Project

Data Processing 2: Scalable Data Processing, Legal & Ethical Foundations of Data Science

Predicting the popularity of Reddit posts

Martin Scholz

Mark Tallai

Felix Gstettner

Stepan Malysh

Dr. Sabine Kirrane, Astrid Krickl, MSc.

Vienna, February 7, 2022

Welthandelsplatz 1, A-1020, Wien
Telefon: +43/1/313 36/4622 ● Telefax: +43/1/313 36/717
Web: www.wu.ac.at/retail

Table of contents

1. Introduction

Reddit is one of the most popular social news websites. Every registered user can upload posts and share his/her thoughts and ideas with the community. Such social websites are becoming increasingly interesting for many companies. Firms might use posts to promote their products or use discussions between Reddit users to boost the firm's image. Some companies even use social websites to search for new employees. To realize such strategies the company needs to decide how to write a strong post that will get the required attention. Furthermore, we are all passionate redditors. For these reasons we have undertaken a project that aims to predict the popularity of reddit posts based on metadata and the written content of posts. We are going to restrict our analysis to two subreddits: /r/Atlanta and /r/elderscrollsonline - a mix of our interests. These subreddits should demonstrate what kind of posts these different communities value. We will compare the results obtained by different machine learning techniques. A subreddit can be generally seen as an online group that focuses on specific topics. Members can up- and downvote posts. As the concrete number of up- and downvotes that constitutes a popular or unpopular post depends on the specific subreddit, the algorithm will only analyse collected data from one specific subreddit at the time. The algorithms itself, however, could be applied to any subreddit. What constitutes a popular post within a subreddit will be automatically detected by clustering. Furthermore, note that our analysis is restricted to posts written in English.

This project qualifies as big data project because there are millions of posts and comments which lead to a huge amount of data that could be relevant for predicting the popularity of Reddit posts. This implies that the volume criterion is fulfilled. Furthermore, new posts and comments are uploaded every second. The velocity criterion is thus fulfilled too. Finally, the data shows variety. On the one hand, the collected data contains metadata (such as the score (difference between upvotes and downvotes, the author (i.e., the username) or the date and time when the post was published). On the other hand, there is the post itself, i.e., a string that contains some written information or a picture. No post is perfectly identical to another and thus the data about the posts themselves is varied. The project will be scalable by using the PySpark-library and the Apache Hadoop Ecosystem.

This report starts by discussing how the data is collected and discusses the subreddits that are analysed. Secondly, algorithms and models are discussed. Results obtained in each step are presented. Then, predictions obtained by logistic regression and decision trees are compared.

Next, our experiences are presented. Finally, we summarize challenges encountered and give recommendations for future work.

2. Data Sources

This project is based on data collected on Reddit. In fact, posts (i.e., the string that constitutes a written post) and corresponding metadata will be accessed through the Reddit API (by using the Python praw-library) and stored by using a pyspark-dataframe. As the algorithm does only analyse a post based on its written text and metadata (such as the score, upvote ratio and number of comments) we did not combine any datasets because the required data is provided by accessing the Reddit API directly.

2.1. Subreddits

2.1.1. Atlanta

/r/Atlanta is the official subreddit of the city Atlanta. It has a population nearly 500 thousand and is the capital and biggest city of the state Georgia. The conversations in r\Atlanta are therefore mainly focused on local issues. The subreddit about Atlanta has approximately 405 thousand members.

2.1.2. r/elderscrollsonline

Elder Scrolls Online is a popular massive multiplayer online roleplaying game (MMORPG) that is developed by ZeniMax Online Studios and published by Bethesda Softworks. The subreddit /r/elderscrollsonline is a forum to talk about the game for the players. This subreddit has around 370 thousand members.

3. Preprocessing

The algorithm starts by calling the pre-processing function. This function uses the praw-library next to some pyspark-modules. First, a connection to the Reddit API is established. Next, the function requests the posts of a given subreddit. Different variables could be retrieved in this step. The preprocessing function collects the post itself, the date and time, the score, the number of comments and the upvote ratio. This data is stored into a pyspark-dataframe. In the next step, the algorithm filters the posts based on the date published. As discussed below, this filtering is needed to get rid of recent posts that had not enough time to gather attention. When accessing the submission of r/elderscrollsonline and r/Atlanta we filtered out every post that was younger than one day. This should ensure that posts had some time to gather comments and votes and returns enough observations. Note that Reddit by default limits the requests to the first 1000 post of a category. A higher time threshold might have improved predictions because there might be more posts with a meaningful number of comments and a relevant score. However, our algorithm requests the newest posts, which means that the collected data is restricted to a maximum of 1000 posts. Therefore, to get enough remaining observations, posts older than 24 hours were kept. Note that there would have existed the option to access the hot posts (i.e., the posts with the highest score) instead. These posts were deliberately not collected by the algorithm as all hot posts constitute popular ones. This will lead to biased estimates as the collected data does not represent the entire population.

Furthermore, the algorithm also cleans the post by removing several strings. These expressions are mainly based on suggestions made in the lecture. Two important strings that were removed are commas and paragraphs. While these symbols did not lead to any problems while working directly with spark dataframes, they lead to problems with the hard save of our used data. Please note that we decided to do a local save of the requested data. This is done to make results reproducible. Calling the accessing function returns different posts depending on the date and time. First, as the requested documents are limited to 1000 newer post replace some of the older posts. Secondly, comments that were filtered out when we called the function will now be older than 24 hours and thus included in the resulting dataframe. For reproducibility, the data that was used can be found in the accompanying csv-files. Because of this local save, the strings had to be cleaned before saving. If commas and paragraphs are kept, the locally saved csv cannot be loaded into spark correctly because each commas indicates a new variable and each paragraph a new observation. Therefore, commas and paragraphs were removed from the posts before the data was saved locally.

Additionally, linked usernames and zero-width spaces were removed too. We decided to drop any observation with missing values. Note that based on our experience the API seems to work reliably and missing values in the metadata occurs rarely. However, if a post consists only of an image the post variable is missing (as there is no text available that could be accessed). As our project analyses reddit posts based on the written text this is not a big problem because our analyses are restricted to subreddits that are not focused on images. Note that, for example, there exists subreddits (such as r/pics which focuses on pictures) that cannot be meaningful used for our project. In the last step, the function standardizes the explanatory variables (score, upvote ratio and number of comments). A discusses about the advantages of standardization follows in the next chapter.

4. Clustering

After pre-processing the data, the posts are then clustered. Clustering means that observations are grouped based on similarity in specific variables. One of the most common approaches is

K-means clustering. K-means clustering can be seen as an unsupervised algorithm. Our goal is to cluster different posts into K groups that are similar in terms of the number of comments, the number of upvotes and the upvote ratio. The idea behind K-means clustering is based on a minimization problem. Assign each observation to one of K groups, so that the total distance between all observations and the corresponding group's mean is minimized. The following chapters discuss some problems associated with K-means clustering and whether they are problematic in the project or not. The clustering step is done by the get_K-function and the cluster-function that can be found in the corresponding notebook.

4.1. Variables

We clustered Reddit posts based on three variables: number of comments, the score (i.e., number of upvotes minus number of downvotes) and the upvote ratio (number of upvotes divided by total number of up-/downvotes). The idea is that a higher number of comments represents higher attention of a posts. A high score implies that more people voted the post up then down. The upvote ratio has a similar interpretation, if the ratio is greater than 0.5 more people voted the post up than down. The number of comments takes on only non-negative integer values and can thus be seen as count data. The score variable has some problems associated with it. First note that there seems to be some inconsistency between the official Praw documentation (Boe, 2022) and an official explanation of the score by Reddit (Frequently Asked Questions, kein Datum). The Praw documentation denotes the score as the number of upvotes, while the Reddit FAQ explain score as the difference between up- and downvotes. We are going to interpret the score variable as the FAQ does, i.e., we treat it as the difference between up- and downvotes. This is supported by several discussions in the reddit community. Secondly, the score variable is fuzzed. That means, if there are more up- than downvotes, the score would be negative. If the score is negative, it will, however, be displayed as 0. This means that the score is a censored variable which might lead to bias. The upvote ratio should be relatively unproblematic. It can only take on values between 0 and 1. If the upvote ratio is greater than 0.5 there are more up- than downvotes. If it is less than 0.5 there are more down- than upvotes.

One general disadvantage of K-means clustering is that the method is based on minimizing a distance measure. Therefore, the method itself cannot be used if grouping should be done with respect to categorical values. Our project groups Reddit posts based on similarities in the number of comments, number of upvotes and the upvote ratio. All three variables are numeric. K-means clustering can therefore be applied.

4.2. Standardization

From a computational perspective, K-means clustering is a minimization problem. In a multidimensional case – such as in our project – different unit measurements of the variables can lead to biased clustering. The absolute distance between two observations' number of comments will likely be significantly greater than the absolute distance between the two upvote ratios. Such differences can have a massive impact on the clustering result. The three relevant variables are number of comments, score and the upvote ratio. First, note that the number of comments represents count data, i.e., these variables can only be a non-negative integer. The score could mathematically take on every integer value (both positive and negative). However, as discussed above, the score is censored, and negative values are replaced as 0. Thus, the score in our sample can also be seen as count data. The upvote ratio is defined as the number of upvotes divided by the total number of down- and upvotes. The number of down- and upvotes represent non-negative integer values. The upvote ratio is therefore between 0 and 1 as the number of upvotes is between 0 and the total number of votes. As the three variables have different unit measurements the data should be standardized, i.e., the variables are transformed so they have mean 0 and a variance of 1. Computationally, we can do this with the StandardScaler function in the Pyspark ML package which standardizes and normalizes the variables in the feature vector. The StandardScaler function is based on Z-score normalization method, which standardizes based on the mean and standard deviation of the sample. This method is especially appropriate if the minimum and maximum value of a variable is not exactly known (Usman & Mohamad, 2013). This standardization helps to omit some problems that could arise if K-means clustering is applied. Most importantly, as mentioned above, K-means clustering minimizes some distance measurement (e.g., the Euclidian distance). Different unit measurements have a significant impact on these distances and thus on the clustering results. This is not a problem if the variables are standardized. Note that the standardization part is included in the pre-processing-function that can be found in the corresponding notebook.

4.3. Number of clusters K

K-means clustering groups the data into K groups. One problem that arises with this method is the question about the optimal number of clusters K. There are two popular methods that could be used to find such a number K: the elbow method and silhouette analysis. For this project we used the silhouette method. Based on (Banerji, 2021) the "silhouette coefficient is a measure of how similar a data point is within-cluster (cohesion) compared to other clusters (separation)." This means, that in a first step, some value for K is chosen. The data is then clustered according to the chosen number of clusters K. Next, for each data point, the average difference between the point and all other points in the same cluster is determined. This difference is then compared to the average distance between the point and all points that do not belong to the same cluster. This procedure is repeated for different values of K. Mathematically, the silhouette score *S(i)* for a specific point i is defined as $S(i) = \frac{\mathrm{b(i)} - a(i)}{\max\{\mathrm{a(i), b(i)}\}}$. *b(i)* denotes the average difference from *i* to all points of different clusters. *a(i)* is the average difference between *i* and all points in the same cluster. The silhouette score is the mean of all individual silhouette scores *S(i)*. This score is restricted to lie between -1 and 1. A higher value implies that clustering based on the corresponding value of K is more efficient. Therefore, the algorithm tries out different values of K and chooses the value for K that maximizes the silhouette score.

One challenge that arises from K-means clustering is to determine the possible number of clusters. While the silhouette method gives a numerical decision rule to choose the optimal K, we first must decide for which values should a silhouette score be computed. We decided to restrict the number of clusters to lie between 3 and 10. This limitation should ensure that the algorithm is efficient with respect to computational time. In general, K means clustering is time consuming and limiting the maximum number of clusters K to 10 also limits computational time. This is especially important if the data would be huge (that would be generally the case if working with big data).

Finally note that the final number of clusters is dependent on the distance measure used. In this project we decided to use the squared Euclidean distance. This measurement is a special version of the Euclidean distance. In fact, the square roots in the calculation of the Euclidean distance are omitted. This version of the Euclidean distance is often more convenient to use as one computational step is omitted which might also improve the computation time of the algorithm. Finally, note that clustering itself can be seen as training a model. Therefore, results depend strongly on the method used.

Figure 1 plotted the silhouettes score against the number of clusters for r/elderscrollsonline. As argued above, the optimal number of clusters is determined by the maximum of the silhouette score. For r/elderscrollsonline the optimal number of clusters is 3.

Figure 2 shows the silhouette scores for r/Atlanta. The optimal number of clusters is 8 which maximizes the silhouette score.

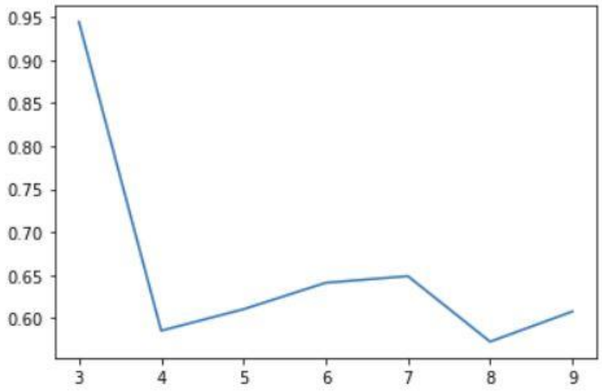

Figure 1: Silhouette score of r/elderscrollsonline

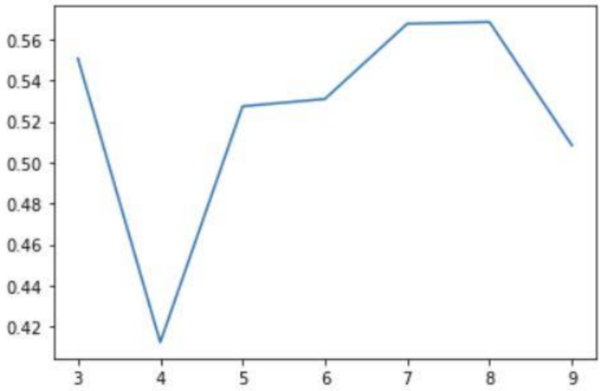

Figure 2: Silhouette score of r/Atlanta

Finally, we can have a look at the distribution across different clusters. Note that the label denotes the name of the clusters.

Label	**Count**
1 (most popular)	1
2	9
0 (least popular)	502

Figure 3: Distribution of clusters of r/elderscrollsonline

K-means clustering determines three groups for r/elderscrollsonline. Based on group means (see the corresponding notebook for the specific numbers) these three groups can be classified into most popular, popular, and least popular posts. Figure 3 indicates sources of bias. Nearly all observations are in the least popular cluster. Only a handful of posts is popular. This could result from outliers which might indicate that the sample size is too small. This problem might be solved by using a bigger sample which – as argued above – cannot be obtained because Reddit limits the request to the first 1000 posts. Because most of the training data is in cluster 0 estimates and models will certainly be meaningless. In fact, as we will see later, the models will predict nearly all test posts to lie in cluster 0.

Label	**Count**
5 (most popular)	1
7	2
2	3
4	22
0	49
3	212
6	154
1 (least popular)	52

Figure 4: Distribution of clusters of r/Atlanta

Posts of r/Atlanta are clustered in eight groups. Figure 4 gives an overview of the distribution across different groups. Like the clusters of r/elderscrollsonline the clusters are sorted from most popular to least popular. Note that this sorting was done manually based on the average number of comments and score. Score and comments were equally weighted for the sorting task. The specific numbers can be found in the accompanying notebook. For r/Atlanta the most popular posts are underrepresented which indicates the presence of outliers. These observations have a large impact on estimated models.

5. Prediction

Based on results obtained by clustering, we now want to predict to which group does a specific post (of the test set) belong. The used models use the body (i.e., the written text of post) as explanatories. For this project we decided to use and compare two different methods, logistic regression, and decision trees.

5.1. Logistic regression

One popular method for classification tasks is logistic regression. Logistic regression estimates the probability that a specific observation belongs to a category or not. Note that logistic regression is often called logit model and both terms will be used synonymous in the following paragraph.

In the case of predicting probabilities there is the problem that the dependent variable (the probability that an observation belongs to a specific group or not) is limited to lie between 0 and 1. Therefore, a linear regression model cannot be used and other methods – such as logistic regressions – must be applied. Based on (Wooldridge, 2016), the logit model uses the cumulative distribution function of a standard normal logistic distribution. This cumulative distribution function is limited to lie between 0 and 1 and can thus be used for modelling probabilities. The logit model can be used to decide between a yes or no decision. In our case, we are going to interpret a probability of more than 50% as success (i.e., the observation belongs to the group that is studied) and a probability of less than 50% as failure (i.e., the observation does not belong into the group that is studied).

One big disadvantage is, however, that logistic regression estimates the probability whether an observation belongs to a specific group or not. This means, that the logit model only answers a yes or no question if a specific group is given. In the case of only two clusters this is not a problem. The yes or no decision eventually leads to a decision whether the observation lies in the first group or the other one. In the case of multiclass classification – such as our task if the optimal number of clusters is greater than two – logistic regression cannot be applied directly because the model must distinguish between more than two clusters and thus the classification problem cannot be solved by a simple yes or no decision. However, logistic regression can still be applied in the case of more than two clusters, but it must be adapted.

First, for each observation we predict the probability of belonging into each single cluster. Based on the probabilities obtained, we can then check whether the logit model predicts if the post belongs to a specific cluster or not. If this procedure generates only one cluster per observation, everything is fine. There are, however, problems if an observation belongs to no cluster or to several clusters at the same time. Recall, that a probability of less than 50% indicates that the observation does not belong to a cluster. If all estimated probabilities for an observation are less than 50%, the post would belong to no cluster if this decision rule were applied. Therefore, for these cases we modified the decision-making process. In these cases, the cluster of an observation is determined by the highest estimated probability. This solves the problem that some posts might belong to multiple or no clusters. However, this decision-making process might also lead to bias. Consider, for example, a post that belongs to two different clusters (based on the estimated logit model). The post will be assigned to the cluster with the corresponding higher estimated probability. If the probabilities of the two clusters are, however, only marginally different it remains unclear why the chosen cluster is the optimal one. If some clusters get consistently only marginally smaller probabilities it might be possible that these clusters are underrepresented in the results.

Computationally, we first split the label column into several columns. Each of these columns represent a binary variable whether a posts belongs to a cluster or not. If the variable is 1, the post is in the cluster. If the variable is 0, the posts does not belong to that cluster. Secondly, a pipeline is created. The used function is taken from pyspark.ml. This takes the body (i.e., the written text of the post) which is then tokenized with the pyspark.ml.sql Tokenizer. In the next step stopwords are removed with the pyspark.ml.sql StopWordRemover. Stopwords are words that are independent of the content and are used for grammar reasons, examples would be “at” or “for”. This words alone have no specific meaning to them. They only become relevant if used together with nouns, verbs and adjectives. Next, the data is reshaped so it can be used for logistic regression. In fact, the code computes the inverse document frequency, i.e., the number of occurrences of a specific word in the code. This is done with the help of the pyspark.ml.sql CountVectorizer and IDF. This is used to predict the probabilities whether a post belongs in the different clusters. As argued above, the highest probability determines the cluster of a post. To do the logistic regression LogisticRegression is imported from pyspark.ml.classification.

Finally, we can have a look at estimates obtained by the two logit models.

Predicted clusters	**Count**
2 (most popular)	1
1	0
0 (least popular)	241

Figure 5: Predictions for r/elderscrollsonline obtained by logistic regression

Figure 5 shows an overview of predicted results for the test data on r/elderscrollsonline. Like the training data nearly all observations are predicted to be in the least popular cluster. Only one observation is predicted to be very popular. No observation is predicted to be in cluster 1. This indicates that the model will not give meaningful results. As anticipated in section 4.3 the poor clustering of the training data leads to problems within the estimated models. There is not enough data outside cluster 0 to get any meaningful results.

Predicted cluster	**Count**
5 (most popular)	14
7	6
2	8
4	48
0	25
3	58
6	26
1 (least popular)	36

Figure 6: Predictions for r/Atlanta obtained by logistic regression

The predictions for r/Atlanta look more promising. The predictions are distributed among all eight clusters. Based on the predictions there are some posts (in cluster 5, 7 and 2) that are predicted to be more popular than most of the posts in the test data.

5.2. Decision trees

Another model we implemented in our project is based on Decision Trees, using the algorithm available in the PySpark library. Decision Trees have the structure of a tree, which consists of decision nodes. These nodes represent a yes-or-not test, and leaf nodes, which represent a result, in our case classification. Decision Trees are one of the most popular ways to solve classification problems. Decision tree models have an advantage over logistic regression for our case since it provides predicted classes (clusters) without any extra actions. Moreover, it can handle not only categorical but also continuous variables. Finally, it is less a less greedy algorithm in terms of computational time. Based on decision trees there also exists random forest which combines several decision trees at once. While such a method improves the accuracy there is also a longer computational time associated with it. Decision trees have, however, some disadvantages. For instance, a Decision Tree model can be easily overfitted, so limiting the maximum depth of the tree is crucial.

We apply Decision Trees model to predict label by the “body” variable. However, some pre-processing steps must be done because the “body” consists of raw text. First, the Tokenizer function is imported from Pyspark.ml and implemented. This splits the text string into words, symbols, or other tokens. Secondly, the StopWordsRemover function is used to remove stop words. As explained above, stopwords are words that have no meaningful interpretation if they stand alone. In the next step we use the CountVectorizer tool provided by the scikit-learn library which is “used to transform a given text into a vector on the basis of the frequency (count) of each word that occurs in the entire text” (Verma, 2020). Furthermore, by using the IDF function we can get the inverse document frequency of the words. Our final column’s name is called “body_vectorised_features”.

After completing all data preparation steps, we are ready to build decision tree model over “body_vectorised_features” as the features column. Another crucial step is to set the maximum depth of the Decision Tree carefully to prevent overfitting and get the highest possible accuracy. One of the ways to find this parameter is the GridSearchCV function with built-in cross validation, which plots every tree with different parameters to find the optimum one. However, this method is known to be very time-consuming. Consequently, we decided not to implement this method and chose the maximum depth parameter by ourselves and set it to 6. This depth should ensure that the algorithm can be run in a satisfactory time. After running the DecisionTreeClassifier function a pipeline is created to reduce computational time and the memory required to cache a table function’s results. After that, the model was trained on the training data and the predictions based on the test data are returned. The final output “predictions” is a data frame, which consists of "body", "num_comments", "score", "upvote_ratio", "features", "features_scaled", "prediction" and “rawPrediction" columns.

Predicted clusters	**Count**
2 (most popular)	2
1	0
0 (least popular)	240

Figure 7: Predictions for r/elderscrollsonline obtained by decision trees

The predictions for r/elderscrollsonline obtained by the decision tree model follow a skewed distribution. 240 posts are predicted to be in cluster 0 and only 2 outcomes are predicted to be in cluster 2. These results are really like the ones obtained by logistic regression and indicate that the model (based on the collected data) may not be used for any meaningful interpretations. Nearly every post about Elder Scrolls Online seems to be not popular.

Predicted cluster	**Count**
5 (most popular)	0
7	0
2	0
4	1
0	25
3	39
6	155
1 (least popular)	1

Figure 8: Predictions for r/Atlanta obtained by decision trees

The results for the Atlanta subreddit are completely different from the Elder Scrolls Online ones. The are more equally distributed but the predictions are still skewed. Most posts are predicted to be in the most unpopular cluster. No post is predicted to be in the three most popular clusters. The majority of collected posts seems to rather unpopular based on the decision tree model.

6. Comparing the models

In general, models can be compared by many different metrics. Among the most common ones are accuracy, true and false positives or the receiver operating curve (ROC). The problem of these methods is that they require the true label of the observations in the test data, i.e., to compute these metrics the data must contain the true cluster of the test observations. Our project has the problem that labels are not automatically provided by the data generating process. The labels must be determined by clustering which itself depends on a model. This means that there is no efficient way to assign the true labels to the test observations. Therefore, several methods – such as accuracy or ROC – cannot be applied with the predictions obtained by our algorithms. To still get some information on the effectiveness of our models we will first compare the results obtained by the two machine learning methods. Comparing both predictions for the test observations from r/elderscrollsonline we discover that approximately 98.76% of the posts in the test set are predicted to be in the same cluster by both methods (i.e., by logistic regression and decision trees). There seems to be no big difference between the estimates obtained by the two models. Note that – as argued above – this might result from the data generating process because there are not many posts in the popular clusters. Based on the results obtained by clustering the training set certainly has some issues which eventually return biased models.
The predictions obtained for r/Atlanta differ between the two machine learning methods. Only 23.98% of all posts get the same predictions by logistic regression and decision trees. Thus, in a next step we will discuss which model gives more accurate results.

Since we do not have data on the true clusters of the test set, we are using the Euclidean distance as a measurement to discuss which model gives better predictions. Mathematically, each observation can be represented as a point in a three-dimensional space (the variables are the number of comments, score and upvote ratio). We can now compute the distance between two points. To do this, we started by standardizing the number of comments, score and upvote ratio for each post. Secondly, we compute the corresponding standardized mean values for each cluster based on the clustered training set. Note that the mean values of the number of comments, score and upvote ratio are the predicted values for an observation in the test set (based on the predicted cluster). Then the Euclidean distance (i.e., the square root of the sum of the squared differences between each variable) is computed for each observation. Finally, the individual distances are summed up. This sum can be used to compare different models. The lower this measurement, the better the model. The predicted features (based on the cluster mean of number of comments, score and upvote ratio) are more accurate for the model that has the lower sum of Euclidean distances.

For the predicted values regarding r/elderscrollsonline, we get a sum of Euclidean distances of 273.42 for logistic regression and 290.43 for decision trees. The results obtained by the two methods are not much different. This is supported by the fact that over 90% of all predictions are the same for both models. Both models predict the number of comments, score and upvote ratio equally well.

As discussed above, the two machine learning models give different predictions for r/Atlanta. To compare the logit model with the decision tree model we compute the sum of Euclidean distances. The logistic regression returns a sum of Euclidean distances of 671.08. The decision tree model performs better, the sum of Euclidean distances is 270.02. For the data collected from r/Atlanta the decision tree model predicts the number of comments, score and upvote ratio with a higher accuracy then the logit model.

7. Licences and legal concerns

The data is collected, stored, and processed in accordance with the Reddit API terms of use (Reddit API Terms of Use, 2016)and the general Reddit user agreement (Reddit User Agreement, 2021), as well as the Reddit Privacy Policy (Reddit Privacy Policy, 2021), the first also binds us to the latter two. Fortunately, all three agreements are accessible and data-scientist-friendly, reserving some rights.

Regarding the API Terms of Use, point 2.d is worth emphasizing. According to this paragraph, user-content (such as text, photos, and videos) are owned by the respective user, not Reddit itself, which may come as a surprise to those who have looked at similar agreements for other social media websites. Nonetheless, users give up some rights to their content by agreeing to the terms of use, in the form of a license that “includes the right for us to make Your Content available for syndication, broadcast, distribution, or publication by other companies, organizations, or individuals who partner with Reddit” (Reddit API Terms of Use, 2016). Reddit may thus transfer elements of the license to API users, in this case individuals who partner with them.

The aforementioned paragraph 2.d accomplishes this by stating: “Subject to the terms and conditions of these Terms, Reddit grants You a non-exclusive, non-transferable, non-sublicensable, and revocable license to copy and display the User Content using the Reddit API through your application, website, or service to end users. You may not modify the User Content except to format it for such display. You will comply with any requirements or restrictions imposed on usage of User Content by their respective owners, which may include "all rights reserved" notices, Creative Commons licenses or other terms and conditions that may be agreed upon between you and the owners" (Reddit API Terms of Use, 2016).

The Privacy Policy (Reddit Privacy Policy, 2021) itself states "Much of the information on the Services is public and accessible to everyone, even without an account. By using the Services, you are directing us to share this information publicly and freely." It should be no surprise to users that others may access their posts – this is in fact the purpose of Reddit.

Furthermore, applicable licenses may depend on the person that submitted a post, and some content may not comply with Reddit's policies, e.g., if a post infringes a copyright or trademark (Reddit User Agreement, 2021). It would be challenging for us to scan for such issues, but fortunately, the platform already does this automatically by using its more impressive resources. Since we exclude new posts from our API requests, any infringing content is likely to be removed before it is collected.

We also examined the legal framework beyond these agreements, namely whether the European General Data Protection Regulation applies to this project. The GDPR is concerned with personal data which it defines as "any information which are related to an identified or identifiable natural person" (GDPR: Personal Data, kein Datum). Otherwise, the data would be anonymous, and "this regulation does not (…) concern the processing of such anonymous information, including for statistical or research purposes" (GDPR: Personal Data, kein Datum). As our data is collected without usernames, any point of identifiability would be located within the body of text, and it is only this possibility that may ties us to GDPR. Although we have no specific mechanism in place to inform these users or receive their data deletion requests there should be any problem arise from the GDPR. First, the GDPR provides a 30-day window to complete such requests and, secondly, the Reddit data will be deleted shortly after collection.

Regulations also aim to assure that no more data is collected than necessary to fulfil the goal at hand, and by extension, that goal is stated specifically. Although our goal may be viewed as quite general within the confines of Reddit, we have done our best to render it more concrete, such as by stating to Reddit which subreddits are to be analyzed. Furthermore, the issue is sidestepped by collecting also publicly available data. How this data collection was further limited is discussed below.

7.1. Privacy Concerns

To comply with API terms of use, our own privacy policy (disclosing how we "collect, use, store, and disclose data collected from visitors, including, where applicable, that third parties (including advertisers) may serve content and/or advertisements and collect information directly from visitors" (Reddit API Terms of Use, 2016)) has been made publicly available and states (Gstettner, Malysh, Scholz, & Tallai, 2021):

"For the purposes of a university project, we will collect information about the content of posts, the subreddit they are posted in, many comments and how much karma said posts receive, as well as the upvote ratios. Post titles and usernames will not be collected and usernames in the text will be censored. The following two subreddits will be analyzed: /r/Atlanta and /r/elderscrollsonline.

The posts will be categorized according to popularity (based on karma and replies). The content of the posts will be analyzed via computer learning and used to predict popularity.

The data will be stored on students' personal computers, and servers hosted by the Vienna University of Economics and Business. Subsequently the data will be shared with Vienna University of Economics and Business. The raw data collected by the API will be stored for no more than 30 days.

The data will be used for a university project dealing with the scalable handling of big data and understanding legal fundamentals and ethical frameworks in dealing with data in an international context. Results will be submitted to the university and presented in class. The purpose of this project is educational, and the collected data will not be used for business purposes."

The policy aims to answer transparency-relevant questions: "What data is collected?", "What is the purpose of data collection and processing?", "How is data processed?", "Where is the collected data stored?", "How long is the data stored?" and "Is there a disclosure to third parties?"

A notice was added to the notebook that is submitted alongside this paper. It states that whoever runs the code to access the Reddit API agrees to our privacy policy and Reddit's terms of use.

We do not take matters of online privacy lightly. The data employed is **publicly available** and Reddit posts do not habitually contain explicit personal information: the data should be quite anonymous in the sense of not being tracible to real people. However, we aimed at a higher bar than "quite" anonymous. We also do not mean to negate further discussions of privacy because our project is based on publicly available data. As shown by social engineers who employ such data to get more confidential information, quite a lot can be accomplished by aggregating such publicly available data.

One option to meet this higher standard would be to employ so-called synthesized data, generated by an engine that preprocesses the gathered data, builds a model of it using deep neural networks and then uses to model to generate data sets that are not real, but are highly realistic, showing the same mathematical properties – analysis on the synthesized dataset should yield the same results as models on the original data while alleviating privacy concerns (Bellovin, Dutta, & Reitinger, 2019).

Ultimately, we decided to achieve complete anonymity via other means. Although concepts such as K-Anonymity are hard to apply to unique products such as Reddit posts, we simply opted not to collect the username of the person who uploaded a post, and to eliminate usernames from the bodies of the posts –this is made easy by all redditor names being prefaced with either "u/" or "/u/". Without these identifiers, the posts of users may not be compiled to analyze the behavior of individuals. Note that even if we wish to concentrate on what was said and not who said it (i.e., potential subreddit celebrities, users with special flairs/identifiers), this step may have omitted one relevant variable for explaining the popularity of Reddit posts.

The final list of parameters gathered is thus brief, consisting only of the body of the post, creation time, and the three explanatories: number of comments, score and upvote ratio. The latter three are straightforward metric variables, and time does not factor into the final analysis. Time only serves to filter out new posts. The body of the post is more varied and may reflect a broad range of attitudes, beliefs, personalities, and backgrounds.

Through these measures and the overall design of the data collecting and preprocessing algorithm, we have also avoided two ways that are sensitive to privacy concerns: aggregating data about people from various sources (which is beyond the scope of this project as we are focusing on Reddit posts only) and deducing information about people (such as a pregnancy) if they have not declared it themselves in the collected data (John, Kim, & Barasz, 2018). For example, one could run a machine learning technique to predict whether a person is pregnant or not based on a Reddit post. Once again, the emphasis is only on using data that publicly shared. Furthermore, we are not combining subsequent data from individuals to profile them (such data might inadvertently reveal private information).

Additionally, gathered data is saved locally as csv-files that are deleted shortly after finishing the project (once the data does not serve any purpose). Any residual personal details will be treated confidentially and not disclosed to people outside of this course.

7.2. Ethical concerns and implications

Several ethical codes have been published for data scientists, such as the *ACM Code of Ethics and Professional Conduct,* or the *IEEE Code of Ethics,* but for our purposes we will focus on the ideas of Wil van des Aaist and Cathy O'Neil.

In his presentation, *Responsible Data Science in a Dynamic World*, Wil van des Aaist said "If data is the new oil on which our society runs, then we should take care of data-related forms of pollution" (van der Aalst, 2019). These are non-transparency, unfair use of data, spurious correlations, privacy violations and bogus conclusions. Let us examine how the project does with respect to these pitfalls:

- Non-transparency: our methods and goals are outlined both here and in the privacy policy that was submitted to Reddit.

- Unfair use of data: data is used as prescribed by the applicable licenses and regulations.

- Spurious correlations: although our ability to control for confounding variables is limited due to the small number of variables collected (this small sample will lead eventually lead to biased models but will honor privacy concerns as only minimal data is collected), we are unmindfully claiming causation if in truth there might only be a correlation.

- Privacy violations: as already stated, all data is publicly available.

- Bogus conclusions: we hope our conclusions are measured and consider several viewpoints.

The Article *The Ethical Data Scientist* by Cathy O'Neil states that a data scientist does not have to be an expert on the social impact of algorithms, but rather:

-A facilitator of ethical conversations

-A translator of the resulting ethical decisions into formal code

-Raise the questions with a larger and hopefully receptive group

While social scientists may dedicate long sections to the potential social impact of similar projects, O'Neil lifts the responsibility of an exhaustive analysis from our shoulders. Several questions arise nonetheless, not only regarding privacy, but also the ramifications of an overreliance on success metrics in shaping intellectual products such as a text, even a simple forum post.

It might be said that such popularity analyses help people game the system. One example of this would be YouTube channels posting videos of a certain length, at certain times and intervals, using certain times and thumbnails, not just to bait clicks, but in hopes of exploiting YouTube's ever-changing algorithm and getting it to show people the video in the first place. However, while Reddit may be used for advertising purposes, posts are not monetized the way YouTube videos are. We would also argue that our analysis does not have strong implications in this regard, because it is focused on a substance-over-style examination of the contents of a post, even excluding titles, which would serve as the potential clickbait element of a post.

Instead, we wish to initiate a conversation on the kind of content people value and perhaps help less successful redditors engage with their communities. How posting trends may change with introspection and fashion cycles is outside the scope of this paper but can also form part of the emerging discussion.

Although we did not run a sentiment analysis, the "undesirable conclusion" that negative posts garner more attention is always a possibility. If true, the application of such a framework may backfire in terms of improving online discourse but could still bring attention to the issue. One factor mitigating such a conclusion is that the text of deleted posts, whose deletion may be connected to negative attention they received (negative attention is still attention) could not be accessed and were thus not included in the samples.

Furthermore, while our data is not about people, we are mindful of the power of language and any discrimination its use might lead to. We are using natural language-processing methods and any biases of the script can affect our predictions e.g., not recognizing words used exclusively or mostly by minorities or more generally, members of cultures other than the American/international mainstream which dominates Reddit. While accessible by the Reddit API, the username of the poster will not be included as predictor. Other than privacy, the variable is also omitted because it might be possible to infer gender and race from the username which might lead to discrimination and bias. Although the language-processing issues remain, this improves problems regarding profiling and leads towards algorithmic fairness.

8. Additional Biases

Several biases have already been documented in previous sections where they tied in naturally (for example, regarding clustering, logistic regression or usernames). This part aims to make the list more comprehensive, though some biases will no doubt remain unidentified.

The project aims to predict the popularity of posts based on content, putting substance to the forefront of our analysis, but disregarding the title in the process, which undoubtedly plays a role in catching people's attention and contributing to popularity, even if actual votes and comments happen after exposure to the post itself. Furthermore, our algorithm does ignore pictures and the contents of any links that might be included in a post. These elements may have a substantive impact on post popularity.

The dimension of time also raises several issues. First, for training the algorithm we only have access to historic data, no matter how recent the data is. Therefore, there might be a bias as historic data might not reflect future events, and the trends identified may be short-lived. The small timeframes (and samples – although both are scalable for later projects) employed might not represent the true preferences of the population. What is more, we do not account for the time when a post was submitted. A subreddit may be more active at certain times of the day and different types of post could garner different levels of attention in the morning or evening. Lastly, while new posts are excluded and the remaining observations are thus more evenly positioned in their life cycles, this method is imperfect, and some differences remain.

Moreover, as the algorithm analyses one subreddit at a time, the result (i.e., the predicted model) is not applicable to other subreddits. The data collected is only a subset of the population (all Reddit posts of all subreddits) and thus not representative. If one wants to analyse a post published in another subreddit, the algorithm must train the model on new data (collected from the new subreddit) first. Different subreddits inherently focus on their designated topics and different bubbles and echo champers might develop within them. Therefore, the trained models heavily depend on the chosen subreddit.

9. Experience gained

9.1. Mark Tallai (Legal & ethical aspects, privacy, biases)

Copyright and other licenses have been challenging to enforce online and remained outside the purview of many users and the corresponding body of regulations and user agreements continued to grow. It is incredibly easy to violate copyright (and now -with GDPR- privacy) on the internet out of simple ignorance. The experience of carefully reading and complying with Reddit's various user agreements –even though they are quite forgiving compared to e.g., Twitter- has done much to raise our awareness and sense of responsibility in this regard.

I also found the various ways of rendering data anonymous, including K-Anonymity and Synthetization, fascinating. Although these methods may not be aimed at projects with such a limited range of data collection, I look forward to making use of them in the future.

The topic of data privacy has interested me for some time, the concept of algorithmic fairness and its societal consequences has provided me with an even greater appreciation of how carefully algorithms must be formulated to assure that everyone is given a fair chance, be it concerning something as important as employment or something as small as being offered a discount at a diner. Although staring at the full range or societal implications can be intimidating for any project, we hope we have done a reasonable job of mitigating similar biases, or at least pointing them out.

9.2. Felix Gstettner (Finding out how the reddit API works, Logistic regression, computing Euclidian distances, helping with model evaluation)

The most interesting thing I learned was how to access reddit's API. Not just how to code the access but also what to do with a reddit account to even make the access possible. Furthermore, I also learned how to find out what conditions applied for using the API by fining and navigating Reddit's three or more web pages that governed the use of APIs on Reddit. In the process I realized that I could set up a public git repository on GitHub to host a publicly available privacy policy, a requirement of reddit's API policy.

I learn how to use pyspark not only in theory but also in practice. I realized quite late that logistic regression is unable to predict more than two distinct categories, a lesson I will never forget.

9.3. Martin Scholz (Collecting and preprocessing the data, clustering, model evaluation)

It was interesting to work with text data. Most of my other courses (especially finance and business mathematics) focus on numerical data. Working with text broadened my knowledge. Furthermore, encountering several challenges taught me new skills. While working with the Reddit API we experienced several different error messages. Some of these resulted from server failures of Reddit, others were our own fault. Troubleshooting itself was not always funny but I learned several important facts to consider in future works. One small, but still important, problem I encountered resulted from locally saving the data frame. While a post itself is stored correctly as one variable in a pyspark data frame even if commas and paragraphs are present, this elements lead to problems when the data frame is saved as a csv-file. I will take this issue in mind for my future work.

By reading into K-means clustering I was able to build a solid understand of clustering. Furthermore, I now understand several problems that can arise if the data does not fit perfectly for clustering. I also learned about limitations and advantages of different clustering methods that might be useful for my future career.

The project also deepened my knowledge in pyspark and machine learning in general. Nearly every step that had be undertaken resulted first in an error message. By solving any upcoming problem, I learned way more than I would have expected.

9.4. Stepan Malysh (Decision Tree model, text filtering)

Since for this course we choose project topic by ourselves, it could not be uninteresting. Firstly, at the very first step of the project, while choosing ideas, I realized, how many unexpected pitfalls can occur, for instance, websites APIs differ a lot and not that much data is public, and some of the ideas were dropped because they ended up being much harder and resource consuming than anticipated.

Secondly, sticking to pandas for months made me think that working with PySpark is harder than it turned out to be. I enjoyed putting my knowledge obtained in the classes into practice.

Finally, I have learnt the lesson that communication is half of the success of the project, we set deadlines and held meetings on a regular basis, discussing the work done and the difficulties encountered.

10. Challenges encountered and recommendations for future work

Working with big data proved to be challenging. However, by overcoming several challenges we were able to deepen our knowledge. Some challenges are noteworthy, and the knowledge gained by solving them could proof to be helpful for future work.

Especially interesting were the Reddit API terms of use. To comply with these, we had to write our own privacy policy. Before this project we never thought about how to write such a policy that complies with general data protection laws.

First, using praw to access the Reddit API returned a wide range of error messages. As already noted above there was the case that the Reddit servers were down (which gave an HTTP response error) or the case that the resulting data frame was empty because all posts were filtered out based on their upload time. The problems encountered by saving the data locally (in fact the existence of commas and paragraphs) showed that tiny details can lead to huge problems. This taught us that it is especially important to consider every possible scenario when designing a general algorithm.

Working with logistic regression showed that this method is not the most suitable machine learning tool for multiclass-classification problems. To classify more than two categories the code had to be adapted. For future work it might be more efficient to use more suitable methods depending on the given task. For this, we recommend, to look up necessary details before deciding to use a specific machine learning model.

Additionally, we also learned that creativity plays a huge role for programming a machine learning algorithm. We made the mistake that we did not think about assessing the model accuracy when we wrote the project proposal. The way we decided to do predictions resulted in the problem that we had to deal with unlabeled data. Therefore, we could not use many model diagnostic metrics because we did not know the true cluster of an observation. Thus, we thought of alternative ways that may assess the accuracy of a model. Our results were then based on the Euclidean distance measurement. Showing creativity is recommended for future work. By being open-minded and thinking of alternative solutions many problems can be solved or omitted.

Lastly, the project showed that estimates obtained by both logistic regression and decision trees heavily depend on the training data. As the discussion about the clustered training data of r/elderscrollsonline shows, data issues have a huge impact on trained models. For future work it is recommended to first, use a bigger sample size and secondly, assure that the data itself is not biased in any other way.

11. Conclusion

This project explored two machine learning methods based on Reddit posts. The project shows that data collection is extremely important to get any meaningful predictions. Model accuracy can heavily depend on the presence of outliers. Our analysis cannot give a clear solution to this problem. One possible solution would be to run the analysis again with a bigger sample size and check the presence of outliers again.

Since the collected data was not label it is extremely hard to find out how well a model predicts the popularity of Reddit posts. However, our results show that for r/Atlanta the decision tree model returns better estimates than logistic regression. Before applying the models more general, Additional metrics based on other subreddits and bigger sample sizes would need to be compared.

The licensing and legal aspects of this endeavor were supported by Reddit's liberal approach to granting access to their API. Great care was taken to alleviate any concerns related to privacy and ethics, and while biases no doubt remain the project may form a basis for further discussion and refinements of our approach.

12. Bibliography

Banerji, A. (2021, May 18). *K-Mean: Getting The Optimal Number Of Clusters*. Retrieved January 29, 2022, from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2021/05/k-mean-getting-the-optimal-number-of-clusters/

Bellovin, S. M., Dutta, P. K., & Reitinger, N. (2019). Privacy and Synthetic Datasets. *Stanford Technology Law Review, 22*(1). Retrieved from https://law.stanford.edu/wp-content/uploads/2019/01/Bellovin_20190129.pdf

Boe, B. (2022). *Submission*. Retrieved 01 14, 2022, from PRAW: The Python Reddit API Wrapper: https://praw.readthedocs.io/en/latest/code_overview/models/submission.html

Frequently Asked Questions. (n.d.). Retrieved 01 14, 2022, from Reddit: https://www.reddit.com/wiki/faq#wiki_how_is_a_submission.27s_score_determined.3F

GDPR: Personal Data. (n.d.). Retrieved from intersoft consulting.

Gstettner, F., Malysh, S., Scholz, M., & Tallai, M. (2021, December 30). *Privacy_policy*. Retrieved January 1, 2022, from GitHub repository: https://github.com/hellfun1/Privacy_policy/blob/main/privacy_policy.txt

John, L., Kim, T., & Barasz, K. (2018). *Ads That Don't Overstep*. Retrieved from Harvard Business Review: https://hbr.org/2018/01/ads-that-dont-overstep

Reddit API Terms of Use. (2016, May 25). Retrieved from Reddit: https://docs.google.com/forms/d/e/1FAIpQLSezNdDNK1-P8mspSbmtC2r86Ee9ZRbC66u929cG2GX0T9UMyw/viewform

Reddit Privacy Policy. (2021, August 12). Retrieved from Reddit: https://www.redditinc.com/policies/privacy-policy-september-12-2021

Reddit User Agreement. (2021, August 12). Retrieved from Reddit: https://www.redditinc.com/policies/user-agreement-september-12-2021

Usman, D., & Mohamad, I. B. (2013). Standardization and Its Effects on K-Mean Clustering Algorithm. *Research Journal of Applied Sciences, Engineering and Technology, 6*(17), 3299 - 3303. doi:http://dx.doi.org/10.19026/rjaset.6.3638

van der Aalst, W. M. (2019). Responsible Data Science in a Dynamic World. *In: Strous L., Cerf V. (eds) Internet of Things. Information Processing in an Increasingly Connected World. IFIPIoT 2018. IFIP Advances in Information and Communication Technology, 548*.

Verma, K. (2020, July 17). *Using CountVectorizer to Extracting Features from Text*. Retrieved from GeeksForGeeks: https://www.geeksforgeeks.org/using-countvectorizer-to-extracting-features-from-text/#:~:text=CountVectorizer%20is%20a%20great%20tool,occurs%20in%20the%20entire%20text.

Wooldridge, J. (2016). *Introductory Econometrics: A Modern Approach* (7 ed.). Cengage Learning Inc.