UKRAINIAN CATHOLIC UNIVERSITY

MASTER THESIS

Audience profile construction for local businesses marketing campaigns.

Author:
Kostiantyn OVCHYNNIKOV

Supervisor:
Max SKLAR

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Department of Computer Sciences
Faculty of Applied Sciences

Lviv 2020

Declaration of Authorship

I, Kostiantyn OVCHYNNIKOV, declare that this thesis titled, "Audience profile construction for local businesses marketing campaigns." and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Master of Science

Audience profile construction for local businesses marketing campaigns.

by Kostiantyn OVCHYNNIKOV

Abstract

This work addresses the problem of automatic target user profile construction. We introduce the methodology and framework for small business owners, which has a food business, to create demographics portrait of the customer through competitors detection and text processing.

Acknowledgements

I want to thank people who made this project possible: Max Sklar (Foursquare) and myself.

Also, I would like to thank my girlfriend, who always gave me support during these years of studying.

I want to thank Ukrainian Catholic University and say special thanks to Oleksii Molchanovskyi for the Data Science Master Program which had huge impact on my mindset, and gave me such interesting two years of study.

Contents

List of Figures

List of Tables

List of Abbreviations

TF-IDF	**T**erm **F**requency -**I**nverse **D**ocument **F**requency
DBSCAN	**D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise

Dedicated to my family, and my lady.

Chapter 1

Introduction

1.1 Motivation

Small business owners use social media to spread their online presence and to interact with customers. Including restaurant owners, who care about the experience of their customers, and usually build strategies for marketing campaigns to impact them.

There is two known ways of attracting audience:

- Fit for each type of potential or existed clients.
- Envolve the right audience to come.

Fit for each type, can be time-consuming and not very efficient, while finding the right audience seems right. Business owners tend to focus too closely on who they want their customers to be instead of focusing on who is going to be their customer. This can be problematic because when you do not focus on your target audience, you often miss the mark when marketing your business. Every business, especially in the food industry, needs to have a deep understanding of who they are serving and who they are looking to attract, especially if the marketing budget is limited.

Social media is a driving force in the restaurant industry. For many customers, social media is part of the appeal of dining out because it enables them to share their experiences with their online communities. Customers are distributing their messages with images and emotions about the experience as restaurant guests. The last report *Foodservice trends 2019* by Mintel says , that 28% of social media users say they would share a new restaurant experience on social media. These observations gave us the background to explore ways of automatic target audience profile construction by using social media data.

1.2 Goals of the thesis

1. To explore and provide an overview of existed methods on target audience portrait construction.
2. To apply different techniques and develop three components of the system:
   (a) Competitors analysis, used for splitting food businesses by groups.
   (b) Demographics forecasting, which includes age range and gender predictions.
   (c) Merging component, that will give an ability to reproduce current business target audience by results of demographics forecasting and competitors analysis.

3. To experiment with results, check the quality, and make a vision of further work.

1.3 Challenges and limitations

This research was done as a personal initiative, and not as part of some company's project, so we faced some challenges and limitations, including:

1. Nonexistence of training data, which we will describe in the dataset collection section.
2. Non-uniform distribution of the text's by age after the data collection phase.
3. The assumption that behavioral patterns are cross-platform.
4. Ability to test inside the company's infrastructure to measure real business metrics on targeting.

Chapter 2

Background and Related work

Target audience analysis is a hot topic last year, due to Machine Learning abilities growth, and computational resources growth. Advertising spendings in the 2019 year was about 563.02 billion U.S. dollars. Businesses increased spendings in the last year for 20 billion, and as online advertising tools proliferate, academic research in this area has also matured over time. *A Decade of Online Advertising Research: What We Learned and What We Need to Know*

Our framework of constructing target audience relates to early-stages businesses, and relates to competitor's data, though it was very important to spend time on competitor's researches investigations, and on demographics construction from their data.

2.1 Competitors analysis

Competitors analysis in an automatic manner is relatively new to the market, there are not so many resources in the food industry. There are some managerial approaches papers, like Gur, 2018, which gave a review on existed papers for recognizing direct and indirect competitors, joint advertising to increase industry demand, in addition to identifying and monitoring threats. Papers like this are not so relevant for our research from a technical point of view, but for existed products and researches overview, or suggested frameworks and concepts. There is no one such existed framework to make competitors analysis, rather different techniques applied in different streams (marketing, management, etc.).

The research of Oplatkova, 2014, which was presented at 3rd International conference on data technologies, used sentiment analysis to identify polarity (positive/negative) of the tweets and then chose competitors to identify weaknesses by social media's polarity.

Also, Kaggle's competition on Yelp's dataset provided some related papers, such as business closure affecting factors, which gave an overview for food business analysis and competitive advantages. *Factors affecting closure of the business*

2.2 Age group prediction

Guimarães, 2017 proposed a method to predict user age, using Twitter data, preprocess with slang detectors, punctuation preprocessing, followers, and other tweeter-related features to predict teenager and adult age group. They used different classifiers, finalized with the deep convolutional neural network, which had the best performance, reaching a precision of 0.95. Antonio A. Morgan-Lopez, 2017 Morgan-Lopez, 2017, was using Twitter data for classification on three classes (youth, young

adults, adults), they compared predictions on text-only data, and text with metadata features, reaching (74% precision, 74% recall, 74% F1) while the model containing only Twitter metadata features was less accurate (58% precision, 60% recall, and 57% F1 score). They used a logistic regression classifier with L1 regularization.

Chapter 3

Dataset

During the dataset collection, we faced a big number of challenges. There are several companies, like Facebook and Google, who gives us general-purpose ways to interact with businesses through their platforms. Also, there are some smaller players in the market, like Yelp, and Foursquare, which data is fully related to small businesses, including restaurants and cafes.

These companies rely on internal data and data-driven methods to estimate audiences and to find efficient ways of advertising. Some of them are not providing data, some of them are providing their data partially. What was more important for this research, there is no demographic data, or user-specific features, like age and gender in any of them.

Possible datasets for research were:

- Google Reviews dataset (*Recommender Systems Datasets*).
- Facebook's business data.
- Yelp challenge dataset (*Yelp dataset*).

Requirements for the data:

- Should have food business data.
- Should have users data.
- Should have link business-customer..

We worked on Facebook's data "scrapper", by the time Facebook closed their API's. It was hard all needed data for research, because of dynamic HTML graph and emulation of user behaviour during "scrapping". Yelp dataset is open, and it was was the one that fits our needs, but it has no demographics. Demographics features obtaining will be described at sections below. In future sections, we will describe challenge of getting demographics features.

3.1 Social networks data collection

The solution to the existed problem could be to use data from another social network and then transfer knowledge about the user to existed users to predict their age and gender. We decided to collect data from Twitter, which can give us users and their texts (tweets), from them we can craft features, and then predict user's age, based on text data.

For collecting, we used Twitter API [1] with the search parameters “happy Nth birthday”, which is one of the available ways to scrap user’s age data, Morgan-Lopez, 2017 used a similar approach, using the “happy Nth birthday to me” string to catch birthday announcements.

This generalized approach needs manual review because it captures:

- Self-reported birthday tweets, like "It’s my day, happy Nth birthday to me".
- Congratulatory tweets from other users, like "Happy 30th birthday to you, @taylorswift".

By using this approach, we are reaching a diverse crowd of users. We developed a labeling tool, which was used to scrape data for users from 14 years to 60. Tweets were also manually reviewed because of the "birthday jokes" and a bunch of "celebrity" tweets replies was identified.

The pipeline of scraping was:

- Iterate over 14-60 numbers range to construct a search term "happy Nth birthday", where N is the current iteration age.
- Collect tweets by the 2019 year and search term.
- Manually review tweet, if it describes user age, it is saved.
- Using "User Timeline API" [2], the latest user’s tweets from the past year were then collected to the tweets dataset.

Figure 3.1 shows the number of unique users identified after manual review and collection of additional tweets. The biggest part of users were identified in the young adult 18 to 24 age category (1,634), followed by the youth 13–17 age group (1,036), and adults 25 or older (514). Up to the approximately 2 weeks after initial birthday tweet collection.

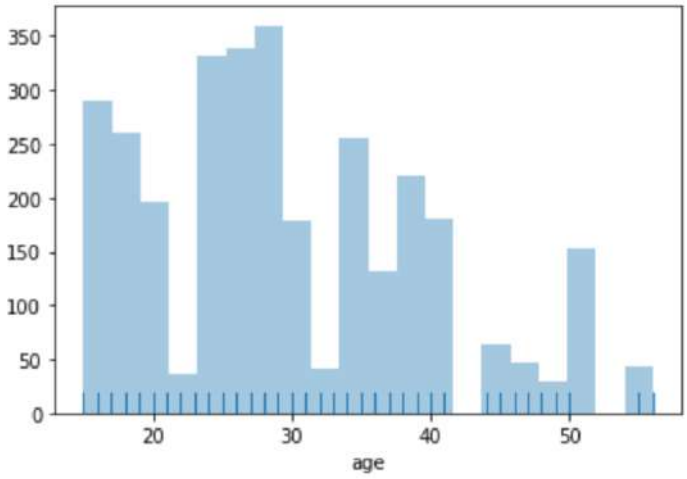

FIGURE 3.1: Histogram of the obtained user ages.

[1] https://api.twitter.com/1.1/search/tweets.json
[2] https://api.twitter.com/1.1/statuses/user_timeline.json

3.2 Yelp data collection

We get business data from Yelp dataset by Toronto city. After cleaning, preprocessing, and filtering dataset by City and business_type, equal to food, includes dataset had 4994 business entities (Table3.1), 179262 user's reviews (Table3.3), and 57243 unique users (Table3.2), that was writing tips and reviews. The user dataset had no user age and gender features. After obtaining bad metrics on Twitter dataset, which we will describe in chapter 6, was decided to get them by existed user profile internally, from Yelp. By having user_id, and using Yelp's GraphQL API [3] we can get user profile photos.

Using the pre-trained Deep EXpectation of apparent age from a single image model Rothe, Timofte, and Gool, 2016, we predicted user age and gender using a profile photo if there is face. As a result, we have got a user dataset, which has 38139 user reviews, and 33 feature for each user review, including Yelp-specific, like the text of the review, likes, Yelp friends, and predicted by model age and gender. Enriched user entity is shown in Table3.4.

address	attributes	business_id	categories	
865 York Mills..	{'RestaurantsDelivery': 'False', 'R...	C9oCPo..	Bakeries, Food	Toronto

TABLE 3.1: Food business entity example.

user_id	name	review_count	yelping_since	useful	funny	...
gvXtMj3XuPr0xHjgmlmtng	Peter	47	2014-01-05 20:45:54	57	26	...

TABLE 3.2: User entity example.

user_id	business_id	text	...
_N7Ndn29bpll_961oPeEfw	y-Iw6dZflNix4BdwIyTNGA	Good selection of classe...	...

TABLE 3.3: Review entity example.

user_id	user_name	user_gender_predicted	user_age_predicted	..
mZ1gXzL6Tn5Oky8_j0Kp7g	Barbara D.	F	32	...

TABLE 3.4: User entity enriched example.

[3] https://www.yelp.com/developers/graphql/guides/intro

Chapter 4

Overview of existed methods

In this chapter, we will cover the background information needed to reach the final goal - target audience construction. Methods that were used and theory. Usage of the methods and the solving pipeline would be covered in chapters 5, 6.

4.1 Clustering

4.1.1 DBScan

DBSCAN algorithm relies on a density-based notion of clusters which is designed to discover clusters of arbitrary shape. It clusters the data points to separate the areas of high density with the areas of low density (marking data points as outliers that are in the low-density regions) (Ester et al., 1996).

The detection procedure is based on the identifying of the dense regions that can be defined by the number of points close to some particular point. The algorithm requires two important hyperparameters: epsilon (the radius around the point) and minimum points (the minimum number of points in the given radius "epsilon"). If the point has more than or equal “minimum points” in its “epsilon” radius - it is marked as a core point. If the number of points in the radius epsilon is less than "minimum points" and it belongs to an "epsilon" radius of some core point - it is defined as a border point (Ester et al., 1996).

DBSCAN starts with an arbitrary point p and retrieves all points density-reachable from p. If p is a core point, this procedure yields a cluster with respect to parameters epsilon and min points. If p is a border point and no points are density-reachable from p and DBSCAN uses iterative "visiting" the next point of the database and recursive connecting of this points to the cluster of the core points. Algorithm creates new clusters for core points, that weren’t assigned to any clusters yet. Those points that weren’t assigned to any cluster are marked as noise or outlier points (Ester et al., 1996).

By the time, DBSCAN is more suitable to find arbitrary shaped clusters, it is often used as a spatial clustering algorithm.

4.1.2 K-Means

K-means is a centroid-based algorithm, or a distance-based algorithm, which calculates the distances to assign a point to a cluster.

The algorithm consists of the next steps: initialization, classification, centroid detection, and convergence. It is an iterative type of clustering that includes partitioning objects into k clusters, in such a way that the objects in one cluster are similar to each other and are different from those in another cluster (Pérez-Ortega, Almanza-Ortega, and Vega-Villalobos, 2019).

The main idea is to find centroids that are placed as much as possible far from each other. The first step is a random extraction of K sample points from the dataset as the center of the initial clusters. Then each point is assigned to the nearest cluster and the center point of all sample points in each cluster becomes the new center point of the cluster. The algorithm results vary with the choice of the center point, resulting in the most stable result after all iterations (Yuan and Yang, 2019 and Kodinariya and Makwana, 2013)

The K-means algorithm aims to choose centroids that minimise the inertia, or within-cluster sum-of-squares criterion (*Scikit-learn. Clustering* 2019):

$$\sum_{i=0}^{n} \min_{\mu_j \in C} (\|x_j - \mu_j\|)^2 \tag{4.1}$$

For a dataset that contains multidimensional points the method usually uses the Euclidean distance as a similarity index (Yuan and Yang, 2019 and Sharma, 2019).

Stopping criteria that can be adopted to stop the K-means algorithm:

- Centroids of newly formed clusters do not change;
- Points remain in the same cluster;
- Maximum number of iterations are reached.

The algorithm has an important hyperparameter - the k value, that can be found using such methods:

1. the rule of thumb;
2. elbow method;
3. information criterion approach;
4. information-theoretic approach;
5. silhouette method;
6. cross-validation.

The most common and used method is the Elbow method. The procedure includes iterative increasing the k value, starting from k=2, the following clustering and calculating the cost function. Until some k parameter cost function decreases dramatically and then starts reaching the plateau as it is shown on the Figure 4.1. Then cost function goes down very slowly. This significant change is defined as the best k parameter for the case (Kodinariya and Makwana, 2013).

Instead of cost function for Y-axis, also any other custom metrics can be used to find best k-value.

4.1.3 Hierarchical clustering

Hierarchical clustering algorithms group similar objects into groups called clusters. There are two types of hierarchical clustering:

- Aglomerative is a bottom-up approach. It starts with many small clusters and merges them together to create bigger ones;
- Divisive is a top-down approach. It starts with a single cluster than break it up into smaller ones.

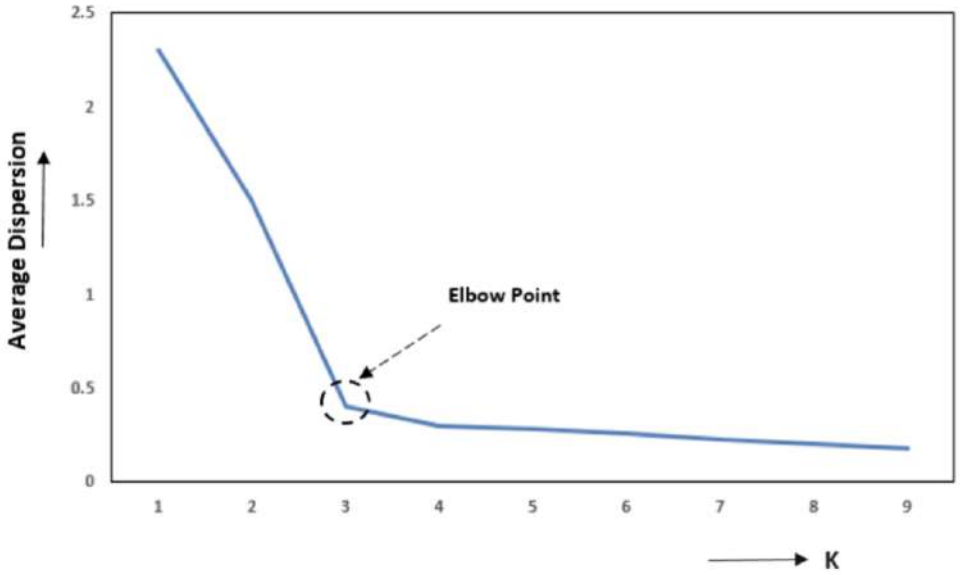

FIGURE 4.1: The Elbow method for detecting best k-value.
Dangeti, 2017

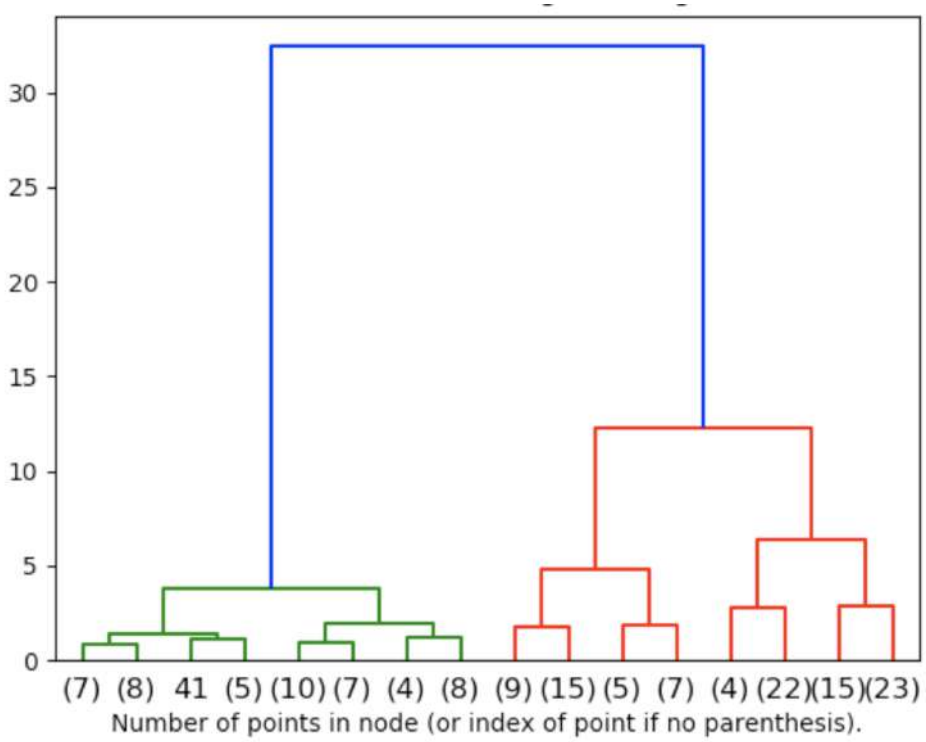

FIGURE 4.2: Hierarchical clustering dendrogram.
Plot Hierarchical Clustering Dendrogram 2019

We can use a dendrogram to visualize the history of groupings and figure out the optimal number of clusters as it is shown on Figure 4.2.

The linkage criteria determines the metric used for the merge strategy.

- Ward minimizes the sum of squared differences within all clusters. It is a variance-minimizing approach and in this sense is similar to the k-means objective function but tackled with an aglomerative hierarchical approach.;

- Maximum or complete linkage minimizes the maximum distance between observations of pairs of clusters.;
- Average linkage minimizes the average of the distances between all observations of pairs of clusters;
- Single linkage minimizes the distance between the closest observations of pairs of clusters (Sarkar, 2019).

Various linkage types are depicted on the Figure 4.3.

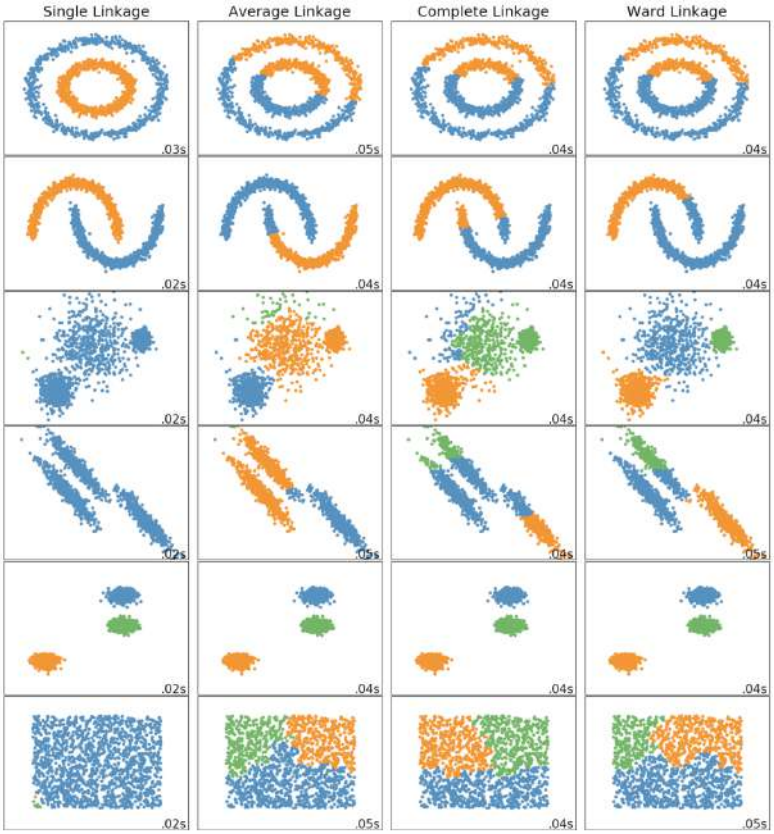

FIGURE 4.3: The comparison of clustering using different linkage types
Comparing different hierarchical linkage methods on toy datasets 2019

4.1.4 Evaluation metrics for clustering

The most popular metrics to evaluate the quality of clusters are:

1. Davies-Bouldin score:

   The metric is calculated as the average similarity measure of each cluster with its most similar cluster, where similarity is the ratio of within-cluster distances to between-cluster distances. Thus, clusters that are situated far away from each other and are less dispersed will result in a better score (Davies, 1979).

   The score is calculated using the following formula:

$$DB = \frac{1}{n}\sum_{i=0}^{n}\max_{j\neq i}\left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\right) \tag{4.2}$$

where n is the number of clusters and i is the average distance of all points in cluster i from the cluster centroid c_i.

The index shows the insights that "good" clusters should be far away from each other and very dense. The 'max' statement in the formula repeatedly selects the values where the average point is farthest away from its centroid, and where the centroids are closer to each other. The minimum value of the score is zero, lower Davies-Bouldin index indicates better separation between the clusters (*Assessment Metrics for Clustering Algorithms* 2018).

2. Silhouette Coefficient:

   Silhouette indicates the separation distance between the clusters obtained after the research. The silhouette plot displays how close each point p in one cluster is to points in the other nearby clusters and therefore gives an ability to assess such parameters like a number of clusters visually.

   This index has a range of [-1, 1]. Silhouette scores near +1 mean that the sample is distant from the bordering clusters, 0 indicates that the sample is approaching the decision boundary between two neighboring clusters and negative values around -1 show that samples are likely assigned to the wrong cluster.

   Bad selection of the clusters using the visual approach can be presented by:

   (a) below average silhouette scores;

   (b) wide fluctuations in the size of the silhouette plots.

   The minimum value of the score is zero, lower Davies-Bouldin index indicates better separation between the clusters (*Selecting the number of clusters with silhouette analysis on KMeans clustering* 2019). The Silhouette Coefficient is calculated using two important values: 1) the mean intra-cluster distance (a); 2) the mean distance between closest neighbouring clusters (b), that are defined for each point. Smaller the value (a) and larger the value of b(i) - better the assignment to the cluster.

   The Silhouette Coefficient is calculated using the formula:

$$S_i = \frac{b_i - a_i}{\max(a_i, b_i)} \tag{4.3}$$

   where, a_i is the average dissimilarity of the point p to all points in the same cluster and b_i is the average dissimilarity of the point with all objects in the closest cluster (J.Rousseeuw, 2016). The analysis is presented on the Figure 4.4. The red line is the average silhouette score for 2 clusters.

   The best clustering results should match such conditions:

   (a) the mean score should be around 1;

   (b) the clusters plot below mean score is not desirable;

   (c) the width of all clusters should be uniform.

   (*Selecting the number of clusters with silhouette analysis on KMeans clustering* 2019)

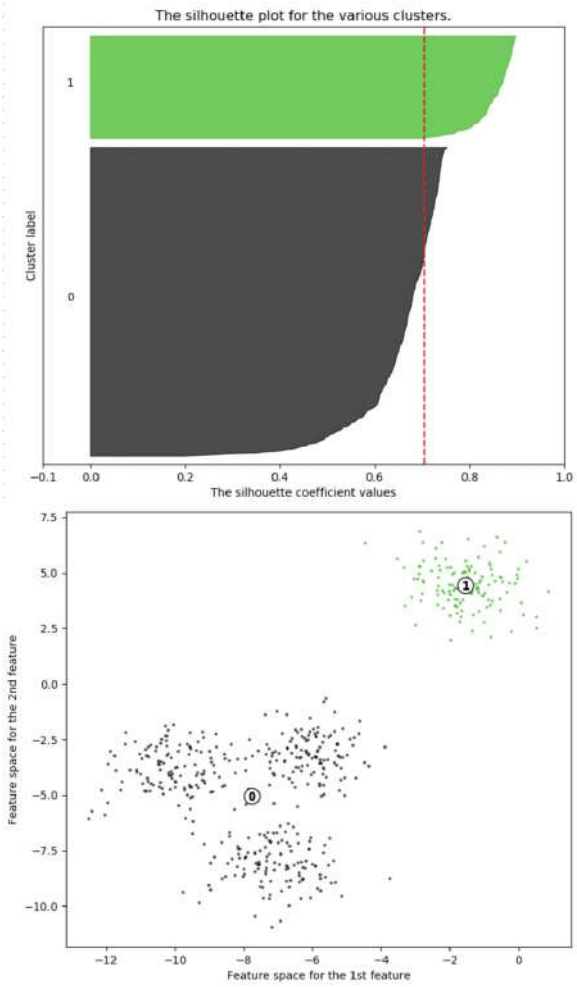

FIGURE 4.4: Silhouette analysis for KMeans clustering on sample data with n=2.
Selecting the number of clusters with silhouette analysis on KMeans clustering 2019

4.2 Text pre-processing

4.2.1 TF-IDF

TF-IDF is a combination of Term Frequency and Inverse Document Frequency. TF-IDF works by determining the relative frequency of words in a specific document compared to the inverse proportion of that word over the entire document corpus (P.A. Pérez-Toro, 2019).

TF identifies how many times the keyword is used in the document. The importance of the keyword ti in the document can be calculated as:

$$TF(word) = \frac{Count(word)}{\sum_{i=0}^{n} Count(word)} \tag{4.4}$$

where $Count(word)$ represent the number of occurrences of word t_i in the document (Jie Chen and Liang, 2016). IDF means the importance of the keyword and it is used to assign a lower weight to frequent words (for example, 'and', 'of', 'that') and higher - for infrequent words. It means that less occurrence of words in some document identifies its higher importance (Shahzad Qaiser, 2018).

Let's say we have N documents in the collection, and that term t_i happens in n_i of them. The IDF can be calculated as:

$$idf(t_i) = log\frac{N}{n_i} \tag{4.5}$$

(Robertson, 2004) Intuitively, the TF-IDF calculation determines how relevant a given word is in a particular document. The formal procedure for implementing TF-IDF overall approach works as follows. Given:

1. document collection D;
2. individual document d D;
3. word w.

$$w_d = f_w, d * log(\frac{|D|}{f_w}, D) \tag{4.6}$$

where f_w, d equals the number of times w appears in d, $|D|$ is the size of the corpus, and f_w, D equals the number of documents in which w appears in D (Salton, 1988).

4.3 Demographics prediction

4.3.1 Random Forest

The random forest approach seems to be notably successful as a general method for classification and regression. Random forests are created by connecting the predictions of several basic classifiers (trees), each of which is trained independently. The predictions of trees in Random Forest are combined through the averaging in comparing to complex weighting approach as basic classifiers are combined in Boosting. The Random Forest can use different schemas during tree constructing:

1. method for splitting the leafs;
2. criterion to choose between different splits;
3. way of introducing randomness to the three.

The trained Random Forest model is used to make predictions for a query point x, each tree independently makes a prediction using the formula:

$$f_i^n(x) = \frac{1}{N^e(A_n(x))} \sum_{Y_i \in A_n(x), I_i = e} Y_i \tag{4.7}$$

and then answers are averaging to use randomness in trees prediction:

$$f_n^M(x) = \frac{1}{M} \sum_{j=1}^{M} f_n^j(x) \tag{4.8}$$

(Misha Denil, 2014)

4.3.2 Boosting algorithms

The traditional ensemble methods like Random forests are based on simple averaging of models in the ensemble. The family of boosting methods relies on a different, constructive approach of ensemble composition. The main idea of boosting is to add new learners to the ensemble sequentially (Natekin and Knoll, 2013). The concept Boosting method is that any weak base-learner can potentially become a strong learner by iterative combining the solutions of weak learners to get the best prediction result. Thus, weak learners are iteratively boosted (improved) to become a strong learner to make an accurate classification.

However, only using the calling of the weak learner multiple times on the training set is not enough to get the best performance. The basic idea of boosting is to manipulate the training data by iteratively re-weighting these objects. The weighting of the objects is based on the learner's performance on the previous iterations. Suchwise, the algorithm is forced to concentrate its output basing on the objects that are hard to classify. The observations that were given the wrong prediction are getting the higher weights up to the current iteration. In the end, the resulting majority vote chooses the class most often selected by base-learners using error on each iteration into the account (Andreas Mayr, 2014).

Gradient boosting methods (simply GBMs) use sequential learning to fit new models. The concept of this approach is to build new base-learners to maximally correlate with the negative gradient of the loss function, that associates with the whole ensemble. If the loss function is a classic squared-error loss, then the learning will result in continuous error-fitting (Natekin and Knoll, 2013).

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function (*Gradient boosting* 2019).

To design a GBM for a given problem, one has to specify the choices of functional parameters $\psi(y, f)$ - loss function and $h(x, \theta)$ - a base-learner. Thus, one has to provide what exactly is going to optimized, and then, choose the form of the function, which will be used in solution construction.

The choice of a loss function is often affected by the demand for specific features of the distribution of the observations. The most frequent examples of such property are the robustness to outliers, but other characteristics can also be considered. Loss-functions can be divided according to the type of response variable y. The most frequently used loss-functions according to the family of the response are:

1. Continuous response (y R): Gaussian L2 loss function, Laplace L1 loss function, Huber loss function, specified, Quantile loss function, specified;

2. Categorical response (y 0, 1): Binomial and Adaboost loss functions;

3. Other families of response variable: Loss functions for survival models and custom loss functions..

The regularly used base-learners can be classified into three distinct categories: linear models, smooth models, and decision trees. There are also other models, that can be used, such as Markov random fields or wavelets, but their application arises for very specific practical tasks (Natekin and Knoll, 2013). One of the boosting algorithms, under the Gradient Boosting framework, that was used in this work is XGboost. XGBoost provides a parallel tree boosting (also known as GBDT, GBM)

that solve many data science problems in a fast and accurate way (*Xgboost Documentation* 2019).

XGBoost is a boosted decision tree algorithm. It is an extension of an approach called Gradient Boosting, which itself is an extension of the AdaBoost algorithm. Rather than trying to parallelize the training of trees, it parallelizes the training of nodes within each tree.

The scalability of XGBoost is due to several important and algorithmic optimizations. These changes include:

- a novel tree learning algorithm for handling sparse data;
- a justified weighted quantile sketch procedure, that enables handling instance weights in approximate tree learning;
- parallel and distributed computing, that makes learning faster and enables quicker model exploration (Chen Tianqi, 2016).

Chapter 5

Proposed method

This section presents the proposed model for finding competitors and their audience analysis, to construct a target audience profile for a given restaurant. That considers two phases, the competitor's forecasting, and user classification phase.

The proposed method consists of two main parts:

- Competitors analysis.
- User demographics prediction phase.

Figure 5.1 shows pipeline of solving task of target audience profile construction through competitors analysis.

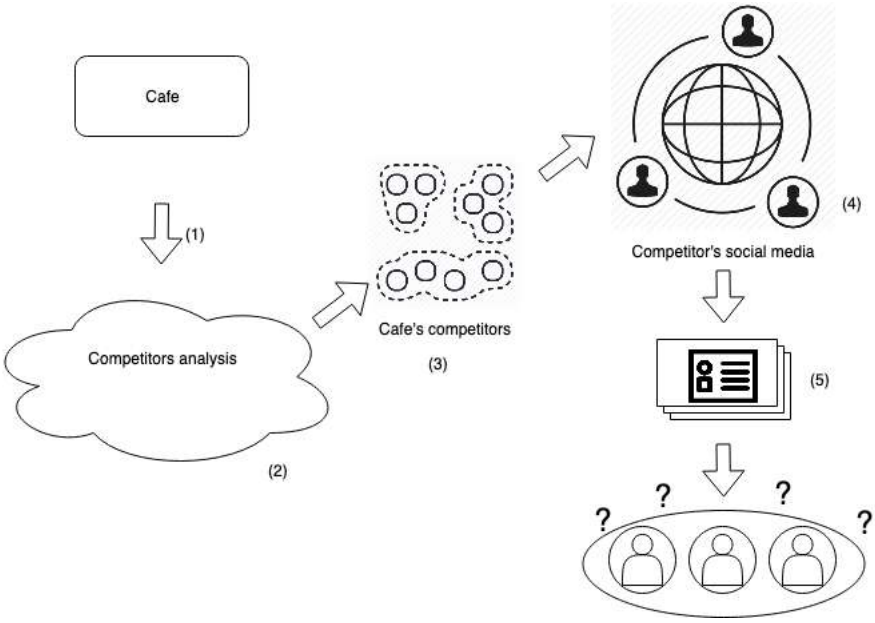

FIGURE 5.1: Proposed model scheme.

5.1 Competitors analysis

The goal of the competitor's analysis is to find competitors by category, location, and attributes, which every food place has.

Examples of the features can be:

- business parking = True/False
- price range = 1/2/3/4/5

- take out = True/False
- wifi = True/False

Food places dataset has 4994 rows, each row relates to food place. Each food place has a category column, which has categories for a food place. Example: Sandwiches, Cafes, Coffee. Main categories for the dataset, which was defined, as those, who had more than 200 samples in the dataset, and numbers of food places in each of them shown in 5.1.

category name	food places number
Bakeries	587
Restaurants	2023
Bars	346
Breakfast & brunch	342
Cafes	481
Coffee & tea	1368
Desserts	545
Fast food	539
Grocery	359
Ice cream & frozen yogurt	295
Juice bars & smoothies	260
Nightlife	363
Sandwiches	376
Seafood	257
Specialty food	801

TABLE 5.1: Food business categories dataset.

For clustering, the following strategy was defined:

1. Make clusters for main categories.
2. Spatial clustering for food places in each category.
3. Cluster by other attributes in each spatial cluster.

As a result, each cluster will have competitors, and they will be used to understand their user demographics.

5.2 Age group prediction

The goal of user demographics forecasting is to get data (reviews and user features) from competitor's business profiles and predict their age group and gender. User demographics were performed on 38139 user data and review samples, and the idea is to determine, which age group does current user belongs to. There was decided to divide people into three groups, distribution by groups looks as follows 5.2:

The classification task was performed after tf-idf preprocessing of the user texts (reviews).

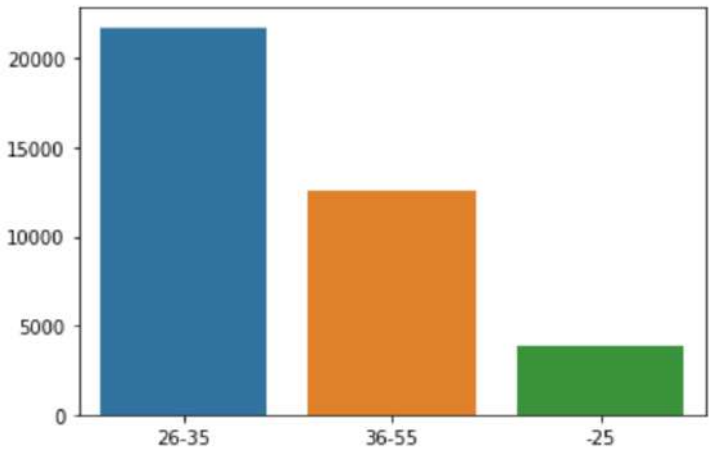

FIGURE 5.2: Distribution of age group labels.

5.3 Merging results

After we have classifier for user age, and clusters with competitors, we will be able to use age classification from competitor's users reviews, and forecast their gender by name. Having data by each competitor, we will be able to combine them, and provide business owner with audience portrait result.

Chapter 6

Experiments

6.1 Competitors analysis

6.1.1 Spatial clustering

Initially, it was decided to split each category spatially, because of the assumption, that competitors should be close-enough to compete on the Toronto city market.

We compared two clustering algorithms, KMeans and DBSCAN, using only latitude and longitude features. There is no general solution to find the optimal number of clusters, especially spatially, so we compared the results of multiple runs with different number of classes/different epsilon in kilometers in terms of DBSCAN and choose the best one according to a given metric, but we decided to split each category spatially into $k = 2...15$ not to overfit. By the time, data is 2-dimensional, we can render it on the map. For each category situation enough alike, so we will describe for "coffee tea" category. Visual representation of clusters on the map can be seen in Figure 6.1. With DBSCAN, and it's different epsilon parameter, some clusters' sizes are too small, and because of food business density in the city center, we are forming one big cluster. With KMeans, all stations would be clustered and their sizes are similar, and we will not take into account businesses in the borders for now.

Later, when we decided to use KMeans for spatial clustering, we need to decide on k parameter. For doing that, we used Elbow method with silhouette metric (J.Rousseeuw, 2016). Elbow plot for category "coffee tea" we can see in Figure 6.2. For all categories, choosed number of spatial clusters shown in Figure 6.3.

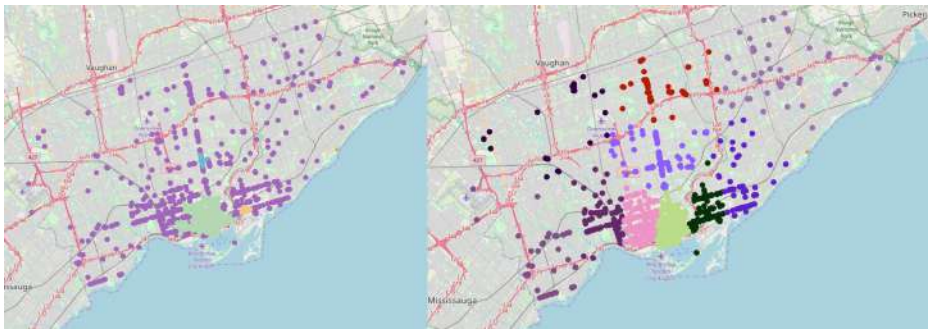

FIGURE 6.1: DBSCAN clustering with eps=0.5km and min_samples=5 and KMeans with k=10.

6.1.2 Inner category clustering

After we finished with spatial clustering of the food businesses, we split food businesses by business attributes, such as wifi availability, vegan options included, price

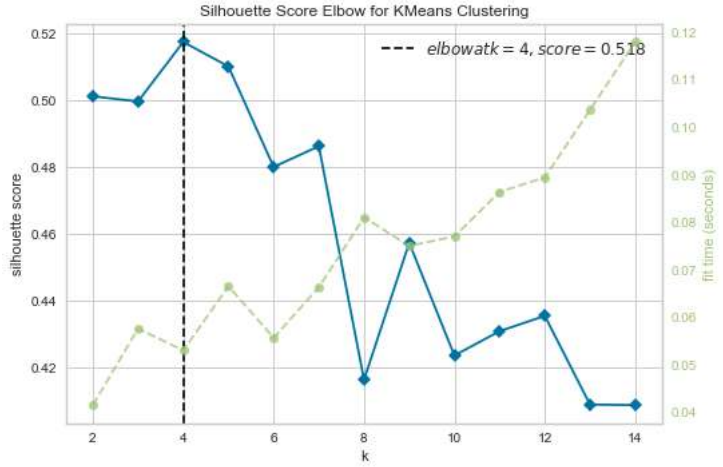

FIGURE 6.2: Visual elbow method plot for selecting k in K-means.

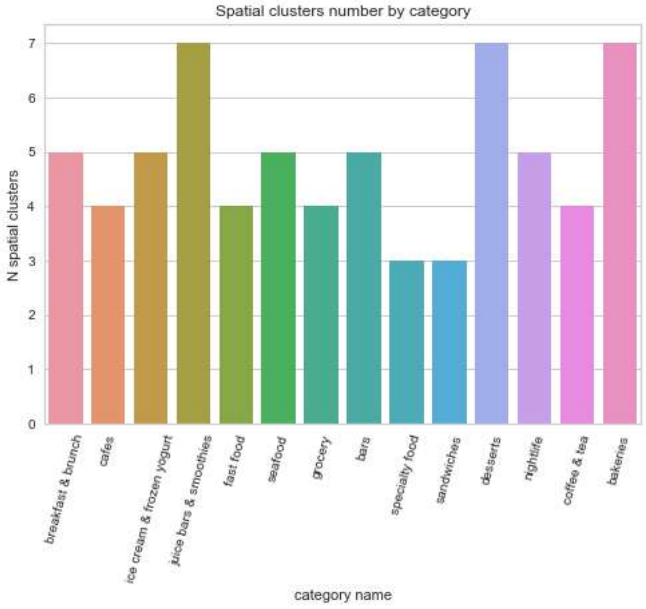

FIGURE 6.3: Selected number of spatial clusters by category.

range, and others. That can be done with clustering algorithms too. For each unique key (category, spatial cluster) it was decided to compare different clustering algorithms with selected parameter. For a comparison, we choosed silhouette metric. We compared Kmeans and Agglomerative clustering algorithms on a different number of k's. Figure 6.4 shows an example for selected "coffee & tea" category. We can see, that in most cases, for different numbers of clusters, that silhouette metrics are close to each other.

In case of agglomerative clustering, we got a hierarchy of objects in clusters that can we can further split, for deeper understanding of competitors inside smaller clusters. Also, trade-off of quality vs quantity should be considered, when we are discussing number of clusters. We can see that for each cluster, average silhouette metric starts with value 0.7 - 0.8 (for number of clusters equals 2), and then when increasing number of clusters, is going down. We should take into account, that interia (inner cluster distances metric) decreases, which means, that closeness of object inside competitors cluster growth, which is relevant for business problem's point of

view. Also, we can look at silhouette plots to compare methods. Figure 6.5 shows the thickness of the silhouette plot, and cluster size can be visualized. For n=11 absolute value is not big (0.5), but we can see that all the plots are more or less of similar thickness.

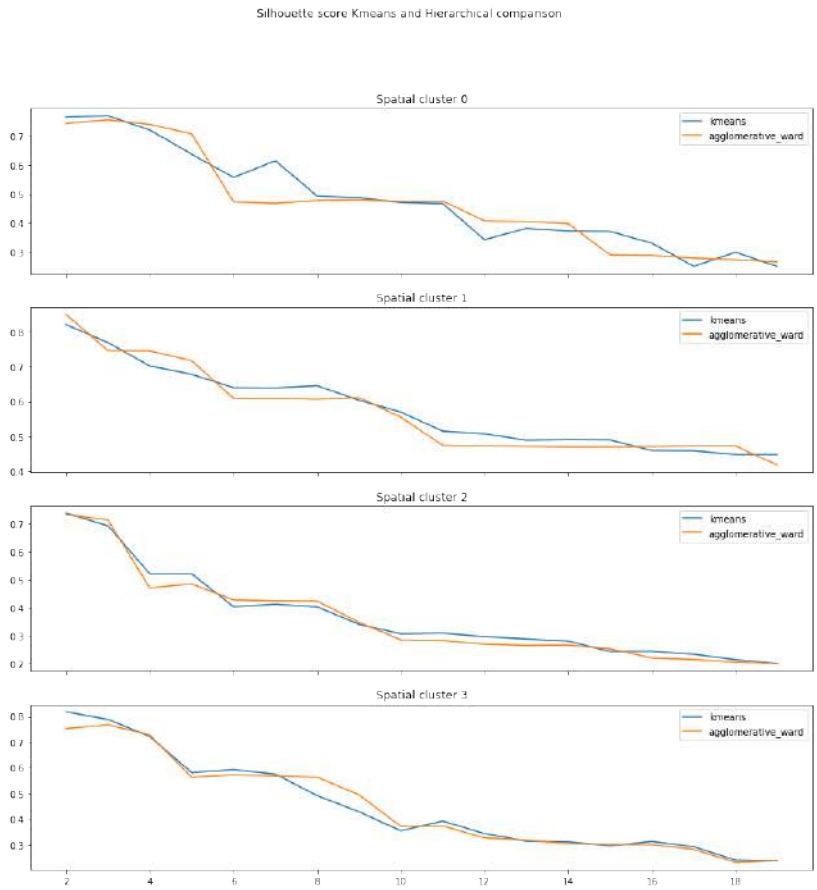

FIGURE 6.4: KMeans and Hierarchical clustering comparison in silhouette metric.

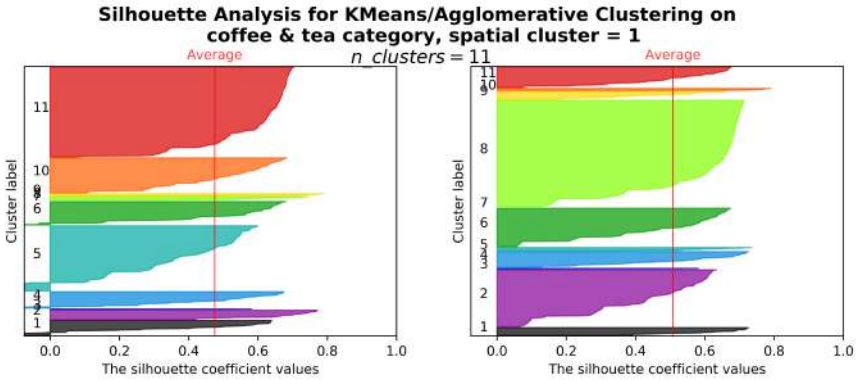

FIGURE 6.5: Silhouette plot for selected category inner clustering.

6.1.3 Human metrics

Human metrics are very important in every research, especially when it comes to comparing metrics not to baseline, but make analysis, which consists of many parts,

and have no baseline, which is tuned. Here, while making clustering, our main goal was to find competitors. By using user reviews, we can check, whether people like to go to the food places at the same cluster, which in some sense may be a good thing to measure. After spatial and internal clustering is done, our unique key for each cluster in category is a compound - it consists of (spatial_cluster, cluster). For each cluster "user return rate" can be calculated. We define *ReturnRate* metric, which can describe, how likely customers, that have visited certain food places in one cluster, will go to other clusters' places.

$$ReturnRate_j = \sum_{i=0}^{n} \left(\frac{k_{i,j}}{\sum_{m=0,m!=j}^{D} (k_{i,m}) / (D-1)} \right) / n \tag{6.1}$$

, where j- cluster's index, D - number of clusters, $k_{i,j}$ - number of ith user visits inside jth cluster.

If Return rate for a given cluster more than 1, it means, that people tend to choose between certain cluster's food places more likely, than places in other clusters. If it is less than 1, than they would prefer another cluster's restaurant. Example of such rate for category "coffee & tea" and first spatial cluster is presented in the Table 6.1

Cluster_0	_1	_2	_3	_4	_5	_6	_7	_8
3.97	4.17	4.9	3.72	3.47	2.93	2.33	4.31	3.38

TABLE 6.1: Return rate for "coffe & tea" category's internal clusters.

6.2 Age prediction

One of the first ideas to predict user age in Yelp, by the time there were no labels in Yelp dataset, was to get data from Twitter, train classifier on text features, using TF-IDF, and then try to test it on Yelp's data. The assumptions was:

1. Different ages people's writing patterns are different.
2. We would be able to predict age based on text features only n_grams with (1, 1) or (1, 2) dimensionality.
3. Trained classifier would be able to predict on other social media's texts with f1 score > 0.65 for each user age range class.

We trained Random Forest classifier on Twitter data, enriched text features with punctuation features, and emoji's, like (!_exists, !_number,)_exists, ...,). After tuning with different features, classifier parameters, we got the result, which can be seen in Table 6.2.

user_age_range	precision	recall	f1_score
-25	0.51	0.38	0.44
26-35	0.47	0.59	0.52
36-55	0.47	0.47	0.47

TABLE 6.2: Twitter text-only data metrics.

Metrics are very bad, which means one of the following:

1. There was not enough data (3k users tweets in total)
2. Classifier pipeline that we choosed was wrong.
3. We cannot predict age only on text patterns.

We rejected strategy of training on Twitter's data, because it would not fulfill our requirements for the model.

As was described in Chapter 3, we found a solution to label data internally, inside Yelp. We tried to train on different datasets, and features, using n_grams for TF-IDF with dimensionality (1, 1) and (1, 2). We used Grid Search to choose optimal parameters, using cross-validation. Also, we tested bagging algorithms (Random Forest) and boosting (XGboost).

A very important part of experiments was dataset pre-processing, we have used two variants for the same user_with_review dataset. The first, treated every row as separate user review, and cannot be used with meta-features, because of each user, having the same meta-features for different dataset rows. The second one has 'text' column, and treats all user reviews as one text, generated by the user, that way, the classifier was able to use other features, related to the user.

On Table 6.3 we can see metrics on test dataset, like avg_precision, avg_recall, and avg_f1_score for different types of model. For testing, 20% of each dataset was used, with stratification by label(user age group).

Model	Dataset	Avg precision	Avg recall	Avg f1-score
Random Forest + (1, 1) gram	grouped	0.58	0.63	0.6
Random Forest + (1, 2) gram	grouped	0.67	0.71	0.69
XGBoost + (1, 1) gram	grouped	0.63	0.63	0.63
XGBoost + (1, 2) gram	grouped	0.68	0.73	0.71
Random Forest + (1, 1) gram	non-grouped	0.71	0.46	0.55
Random Forest + (1, 2) gram	non-grouped	0.58	0.5	0.53
XGBoost + (1, 1) gram	non-grouped	0.65	0.48	0.55
XGBoost + (1, 2) gram	non-grouped	0.6	0.61	0.6

TABLE 6.3: Macro average metrics for age prediction.

Chapter 7

Conclusion

7.1 Results

The goal of this work was to create a framework for business owners to construct their audience profile using competitors' data. We tried different methods for people's age forecasting and competitors detection. By combining these methods, we have built a baseline, which can be further tuned and made more flexible for business needs. As a result of this research, the obtained model already can be used by business owners.

The main difference of this work is a fully-automated approach to business marketing needs. We chose the hierarchy of clustering algorithms to solve competitors' detection problem (K-means + Hierarchical). And Boosting algorithm (XGBoost) to solve age group prediction problem.

Also, by the time this research was done as a personal initiative, we had no marketing data, and we cannot A/B test, or do any user-related testing. Also, we faced a lack of data problems. We discussed from the engineering side the process of obtaining needed for research data in Dataset chapter, but the process itself was slow and does not create any guarantee for future work.

For now, using manually-crafted dataset, we have built a core for future research and proved the possibility of using this concept.

7.2 Further work

Luckily, this result can be improved, using more training data, and more user-specific features. For the first time, we thought about this research as platform-agnostic, but we faced a lot of problems, which cannot be solved anyhow but using more data inside some company's infrastructure.

It is crucial to work on age forecasting problems using text and social network meta-features, because this problem is rarely touched is scientific papers. For now, we predict only for 3 ranges of ages, the number of classes can be higher.

Bibliography

Andreas Mayr Harald Binder, Olaf Gefeller Matthias Schmid (2014). "The Evolution of Boosting Algorithms From Machine Learning to Statistical Modelling". In: *Methods of Information in Medicine* 53.5.

Assessment Metrics for Clustering Algorithms (2018). URL: https://medium.com/@ODSC/assessment-metrics-for-clustering-algorithms-4a902e00d92d.

Chen Tianqi, Guestrin Carlos (2016). "XGBoost: A Scalable Tree Boosting System". In: pp. 784–794.

Comparing different hierarchical linkage methods on toy datasets (2019). URL: https://scikit-learn.org/stable/auto_examples/cluster/plot_linkage_comparison.html.

Dangeti, Pratap (2017). *The elbow method. Statistics for Machine Learning*. URL: https://www.oreilly.com/library/view/statistics-for-machine/9781788295758/c71ea970-0f3c-4973-8d3a-b09a7a6553c1.xhtml.

Davies David L. Bouldin, Donald W. (1979). "A Cluster Separation Measure". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence. PAMI* 1.2, pp. 224–227.

Ester, Martin et al. (1996). "A Density-Based Algorithm for Discovering Clustersin Large Spatial Databases with Noise". In:

Gradient boosting (2019). URL: https://en.wikipedia.org/wiki/Gradient_boosting.

Guimarães, Rita Georgina (2017). *Age Groups Classification in Social Network Using Deep Learning*. URL: https://ieeexplore.ieee.org/document/7932459.

Gur, Furkan Amil (2018). *Know Thy Enemy: A Review and Agenda for Research on Competitor Identification*. URL: https://journals.sagepub.com/doi/10.1177/0149206317744250.

Jie Chen, Cai Chen and Yi Liang (2016). "Optimized TF-IDF Algorithm with the Adaptive Weight of Position of Word". In: *Advances in Intelligent Systems Research* 133, pp. 114–117.

J.Rousseeuw, Peter (2016). "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis". In: *Computational and Applied Mathematics* 20, pp. 53–65.

Kodinariya, Trupti m and P.R. Dan Makwana (2013). "Review on Determining of Cluster in K-means Clustering". In: *International Journal of Advanced Research in Computer Science and Managment Studies* 1.6, pp. 90–95.

Liu-Thompkins, Yuping. *A Decade of Online Advertising Research: What We Learned and What We Need to Know*. URL: https://www.tandfonline.com/doi/abs/10.1080/00913367.2018.1556138?journalCode=ujoa20.

Ltd., Mintel Group. *Foodservice trends 2019*. URL: https://media1-production.mightynetworks.com/asset/4406749/Mintel-Foodservice-Trends-2019.pdf.

McAuley, Julian. *Recommender Systems Datasets*. URL: http://cseweb.ucsd.edu/~jmcauley/datasets.html.

Misha Denil David Matheson, Nando de Freitas (2014). "Narrowing the Gap: Random Forests In Theory and In Practice". In: *Proceedings of the 31st International Conference on Machine Learning* 32.1, pp. 665–673.

Morgan-Lopez, Antonio A. (2017). *Predicting age groups of Twitter users based on language and metadata features*. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0183537.

Natekin, Alexey and Alois Knoll (2013). "Gradient boosting machines, a tutorial". In: *Front Neurorobot* 7.21.

Oplatkova, Zuzana Kominkova (2014). *Enterprise Competitive Analysis and Consumer Sentiments on Social Media Insights from Telecommunication Companies*. URL: https://www.researchgate.net/publication/268669101_Enterprise_Competitive_Analysis_and_Consumer_Sentiments_on_Social_Media_Insights_from_Telecommunication_Companies.

P.A. Pérez-Toro J.C. Vásquez-Correa, M. Strauss J.R. Orozco-Arroyave E. Nöth (2019). "Natural Language Analysis to Detect Parkinson's Disease". In: pp. 82–90.

Plot Hierarchical Clustering Dendrogram (2019). URL: https://scikit-learn.org/stable/auto_examples/cluster/plot_agglomerative_dendrogram.html.

Pérez-Ortega, Joaquín, Nelva Nely Almanza-Ortega, and Andrea Vega-Villalobos (2019). *The K-Means Algorithm Evolution*. URL: https://www.intechopen.com/online-first/the-k-means-algorithm-evolution.

Robertson, Stephen (2004). "Understanding Inverse Document Frequency: On theoretical arguments for IDF". In: *Journal of Documentation* 60.5, pp. 503–520.

Rothe, Rasmus, Radu Timofte, and Luc Van Gool (2016). "Deep expectation of real and apparent age from a single image without facial landmarks". In: *International Journal of Computer Vision (IJCV)*.

Salton G. Buckley, C. (1988). " Term-weighing approache sin automatic text retrieval. " In: *Information Processing Management*, 24.5, pp. 513–523.

Sarkar, Dipanjan (2019). *Text Analytics with Python: A Practitioner's Guide to Natural Language Processing*. Apress.

Scikit-learn. Clustering (2019). URL: https://scikit-learn.org/stable/modules/clustering.html.

Selecting the number of clusters with silhouette analysis on KMeans clustering (2019). URL: https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html.

Shahzad Qaiser, Ramsha Ali (2018). "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents". In: *International Journal of Computer Applications* 181.1.

Sharma, Pulkit (2019). *The Most Comprehensive Guide to K-Means Clustering You'll Ever Need*. URL: https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/.

Teza. *Factors affecting closure of the business*. URL: https://www.kaggle.com/tejasrinivas/factors-affecting-closure-of-a-business-on-yelp.

Xgboost Documentation (2019). URL: https://xgboost.readthedocs.io/en/latest/.

Yelp. *Yelp dataset*. URL: https://www.yelp.com/dataset.

Yuan, Chunhui and Haitao Yang (2019). "Research on K-Value Selection Method of K-Means Clustering Algorithm". In: *Multidisciplinary Scientific Journal* 2.2, pp. 226–235. DOI: https://doi.org/10.3390/j2020016.