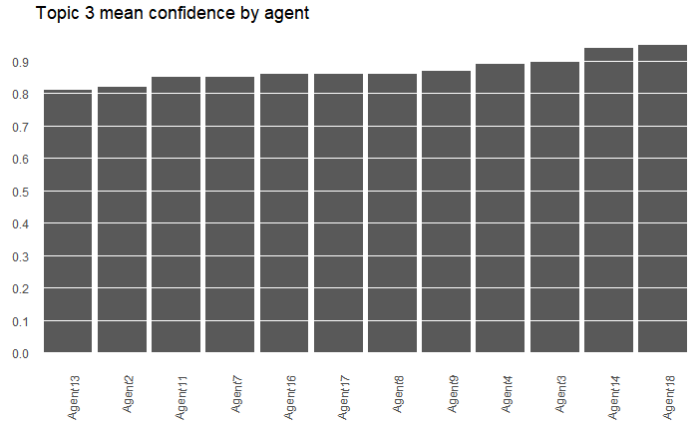


Figure 16 Topic 3

For topic 3, the best performing agent is Agent 16 and Agent 14 again. The worst performing agent is Agent 13.

For topic 4, the scores are also very close to each other. But the best performing agent is Agent15.

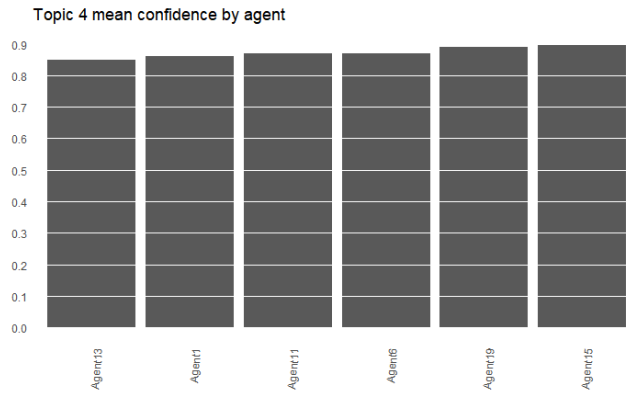


Figure 17 Topic 4

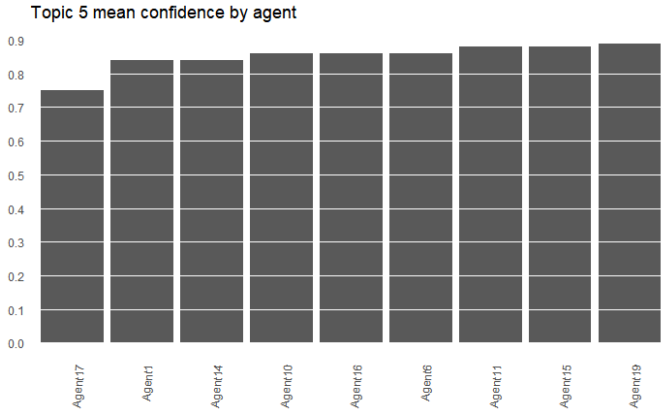


Figure 18 Topic 5

Here we can see the results for Topic 5. The best performing agent is Agent 19 and the worst performing agent is Agent 17.

For the last topic the best performing agent is Agent 18 and the worst performing agent is agent 7.

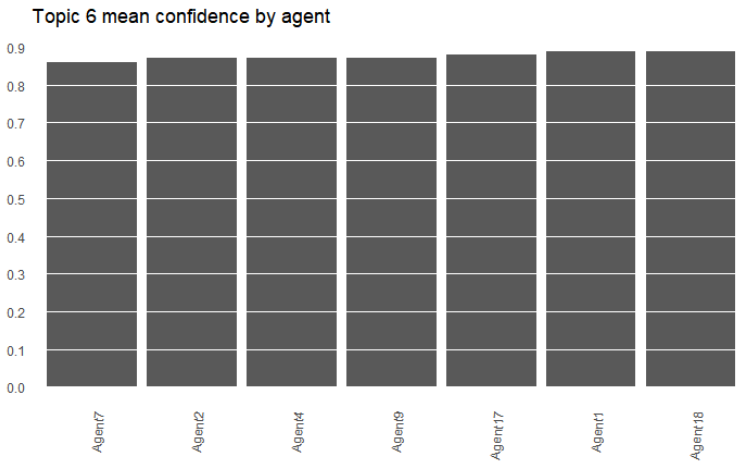


Figure 19 Topic 6

SENTIMENT ANALYSIS

Then we continued with a sentiment analysis. The dataframe already contains sentiment analysis, but we decided to try it anyway so we could compare our results with those already created by 8x8.

Firstly, we started by creating a corpus, cleaning the text from punctuation, numbers and stopwords, and also by changing all words to lower case. Then we created a document term matrix. And then we plotted the most common words contained in our transcriptions.

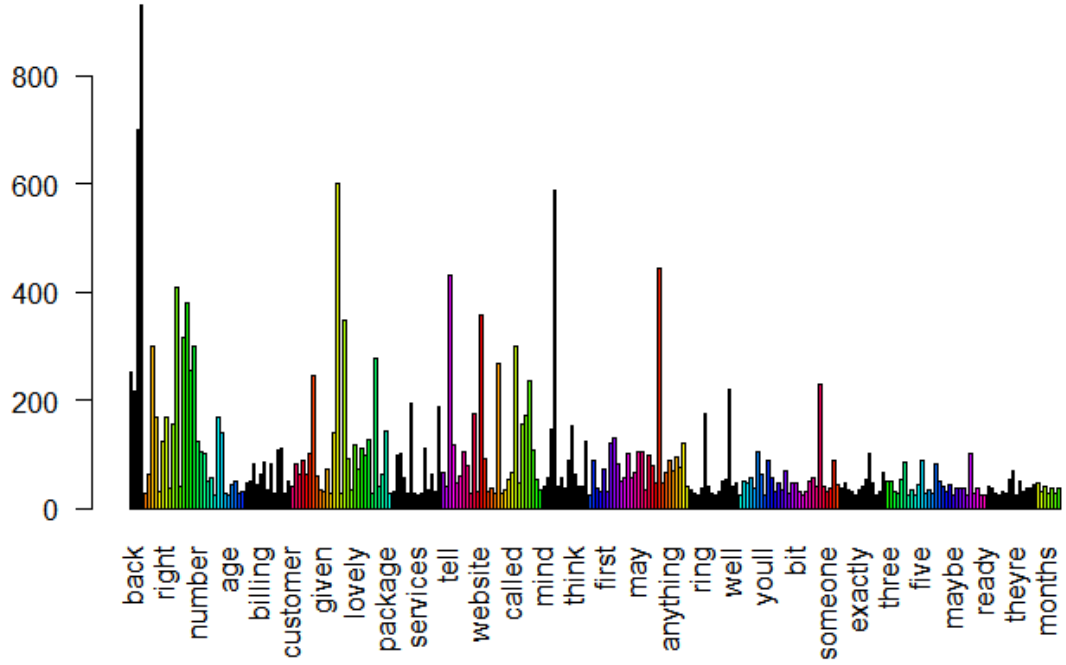


Figure 20 Most Common Words

Here we can see that the most common words in our transcripts are: back, given, think, anything, someone, etc…

Then we created a wordcloud from these words, containing maximum 150 words with those occuring at least 5 times.

Figure 21 Wordcloud

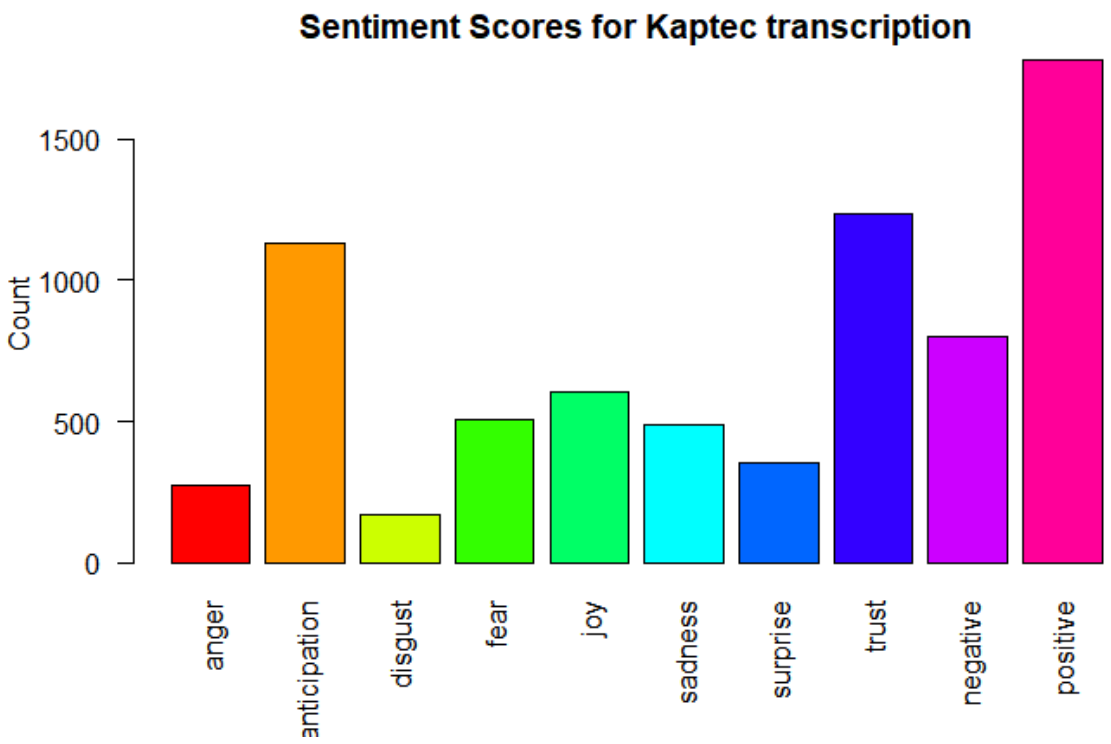


Figure 22 Sentiment

Here we can see a bar plot from most frequent sentiments occuring in Kaptec call center transcripts. The most common sentiment is positive, the least common sentiment is disgust and surprise. Which is a very good result.

```r
```{r}
#Creating dfm
transcript_df2 %>%
  corpus(text_field = "transcription")%>%
  dfm()
#Converting to GI_dict
GI_dict = dictionary(DictionaryGI)

result = dfm %>%
  dfm_lookup(GI_dict) %>%
  convert(to="data.frame")%>%
  as_tibble

result
```
```

R Console

tbl_df
189 x 3

doc_id <chr>	negative <dbl>	positive <dbl>
text1	3	1
text2	2	2
text3	14	29
text4	8	9
text5	5	7
text6	4	6
text7	1	4
text8	5	5
text9	2	1
text10	14	10

1-10 of 189 rows

Previous 1 2 3 4 5 6 ... 19 Next

Figure 23 Sentiment 2

As the last step of our sentiment analysis, we created a data drame containing negative or positive sentiment for each transcript.

To evaluate our sentiment analysis, we decided to test it on 30 different transcripts. The numbers of transcripts were picked randomly. The numbers of transcripts which we tested are:
3,7,12,15,22,25,35,52,58,60,62,75,79,82,94,101,111,115,119,120,125,135,137,138,140,142, 150,155,166,180.

The results are visible in table below.

Transcript number	8x8 analysis	Our sentiment analysis
3	Positive	Positive
7	Positive	Positive
12	Positive	Positive
15	Positive	Positive
22	Positive	Positive
25	Neutral	Neutral
35	Mixed	Positive
52	Neutral	Positive
58	Negative	Positive
60	Positive	Negative
62	Neutral	Neutral
75	Positive	Positive
79	Positive	Negative
82	Negative	Negative
94	Negative	Negative
101	Mostly Positive	Positive
111	Positive	Positive
115	Positive	Positive
119	Positive	Negative
120	Positive	Negative
125	Mostly Negative	Positive
135	Positive	Negative
137	Negative	Positive
138	Positive	Positive
140	Positive	Positive
142	Positive	Negative
150	Neutral	Neutral
155	Mixed	Mixed
166	Positive	Positive
180	Mostly Positive	Positive

Table 1 Sentiment Results

From this table we can see that our sentiment analysis assigned most of the transcription correctly in this sample of 30 transcripts. Our successful score is = 20/30 = 66%. So, we assigned 66% of the sentiments correctly based on 8x8 evaluation.

The last part of our project is making predictions about sentiment. We used confidence, queueNumber and duration of the call for making predictions. After looking at the data, we decided to use two methods. One of them was Naïve Bayes and the second one was Decision trees.

NAÏVE BAYES

Firstly, we started by splitting the data into training and testing datasets with 80:20 ratio. Then we changed the sentiment to factor and created a model.

```
============================================= Naive Bayes =============================================

 Call:
naive_bayes.formula(formula = sentiment ~ ., data = train)

-------------------------------------------------------------------------------------------------------

Laplace smoothing: 0

-------------------------------------------------------------------------------------------------------

 A priori probabilities:

       -1          0          1
0.37500000 0.05882353 0.56617647

-------------------------------------------------------------------------------------------------------

 Tables:

-------------------------------------------------------------------------------------------------------
 ::: duration (Gaussian)
-------------------------------------------------------------------------------------------------------

duration        -1         0         1
    mean 273.90196  36.62500 286.46753
    sd   206.38297  26.33812 219.71378

-------------------------------------------------------------------------------------------------------
 ::: queueNumber (Gaussian)
-------------------------------------------------------------------------------------------------------

queueNumber         -1          0          1
       mean 226.254902 237.625000 226.480519
       sd     7.421167   8.348439   9.332471

-------------------------------------------------------------------------------------------------------
 ::: confidence (Gaussian)
-------------------------------------------------------------------------------------------------------

confidence         -1          0          1
      mean 0.86039216 0.83250000 0.86935065
      sd   0.03237041 0.17417151 0.02876116

-------------------------------------------------------------------------------------------------------
```

Figure 24 Naive Bayes 1

This model shows us that in the training data we have about 37.5% of the data belonging to -1 sentiment. Around 6% of data belonging to 0 sentiment and 57% of data belonging to 1 sentiment. We can also see all calculated means and standard deviations for all categorical variables.

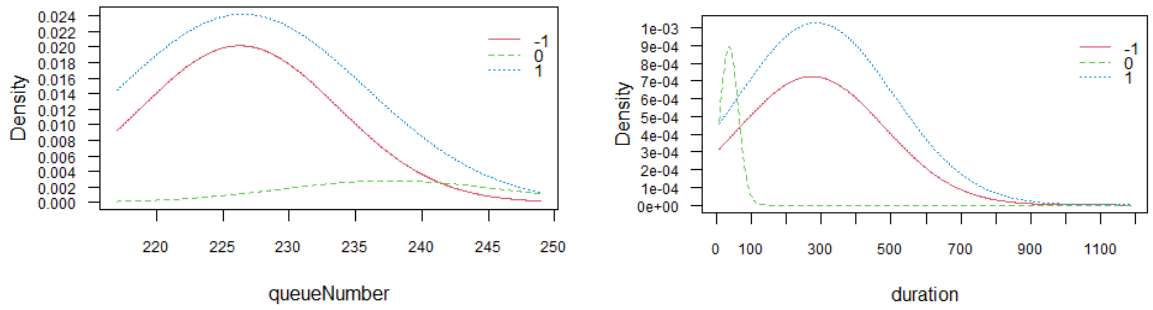


Table 2 Naive Bayes 2

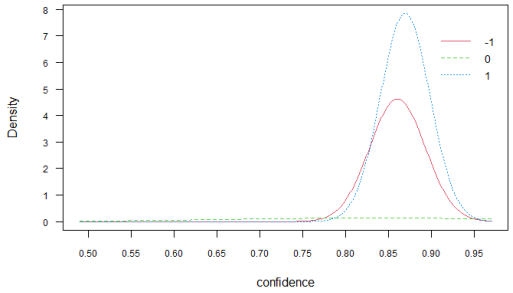


Figure 25 Naive Bayes 2

for duration and one for confidence.

Then we continued by creating predictions.

	-1 <dbl>	0 <dbl>	1 <dbl>	duration <int>	queueNumber <int>	sentiment <fctr>	confidence <dbl>
1	0.0486254	5.995343e-01	0.3518403	54	249	-1	0.86
2	0.1489653	6.774118e-01	0.1736229	40	225	0	0.95
3	0.3609253	7.729946e-55	0.6390747	445	218	1	0.87
4	0.3437235	4.211346e-09	0.6562764	180	217	-1	0.89
6	0.4063500	4.957514e-30	0.5936500	333	221	-1	0.87
7	0.3871160	2.873165e-02	0.5841523	51	222	1	0.88

6 rows

Figure 26 Naive Bayes 3

From this table of results, we can see that first transcript has a probability of 4.8% probability to be sentiment -1, 60% chance to be sentiment 0 and 35% chance to be sentiment 1. Second transcript has a probability of 15% to be sentiment -1, 68% probability to be sentiment 0 and 17% probability to be sentiment 1.

The next step was creating a confusion matrix for training and testing data. And also calculating the misclassification rate.

```
predict.naive_b
object. Calcula
p1    -1  0  1
  -1   9  0  6
  0    2  6  6
  1   40  2 65
[1] 0.4117647
```

```
object. Calculatic
p2    -1  0  1
  -1   1  0  1
  0    0  0  1
  1    4  1 22
[1] 0.2333333
```


Figure 27 Naive Bayes 4

For training set, we can see that 9 transcripts were correctly predicted to be sentiment -1, 6 were correctly predicted to be sentiment 0 and 65 were correctly predicted to be sentiment 1. The misclassification rate is around 41%.

For testing set, we can see that 1 transcript was correctly predicted to be sentiment -1, 0 were correctly predicted to be sentiment 0 and 22 were correctly predicted to be sentiment 1. The misclassification rate is around 23%.

This missclassification rate is not very good.

DECISION TREES

The next supervised learning method which we used was Decision trees.

Here we also started by setting sentiment as a factor and splitting the dataframe into training and testing data by using ratio 80:20.

```
        Conditional inference tree with 3 terminal nodes

Response:  sentiment
Inputs:  duration, queueNumber, confidence
Number of observations:  136

1) queueNumber <= 242; criterion = 0.992, statistic = 11.884
  2) confidence <= 0.89; criterion = 1, statistic = 21.5
    3)*  weights = 100
  2) confidence > 0.89
    4)*  weights = 16
1) queueNumber > 242
  5)*  weights = 20
```

Figure 28 Decision Trees 1

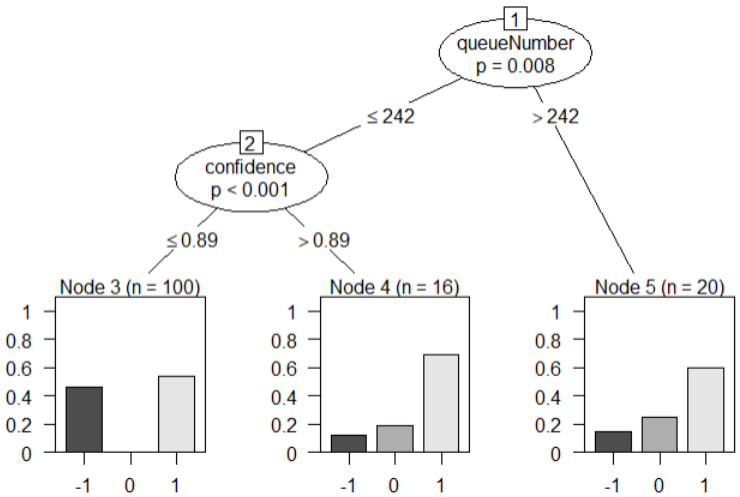


Figure 29 Decision Trees 2

We can see there is 3 terminal nodes in this tree. We can see that queueNumber is the most important variable from all thr/ee. So, for example if queueNumber is less than 242 we go on the left site, if confidence is bigger than 0.89, we go on the right side to the terminal node, where the biggest probability is that the sentiment is equal to 1, where the probability is equal to 70%.

For all 30 transcripts in the validate data set the model gives a prediction whether they belong to sentiment -1 0 or 1. For all 30 transcripts the prediction is that all sentiment belongs to 1.

```
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Levels: -1 0 1
```

Figure 30 Decision Trees 3

Here we again created a misclassification matrix for train and test data and calculated misclassification error.

For training data 0 transcripts were correctly predicted to be sentiment -1, 0 were correctly predicted to be sentiment 0 and 77 were correctly predicted to be sentiment 1. The missclassification error is 43% based on the training data.

For testing data 0 transcripts were correctly predicted to be sentiment -1, 0 were correctly predicted to be sentiment 0 and 24 were correctly predicted to be sentiment 1 out of 30 transcripts. The missclassification error is 20% based on the testing data.

```
     -1  0  1
  -1  0  0  0
  0   0  0  0
  1  51  8 77
[1] 0.4338235
```

```
testPred -1  0  1
      -1  0  0  0
      0   0  0  0
      1   5  1 24
[1] 0.2
```

Figure 31 Decision trees 4

4.2. ANALYSIS AND VISUALISATION METHODS EMPLOYED

In this project we used different analysis to obtain our results.

1. TOPIC MODELLING - The biggest part of our project was creating a topic modeling algorithm using LDA – Latent Dirichlet allocation method. Using this algorithm, we then found the most common topics discussed in the call center transcripts.
2. SENTIMENT ANALYSIS - When we split our conversations into different topics, we then performed a sentiment analysis. From different words used in our conversations we infered whether a part of text is positive or negative
3. COMPARING OF AGENTS SCORES - The next part of our project was analysing a performance of each agent based on the different conversation topic. We found the topics in which each agent excels or on the other side needs more training to gain better results.
4. PREDICTIONS USING NAÏVE BAYES - The last part of our project was predicting sentiment of each topic using different predicting classification. We used queueNumber, confidence and duration of the call. We used Naïve Bayes method for this.
5. PREDICTIONS USING DECISION TREES – We also used decision trees for our predictions.

For the visualisation we used:

1. BAR CHARTS – To represent different words belonging to each topic, to represent performance of agents for each topic, also to visualise sentiments
2. WORD CLOUD – We used wordcloud to visualise most common words in those transcripts
3. DENSITY PLOTS – We used them to represent naïve bayes.
4. DECISION TREES
5. TABLES

4.3. WORK BREAKDOWN AND PLANNING

Phases:

1. Analysis of the clients' satisfaction in Kaptec company.

2. Analysis of the effiency of agents over time from Kaptec company.

Tasks:

Steps	Tasks	Deadlines
1	Analysing the 8x8 system and obtaining the data through API.	25th April
2	Data preprocessing and cleaning the data	10th May
3	Preparing Interim Report	15th May
4	Building an algorithm with R studio for data analysis	30th June
5	Data visualisation of the results from data analysis	30th July
6	Preparing Final Report	15th August

Below is the task planner over the project duration:

Months Steps	April	May	June	July	August
Step 1					
Step 2					
Step 3					
Step 4					
Step 5					
Step 6					

6. CONCLUSIONS

6.1. EVALUATION OF THE PROJECT AS A WHOLE

Call center data is very important for helping to identify strengths and weaknesses in the contanct center operation. With the right data and right analytics, you can uncover patterns and highlight trends and make more educated business decisions based on real-time insight.

Within this project analysis were conducted of the contact center call transcripts. The data were provided to us in csv file containing different variables like: createdTime, updatedTime, objectState, result, agentId, branchId, callerId, duration, language, agentName, direction, queueName, queueNumber, sentiment, emotion, confidence, donedate and transcription. From these variables we decided to use for our analysis the agentId, duration, queueNumber sentiment emotion, confidence and transcription. By opening this file in R studio different analysis were completed. The first part of our project consisted of creating a topic modeling algorithm using latent dirichlet allocation. Firstly, we started by splitting the transcripts into separate words. Then we created 6 different topics occuring in our transcripts. The next step was to compare effectivity of agents in those 6 topics, based on their confidence score. It would be easier for us if we got a supervisor's evaluation score for each conversation, but we had to work with a data we got. The next step was a sentiment analysis, where we assigned each conversation a sentiment. We found that most occuring sentiments in these conversations were positive, trust and anticipation. We then used 8x8 sentiment analysis to evaluate our results. Our sentiment analysis was fairly good, we got a score of 66%. The last part of our project was to predict sentiment of each conversation. After a long decision, we decided to use Naïve Bayes and Decision trees. The misclassification rate for testing data in Naïve bayes was 23%, while for decision trees it was 20%. Which means, our predictions weren't very accurate. It would be more useful if our sample was larger, or we could get different variables which are more related to this topic.

6.2. LESSONS LEARNT

While working on this project, there were a lot of lessons which we learnt. The first part was to prepare and clean the data mostly for the topic modeling and sentiment analysis. Which took us quite a long time. It gave us a lot of experience of how to work with a big data. We learnt a lot of different programming skills and we also improved them. The visualisation part of the project was also a big challenge, to understand what the best graph or visualisation for each analysis is, to present given results in the best possible way. We also learnt what numerous packages which we never used before in R studio do.

6.3. POTENTIAL FUTURE WORK

This project could be a starting point to another project. By obtaining a bigger sample or using different variables we could come to more accurate results.

BIBLIOGRAPHY

[1] Towars Data Science. (2019, Feb. 4). *A comprehensive guide to to data visualisation in R for beginners* [Online]. Available: https://towardsdatascience.com/a-guide-to-data-visualisation-in-r-forbeginners-ef6d41a34174

[2] Flipboard. (2017, Feb. 8). *Clustering Similar Stories Using LDA* [Online]. Available: https://engineering.flipboard.com/2017/02/storyclustering

[3] Github. (2020, Oct. 8). *Tutorial 6: Topic Models* [Online]. Available: https://tm4ss.github.io/docs/Tutorial_6_Topic_Models.html

[4] Tidy Text Mining. *6 Topic Modeling* [Online]. Available: https://www.tidytextmining.com/topicmodeling.html

[5] Tidy Text Mining*. 2 Sentiment Analysis with tidy data* [Online]. Available: https://www.tidytextmining.com/sentiment.html

[6] Geeks for Geeks. (2020, May 15.). *Naïve Bayes Classifier*. [Online]. Available: https://www.geeksforgeeks.org/naive-bayes-classifiers/

[7] Scikit Learn. *Decision Trees.* [Online]. Available:

https://scikit-learn.org/stable/modules/tree.html

[8] 8x8. (2021, Jun. 28). *8x8 Contact center APIs* [Online]. Available: https://support.8x8.com/cloud-contact-center/virtual-contact-center/developers/virtual-contact-center-apis

[9] Kaptec [Online]. Available: https://www.kaptec.com/

[10] 8x8 [Online]. Available: https://www.8x8.com/

Figure 1 Latent Dirichlet Method (image source: https://www.sciencedirect.com/science/article/abs/pii/S095741741930154X)

Figure 2 Sentiment Analysis (image source: https://www.tidytextmining.com/sentiment.html)

Figure 3 Naïve Bayes (image source: https://towardsdatascience.com/introduction-to-na%C3%AFve-bayes-classifier-fa59e3e24aaf)

Figure 4 Decision Trees (image source: https://techvidvan.com/tutorials/decision-tree-in-r/)

Figure 5 8x8 screen 1 (image source: https://www.8x8.com/products/contact-center/analytics)

Figure 6 8x8 screen 2 (image source: https://www.8x8.com/products/contact-center/analytics)