ITM 760:
Course Project Report

Written by: Jacqueline Chung

Prepared for: Dr. Mehdi Kargar

Table of Contents

Introduction

The learning objective of this project is to gain hands-on experience using Python (one of the most popular tools for data mining and machine learning) to build models from real-world datasets. You will also evaluate different data mining algorithms in terms of accuracy and run time.

The report's objective is to apply exploratory analysis on the two datasets, "Online News Popularity" and "Online Shoppers Purchasing Intention", and three classification algorithms, SVM, k-Nearest Neighbour, and Decision Tree.

This report will be presenting the following:

1. Data exploration including correlation analysis
2. Data cleaning (feature selection) and transformation based on the exploratory analysis
3. Visual models from the three learning methods
4. Evaluation of each method
5. Discussion of the evaluations
6. Conclusion and summary of the main results

I will be using the following Code on Google Colab to model the two datasets.

Data Exploration and Data Cleaning

The steps that will be followed:

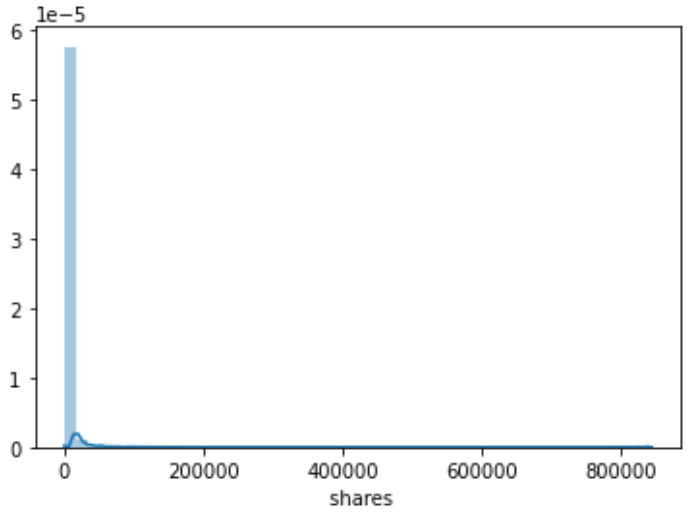
1. Descriptive statistics summary (i.e. histogram, boxplot, correlation matrix, scatter plot)
2. Delete irrelevant data
3. Remove outliers
4. One hot vector encoding (if needed)
5. Normalization
6. Descriptive statistics summary (again)
7. Filter correlation
8. Filter information gain

News

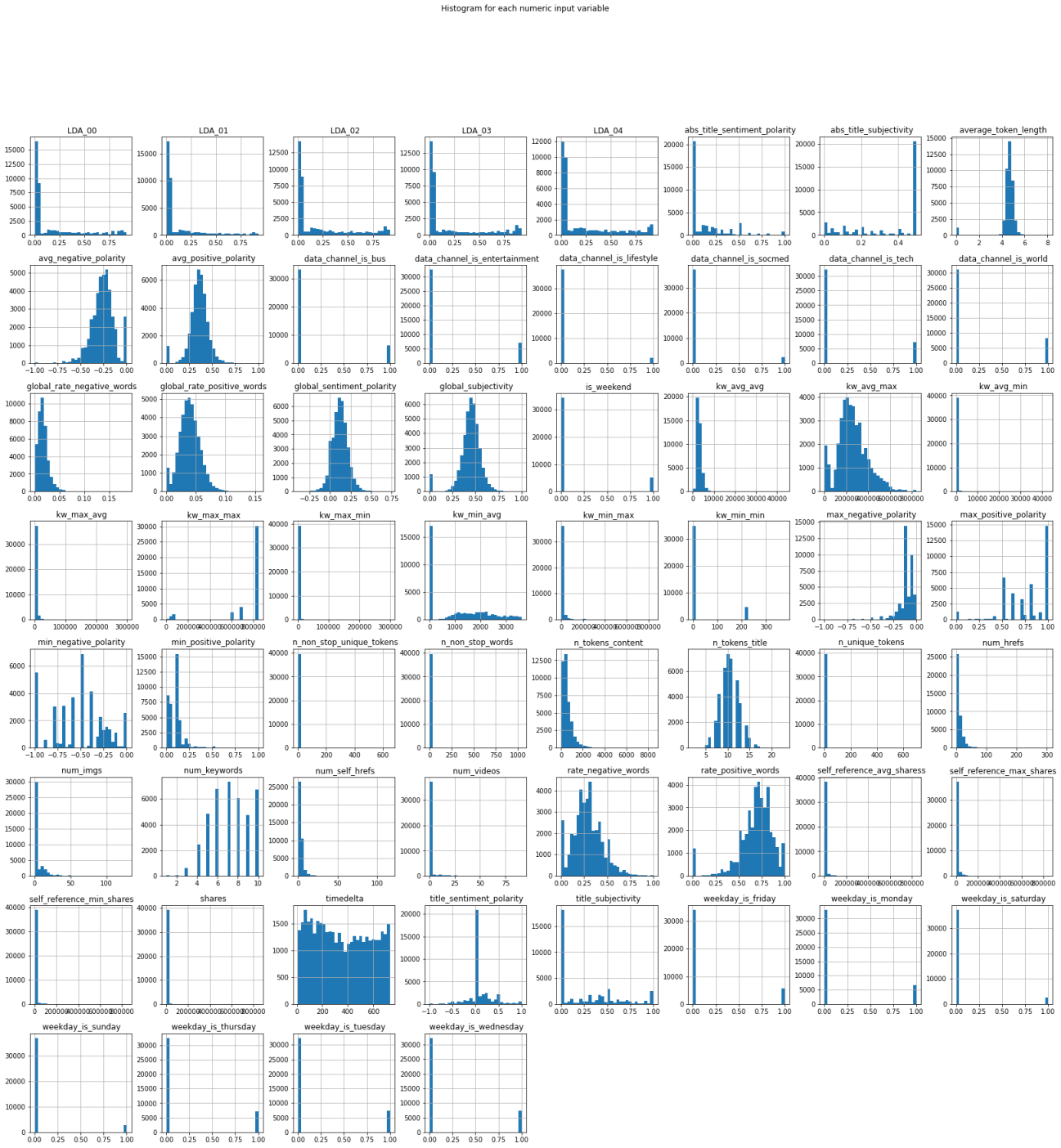
Having 39644 rows shows there is a lot of data needed to explore. This is a regression problem because we are estimating the variable, “shares”. Exploring the target class, the mean (average) is 3395.38. However, the minimum and maximum are 843300 and 1 respectively. This has large discrepancies from the mean.

```
count     39644.000000
mean       3395.380184
std       11626.950749
min           1.000000
25%         946.000000
50%        1400.000000
75%        2800.000000
max      843300.000000
Name:  shares, dtype: float64
```

Using a histogram, this shows the deviation from the normal distribution (skewness) and peakedness. We can see that the shares are skewed toward the left of the chart, so this means that although there is a large range, the majority of the data is less than 5000.



Below is a diagram of another type of histograms for all the attributes in the dataset:



This represents the frequency of the values of each attribute. Many attributes are skewed to one side. Other attributes have a clear average or median. When there is infrequent data, this indicates that it will need to be removed. The infrequent values can also show that the attribute has outliers that go outside of the trend because it can make the classifiers more bias.

Furthermore, it is clearly shown that this dataset has a large skewness, looking at the image below:

```
Skewness: 33.963885
Kurtosis: 1832.672657
```

The Kurtosis[1] further clarifies this skewness as it significantly larger than 0. This shows it has a leptokurtic distribution.[2]

After exploring the “url” attribute, it is a string of characters that is not useful for understanding the “share” target class. As a result, it is removed.

After exploring the data, the data is normalized since most of attributes are numerical. Normalizing the data reduces the impact the data will have.

	timedelta	n_tokens_title	n_tokens_content	n_unique_tokens	n_non_stop_words	n_non_stop_unique_tokens	num_hrefs	num_self_hrefs	num_imgs	num_videos	average_token_length	num_keywords
0	0.520705	0.076250	-0.038649	0.000165	0.000003	0.000194	-0.022644	-0.011152	-0.027689	-0.013735	0.016430	-0.247085
1	0.520705	-0.066607	-0.034401	0.000081	0.000003	0.000158	-0.025933	-0.019773	-0.027689	-0.013735	0.045450	-0.358196
2	0.520705	-0.066607	-0.039593	0.000038	0.000003	-0.000039	-0.025933	-0.019773	-0.027689	-0.013735	-0.019259	-0.135974
3	0.520705	-0.066607	-0.001831	-0.000063	0.000003	-0.000036	-0.006196	-0.028393	-0.027689	-0.013735	-0.017825	-0.024863
4	0.520705	0.123869	0.062011	-0.000189	0.000003	-0.000228	0.026698	0.135400	0.120749	-0.013735	0.016738	-0.024863
...	...	...	...	...	...	...	...	...	...	...	...	...
39639	-0.479295	0.028631	-0.023662	-0.000027	0.000003	-0.000007	-0.006196	0.031951	-0.027689	-0.002746	-0.003124	0.086248
39640	-0.479295	0.076250	-0.025786	0.000211	0.000003	0.000301	-0.006196	0.031951	-0.012064	0.513738	-0.017752	-0.024863
39641	-0.479295	-0.018988	-0.012334	-0.000045	0.000003	-0.000069	0.043146	-0.019773	0.058249	-0.002746	0.065744	0.086248
39642	-0.479295	-0.209464	0.015988	-0.000012	0.000003	0.000005	-0.002907	-0.019773	-0.027689	-0.013735	0.053079	-0.247085
39643	-0.479295	-0.018988	-0.045966	0.000219	0.000003	0.000242	-0.032512	-0.019773	-0.035501	0.008243	-0.009563	-0.358196

As shown, the data is between -1 and 1.

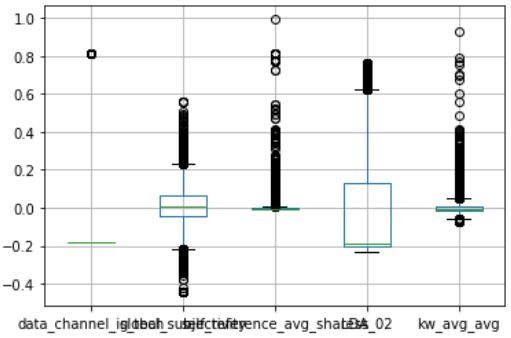
	timedelta	n_tokens_title	n_tokens_content	n_unique_tokens	n_non_stop_words	n_non_stop_unique_tokens	num_hrefs	num_self_hrefs	num_imgs	num_videos	average_token_length
count	3.964400e+04	3.964400e+04	3.964400e+04	3.964400e+04	3.964400e+04	3.964400e+04	3.964400e+04	3.964400e+04	3.964400e+04	3.964400e+04	3.964400e+04
mean	-2.556098e-15	5.510227e-16	-2.121137e-17	6.315895e-18	8.142612e-19	-1.168266e-17	7.864628e-17	-1.108895e-16	1.061377e-15	6.017054e-18	-3.572075e-15
std	2.962154e-01	1.006684e-01	5.559447e-02	5.022408e-03	5.020375e-03	5.022794e-03	3.727637e-02	3.323398e-02	6.491745e-02	4.514126e-02	1.050055e-01
min	-4.792953e-01	-3.999404e-01	-6.449312e-02	-7.820481e-04	-9.563038e-04	-1.060270e-03	-3.580161e-02	-2.839343e-02	-3.550112e-02	-1.373488e-02	-5.655935e-01
25%	-2.635276e-01	-6.660709e-02	-3.546315e-02	-1.103357e-04	3.389086e-06	-9.759383e-05	-2.264372e-02	-1.977274e-02	-2.768862e-02	-1.373488e-02	-8.684330e-03
50%	-2.148060e-02	-1.898804e-02	-1.622784e-02	-1.282478e-05	3.389089e-06	2.001222e-06	-9.485822e-03	-2.531365e-03	-2.768862e-02	-1.373488e-02	1.440557e-02
75%	2.592940e-01	7.625005e-02	2.000062e-02	8.627665e-05	3.389090e-06	1.006988e-04	1.025102e-02	6.089324e-03	-4.251116e-03	-2.745867e-03	3.812698e-02
max	5.207047e-01	6.000596e-01	9.355069e-01	9.992180e-01	9.990437e-01	9.989397e-01	9.641984e-01	9.716066e-01	9.644989e-01	9.862651e-01	4.344065e-01

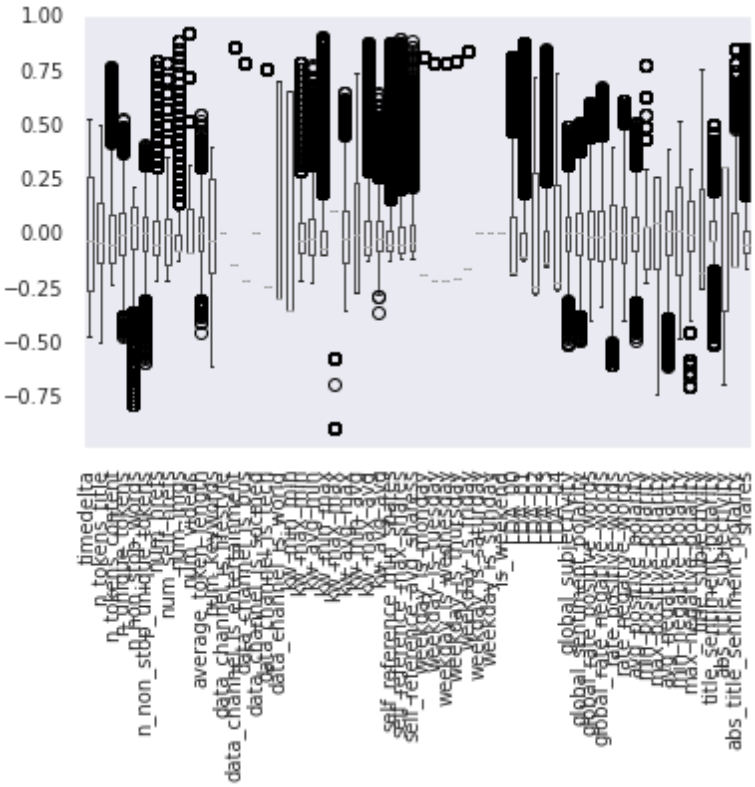
The target class:

```
count    3.964400e+04
mean    -3.655756e-18
std      1.378746e-02
min     -4.025121e-03
25%     -2.904522e-03
50%     -2.366160e-03
75%     -7.060132e-04
max      9.959749e-01
Name: shares, dtype: float64
```

[1] The sharpness of the peak of a frequency-distribution curve.
[2] https://www.spcforexcel.com/knowledge/basic-statistics/are-skewness-and-kurtosis-useful-statistics

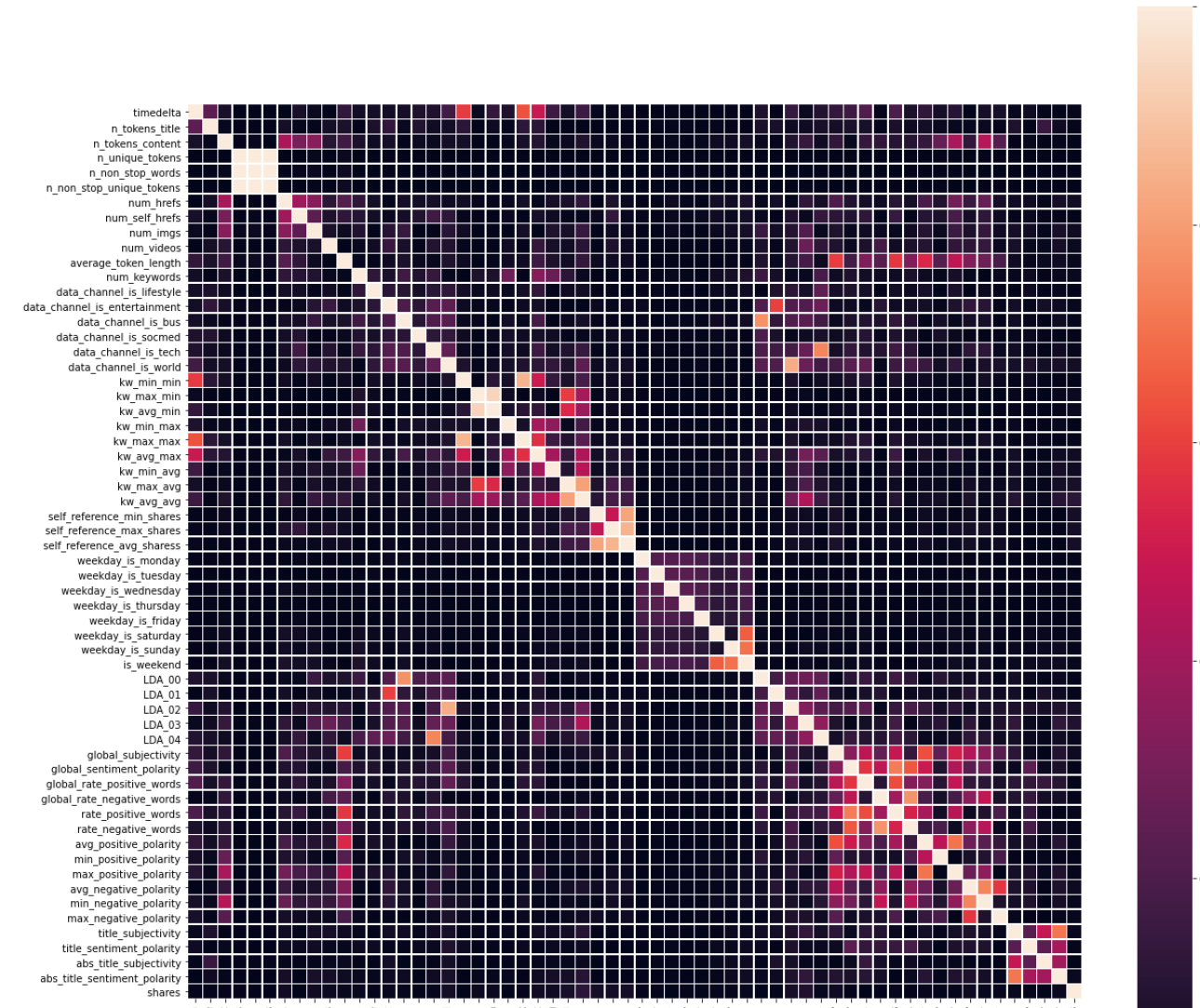
Below are a few boxplots from the dataset:

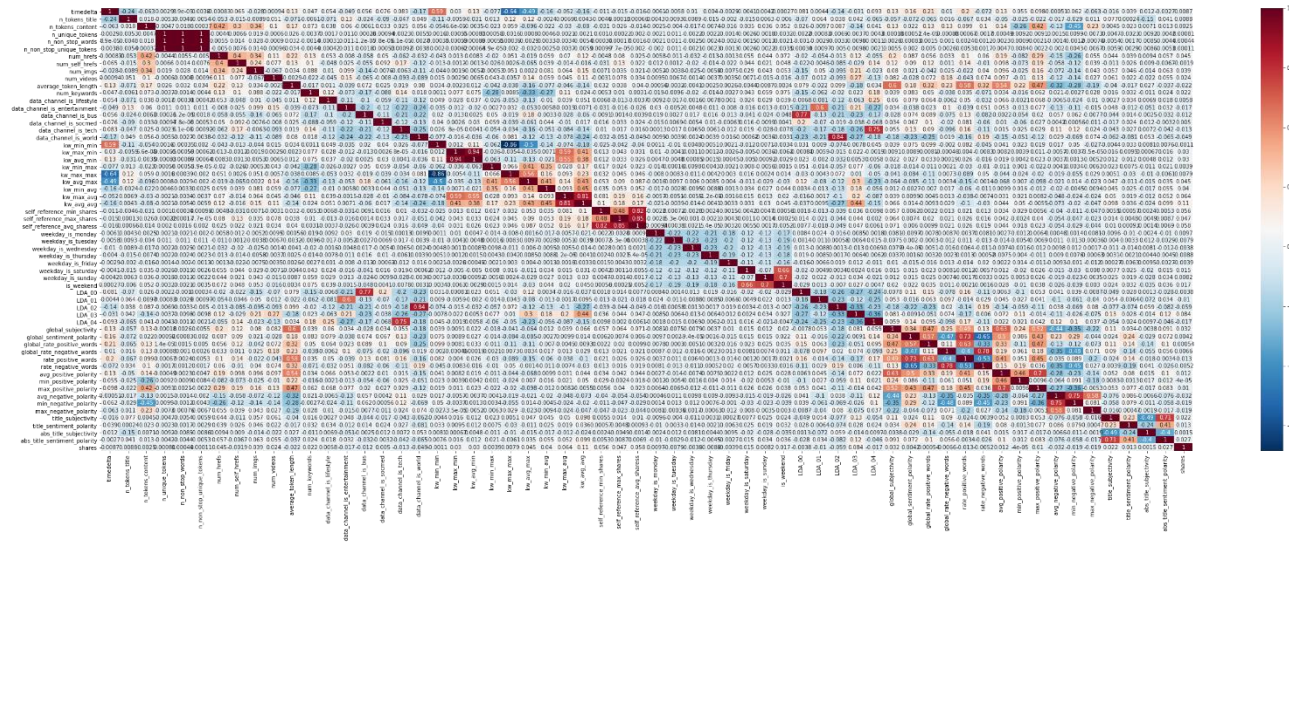




As seen above, the variables have many outliers (the black holes).

A function is defined and shows the calculated correlation between all the features pairwise. For better inference correlations are shown in a heatmap style. The colour bar on the right side. The closer to the red the more correlation exists between two variables. As you can see, the diagonal cells have the most correlation. That's because it is between a feature and itself, which is fully correlated. This plot identifies those features that are more correlated to each other and alter similarly. Therefore, you can identify features that do not provide additional information for our model and are just acting as a duplicate for the model because they have a very close correlation to another feature. To predict a variable, the features are compared with each other. The features with bigger correlations are better.


timedelta
n_tokens_title
n_tokens_content
n_unique_tokens
n_non_stop_words
n_non_stop_unique_tokens
num_hrefs
num_self_hrefs
num_imgs
num_videos
average_token_length
num_keywords
data_channel_is_lifestyle
data_channel_is_entertainment
data_channel_is_bus
data_channel_is_socmed
data_channel_is_tech
data_channel_is_world
kw_min_min
kw_max_min
kw_avg_min
kw_min_max
kw_max_max
kw_avg_max
kw_min_avg
kw_max_avg
kw_avg_avg
self_reference_min_shares
self_reference_max_shares
self_reference_avg_sharess
weekday_is_monday
weekday_is_tuesday
weekday_is_wednesday
weekday_is_thursday
weekday_is_friday
weekday_is_saturday
weekday_is_sunday
is_weekend
LDA_00
LDA_01
LDA_02
LDA_03
LDA_04
global_subjectivity
global_sentiment_polarity
global_rate_positive_words
global_rate_negative_words
rate_positive_words
rate_negative_words
avg_positive_polarity
min_positive_polarity
max_positive_polarity
avg_negative_polarity
min_negative_polarity
max_negative_polarity
title_subjectivity
title_sentiment_polarity
abs_title_subjectivity
abs_title_sentiment_polarity
shares
1.0
0.8
0.6
0.4
0.2



A scatter plot was not used because there are too many features.

Overall, the dataset is very large, but it is evident that many of the features are skewed, irrelevant and redundant, so feature selection is needed.

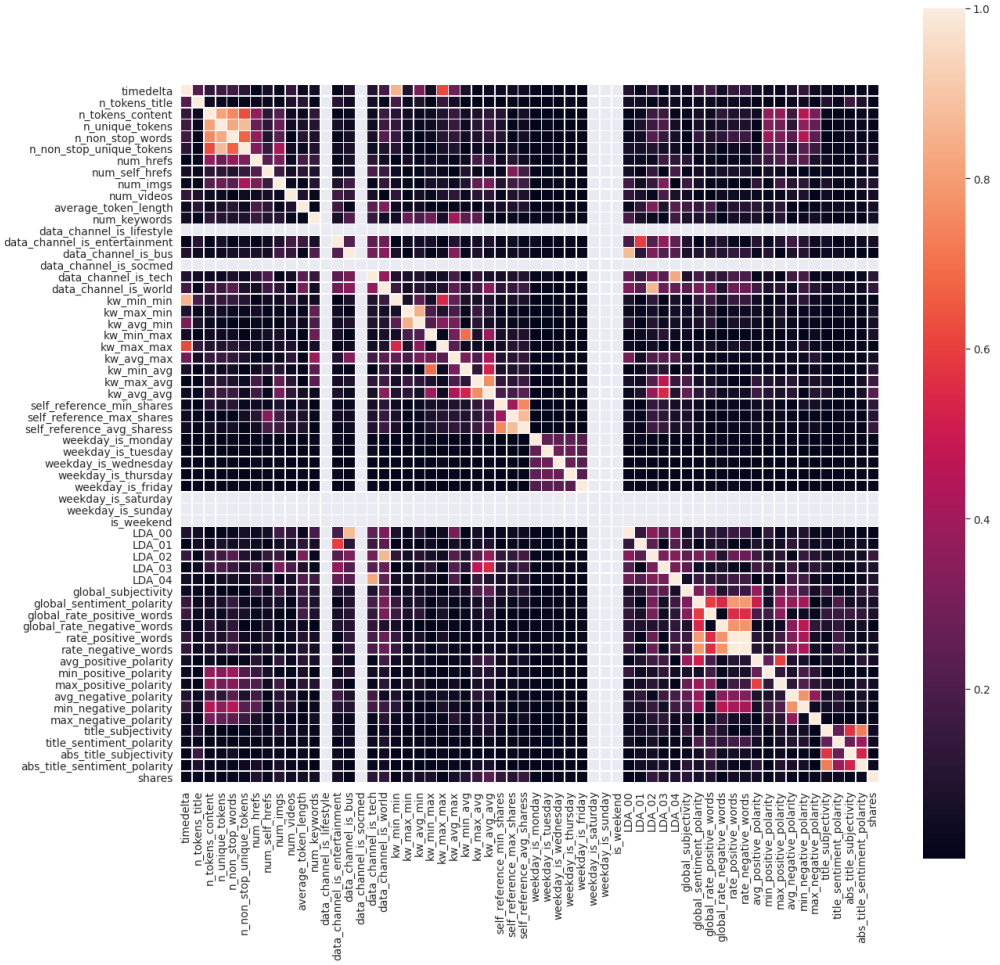
Feature Selection

Removing outliers using z-score (3 standard deviations) and interquartile range (IQR) methodologies. 16776 out of 39644 were filtered.

Normalization uses min-max scaling. After normalization, the feature's values are between -1 and 1.

Pearson is a standard correlation coefficient or collaborative filter method. Values range between 1 and +1, where +1 is very similar (total positive linear correlation) and 1 is very dissimilar (total negative linear correlation). Pearson correlation takes differences in rating behaviour into account, as mentioned in the lectures.

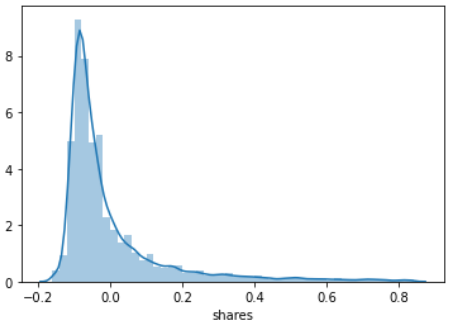
Using Pearson pairwise correlation[3], this is the list of correlation rates against shares.[4]



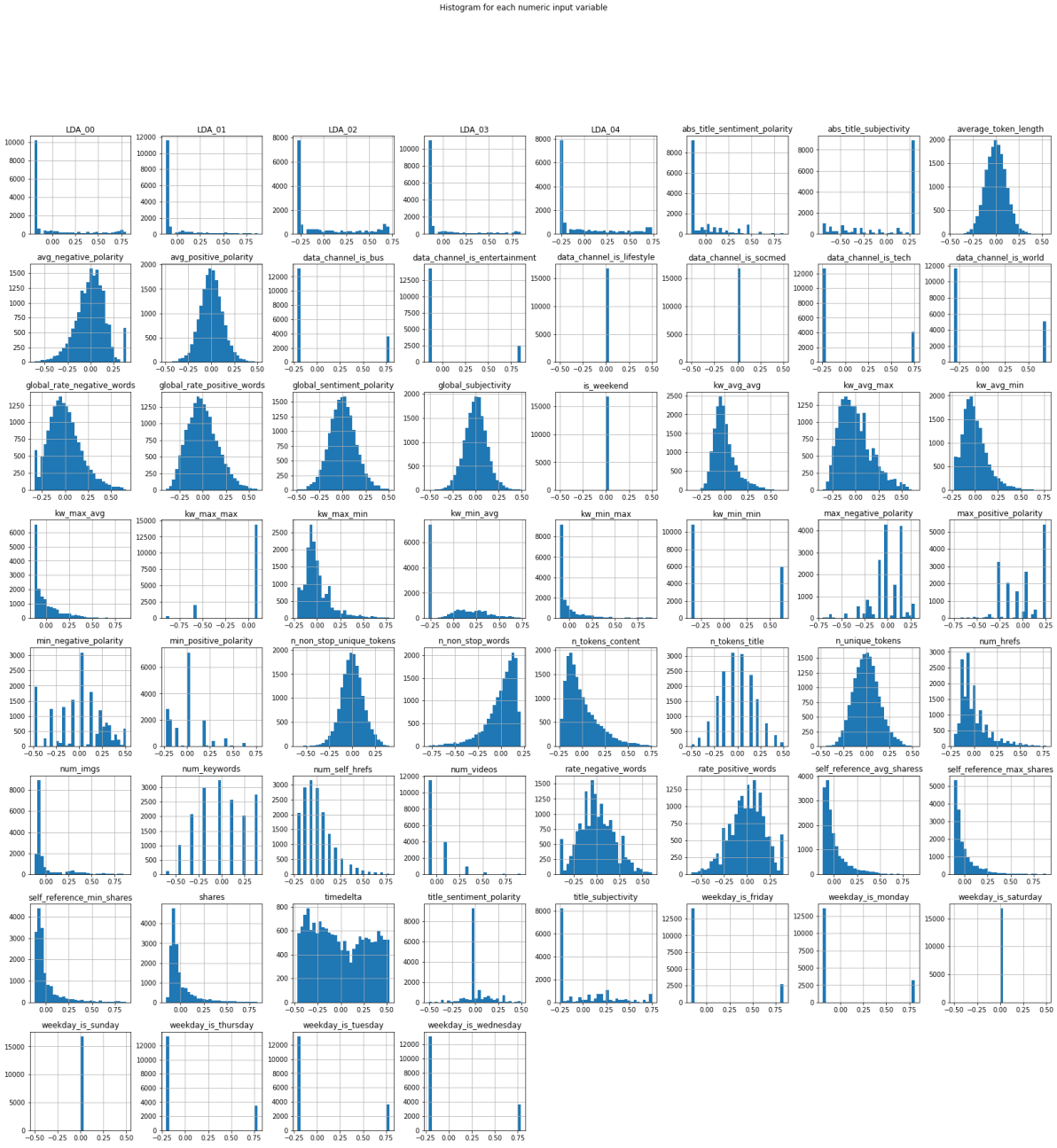
Now the features are normalized and remove the outliers, so the variables, including the target

[3] https://www.geeksforgeeks.org/python-pandas-dataframe-corr/
[4] https://towardsdatascience.com/feature-selection-with-pandas-e3690ad8504b

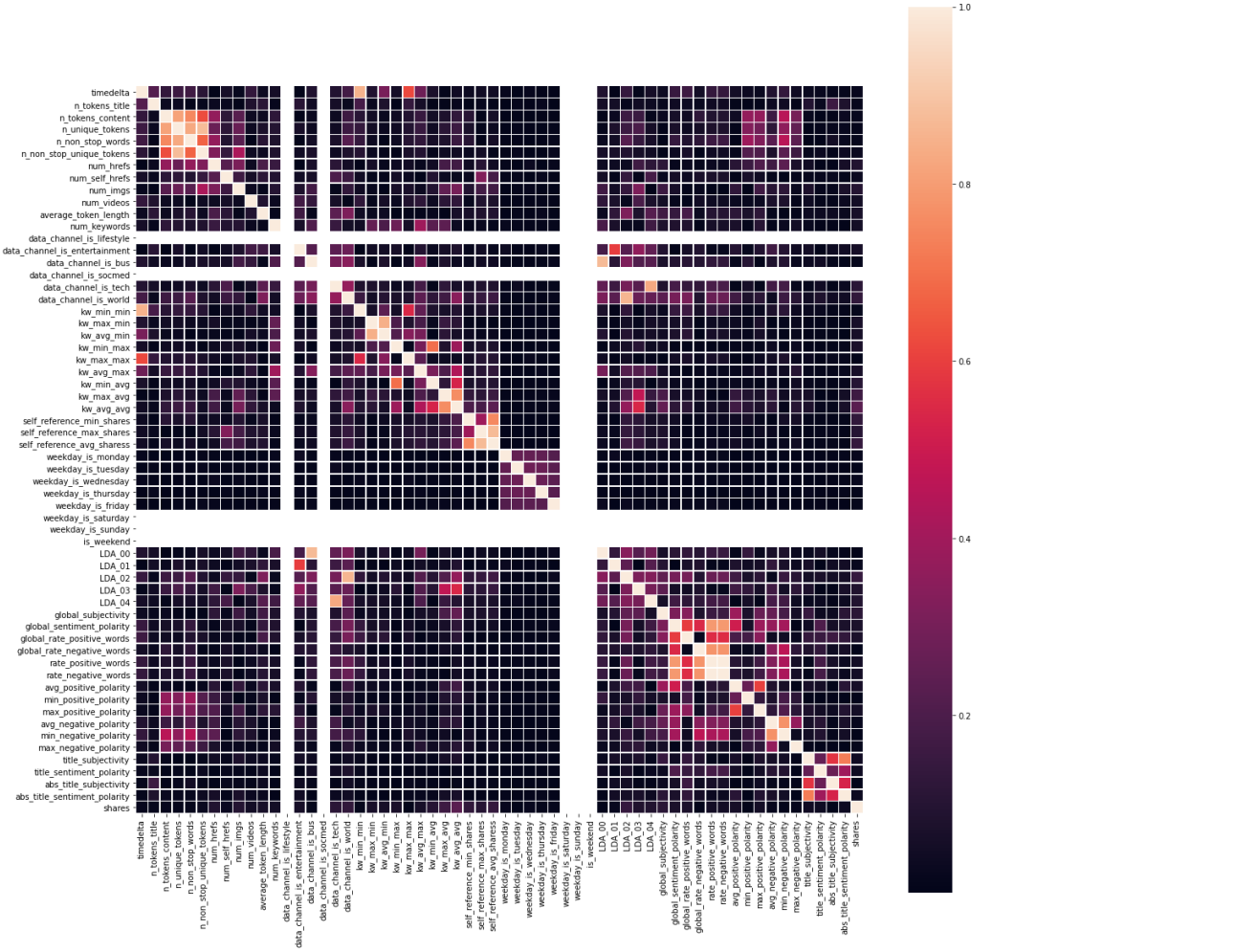
variable, is not skewed.



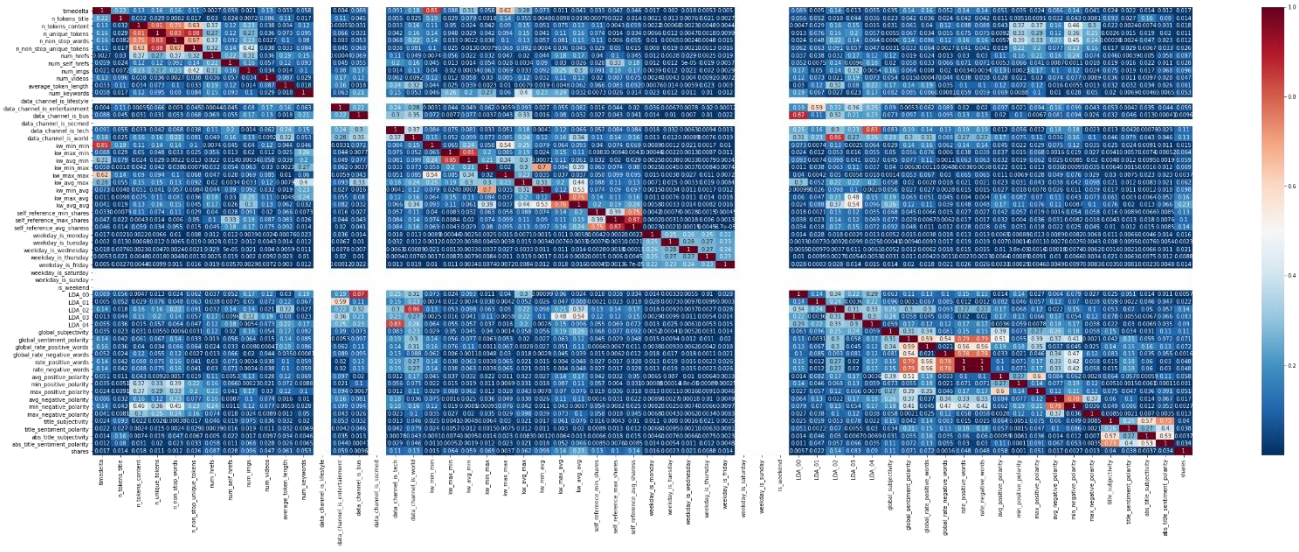
Below is the set of histograms for all the features:



Below is the heatmap:



The white colours shows that there is no correlation, so these features would be eliminated during the feature selection process. It os shown more clearly in this correlation matrix:

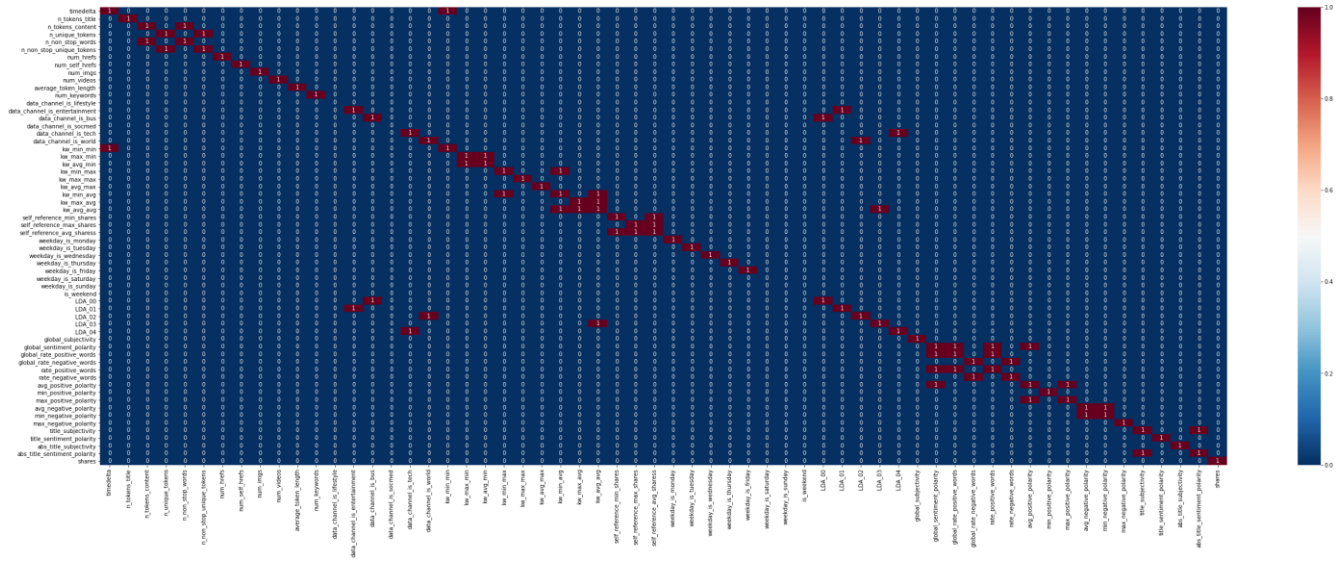


A popular technique for selecting the most relevant attributes in the dataset is to use correlation. This would calculate the correlation between each attribute and the output variable and select only those attributes that have a moderate-to-high positive or negative correlation (close to -1 or 1) and drop those attributes with a low correlation (value close to zero).[5]

Comparing the correlation with the target value, the threshold starts at 0.5. However, there are not many attributes that correlate higher than 0.5 and 0.1. The top variables (i.e. 5, 10) are filtered, so this ensures that are at least five features. With these low correlation rates, this shows that the attributes do not have a strong correlation with the target value, which is concerning. The threshold filters out features that are not relevant. However, another threshold (i.e. 0.5) to ensure that the correlation with other features are less than 0.5.

In addition, exploring the correlation with the variables (other than the target variable), the high correlations need to be removed because the attributes are redundant. The correlation between the features is compared. Then, one of the two features that correlate larger than 0.9 will be removed. Now, the dataset will only have columns with a correlation with less than 0.9. Another way to do this is by filtering the p-value, but this will not be conducted for this report. [6]

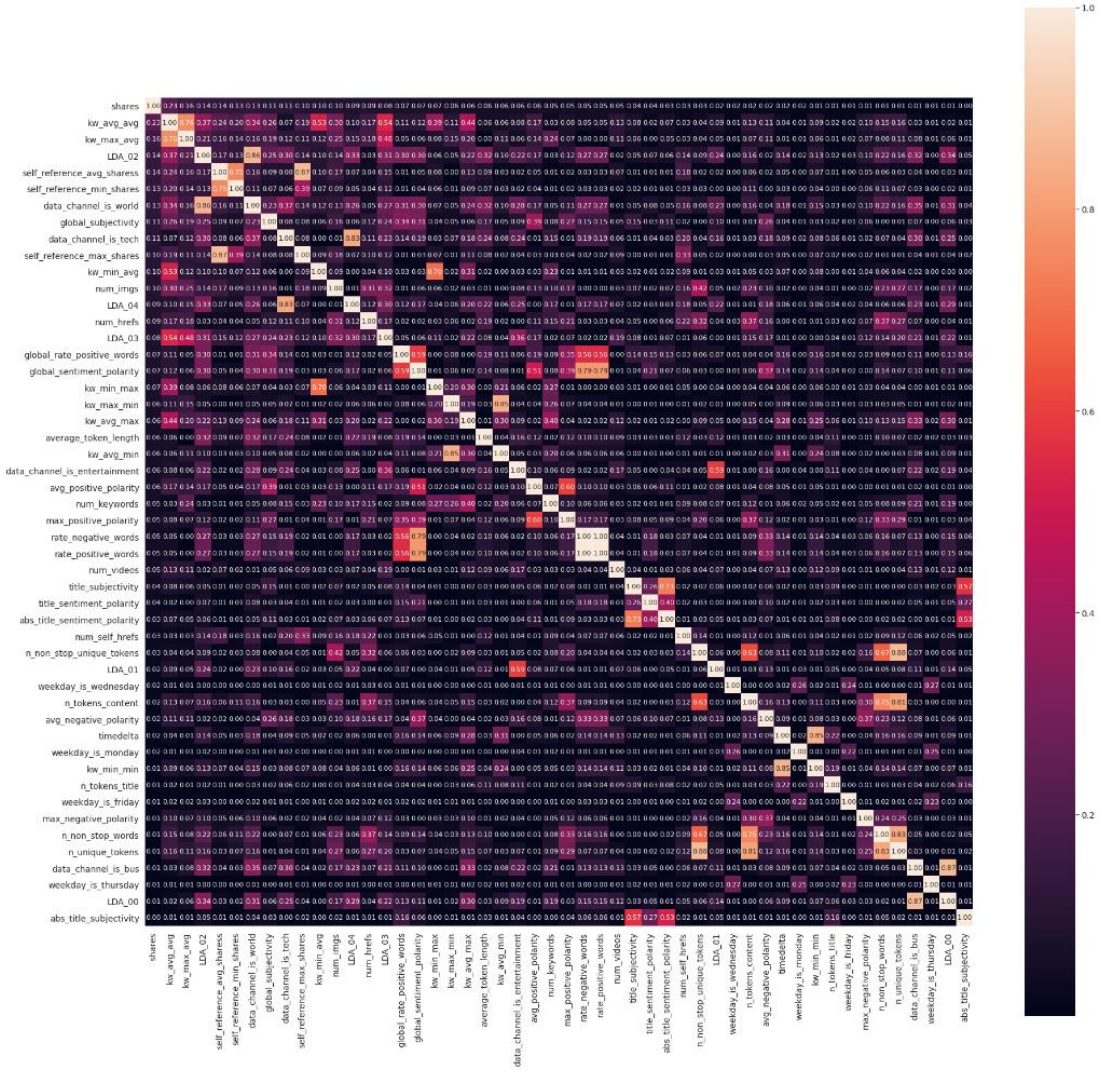
The highly correlation variables are shown more clearly below:



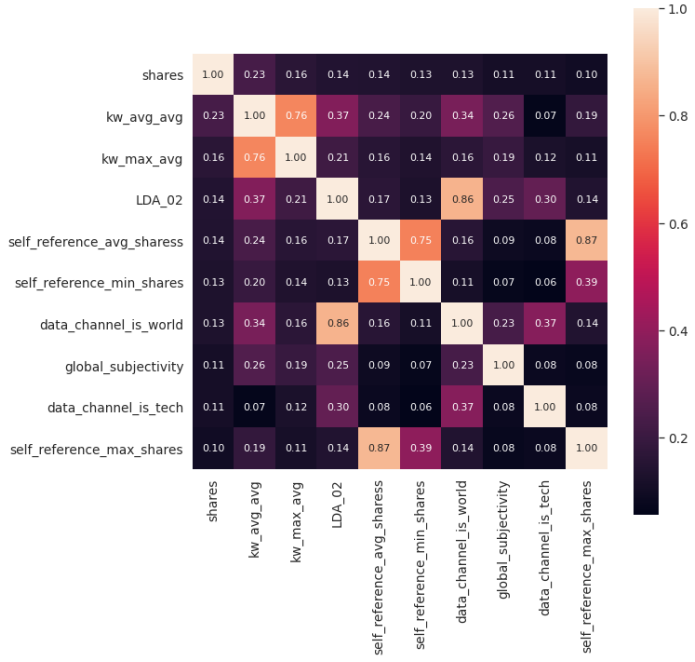
The red colours mean that the variables have a correlation greater than 0.5, while the blue colours are 0.5 and less.

[5] https://machinelearningmastery.com/perform-feature-selection-machine-learning-data-weka/
[6] https://towardsdatascience.com/feature-selection-correlation-and-p-value-da8921bfb3cf

Below are the correlation pairs with top 50 correlations with target variables:



Below are correlation pairs with top 10 correlations with the target variable:



This is the resulting dataframe after filtering the top 10 correlations:

	shares	kw_avg_avg	kw_max_avg	LDA_02	self_reference_avg_sharess	self_reference_min_shares	data_channel_is_world	global_subjectivity	data_channel_is_tech	self_reference_max_shares
4715	-0.039202	0.150198	0.143639	-0.268182	0.009187	-0.043133	-0.304006	0.083240	-0.243443	0.064509
4717	-0.004047	-0.036499	-0.113901	-0.273458	-0.009130	-0.030633	-0.304006	0.079199	0.756557	0.027563
4718	-0.047991	0.075764	-0.022156	-0.260787	-0.060120	-0.055633	-0.304006	-0.074567	-0.243443	-0.063570
4720	-0.095363	-0.021609	0.029513	0.705160	-0.121506	-0.124383	0.695994	-0.153895	-0.243443	-0.112831
4721	0.075053	-0.025733	-0.113901	-0.268171	-0.044279	-0.011883	-0.304006	-0.248714	0.756557	-0.061107
...	...	...	...	...	...	...	...	...	...	...
39635	-0.047991	0.172641	0.208807	-0.268142	0.163642	0.325617	-0.304006	-0.076391	-0.243443	0.064509
39637	-0.039202	-0.010528	-0.032300	-0.276890	0.083698	-0.070820	-0.304006	-0.003451	-0.243443	0.303425
39638	-0.056780	-0.023090	-0.031206	-0.268167	0.022388	0.000617	-0.304006	0.163434	-0.243443	0.027563
39641	0.004742	0.215585	0.195785	-0.277193	-0.066061	-0.036883	-0.304006	0.101171	-0.243443	-0.078348
39642	-0.065569	-0.160622	-0.111906	0.626294	-0.103605	-0.096133	0.695994	-0.125121	-0.243443	-0.101698

16776 rows × 10 columns

After filtering correlations above 0.5:

	shares	kw_avg_avg	LDA_02	self_reference_avg_sharess	global_subjectivity	data_channel_is_tech
4715	-0.039202	0.150198	-0.268182	0.009187	0.083240	-0.243443
4717	-0.004047	-0.036499	-0.273458	-0.009130	0.079199	0.756557
4718	-0.047991	0.075764	-0.260787	-0.060120	-0.074567	-0.243443
4720	-0.095363	-0.021609	0.705160	-0.121506	-0.153895	-0.243443
4721	0.075053	-0.025733	-0.268171	-0.044279	-0.248714	0.756557
...	...	...	...	...	...	...
39635	-0.047991	0.172641	-0.268142	0.163642	-0.076391	-0.243443
39637	-0.039202	-0.010528	-0.276890	0.083698	-0.003451	-0.243443
39638	-0.056780	-0.023090	-0.268167	0.022388	0.163434	-0.243443
39641	0.004742	0.215585	-0.277193	-0.066061	0.101171	-0.243443
39642	-0.065569	-0.160622	0.626294	-0.103605	-0.125121	-0.243443

16776 rows × 6 columns

Then, the problem is converted into a classification problem in order to utilize the accuracy metrics, as well as the classification models.

Information Gain

Another popular feature selection technique is to calculate the information gain (also known as entropy) for each attribute for the output variable. Entry values vary from 0 (no information) to 1 (maximum information). Those attributes that contribute more information will have a higher information gain value and can be selected, whereas those that do not add much information will have a lower score and can be removed.[7]

After splitting the test and training data, the information gains are identified.

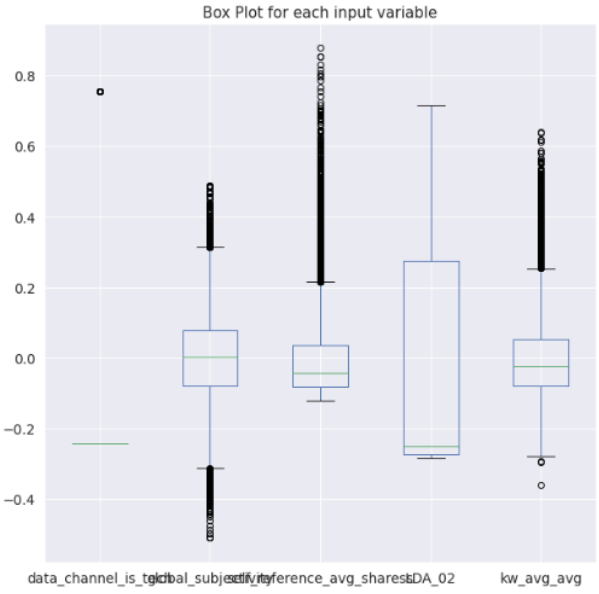
It is important to use information gain, especially for the decision tree. This is because:
1. Information gain is the main key that is used by Decision Tree Algorithms to construct a Decision Tree.
2. Decision Trees will always try to maximize Information gain.
3. An attribute with the highest information gain will be split first.[8]

[7] https://machinelearningmastery.com/perform-feature-selection-machine-learning-data-weka/
[8] https://medium.com/coinmonks/what-is-entropy-and-why-information-gain-is-matter-4e85d46d2f01

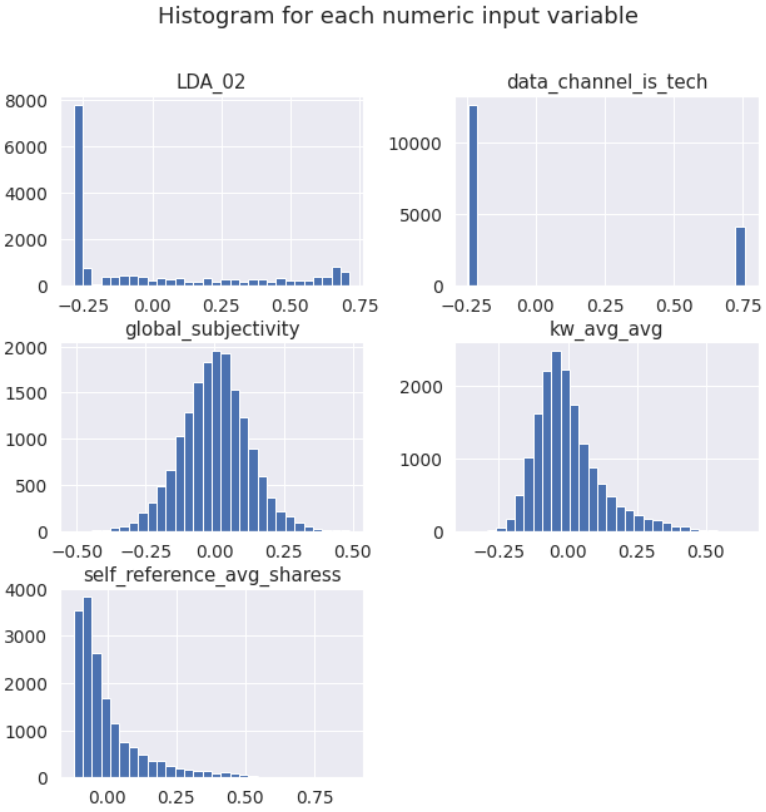
Filtering correlation and information can be redundant, but both are tested separately and together to compare performances.

The top information gains are filtered and used for the final models. This can be helpful as some of the information gains is very low.

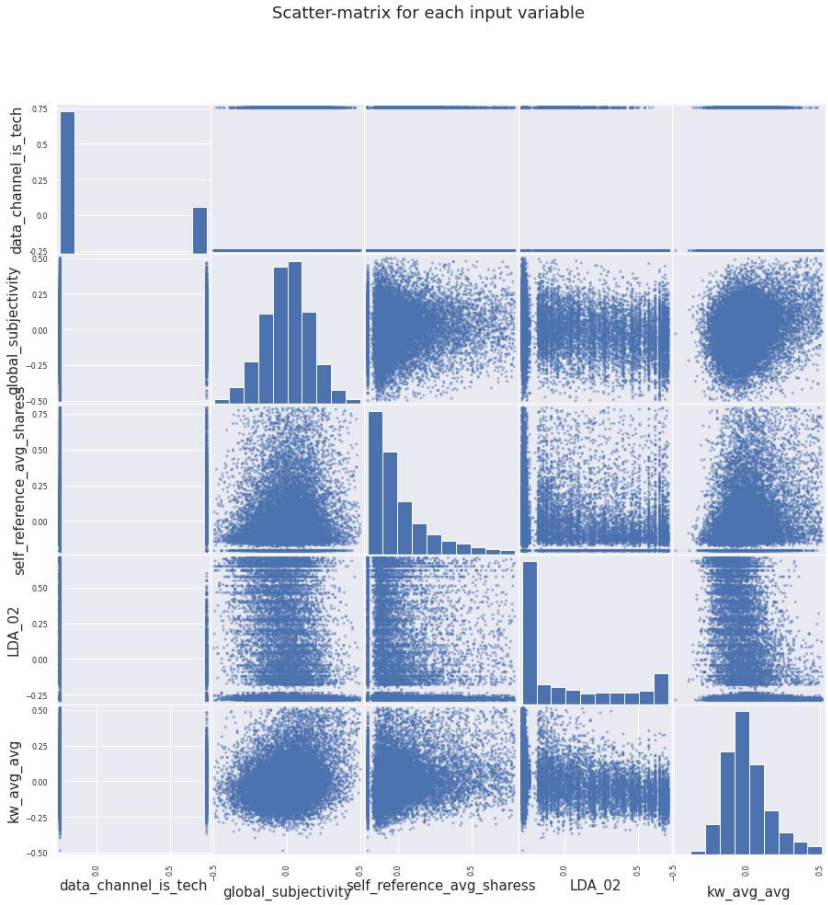
After exploring the data again, the boxplot below:



This shows that some of the outliers were removed. This would safer as removing from a larger percentile can generalize the information too much.



After the data cleaning feature selection, some of the features are less skewed.

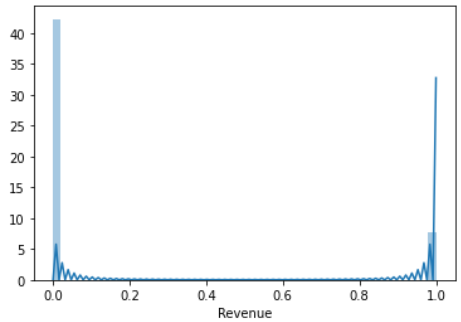


None of the features have a correlation with each other, which shows there is no redundancy.

Shopping

There are 12330 rows in the dataset. ‘False’ is very frequent in the dataset with 10442 values.

```
count     12330
unique        2
top       False
freq      10422
Name: Revenue, dtype: object
```

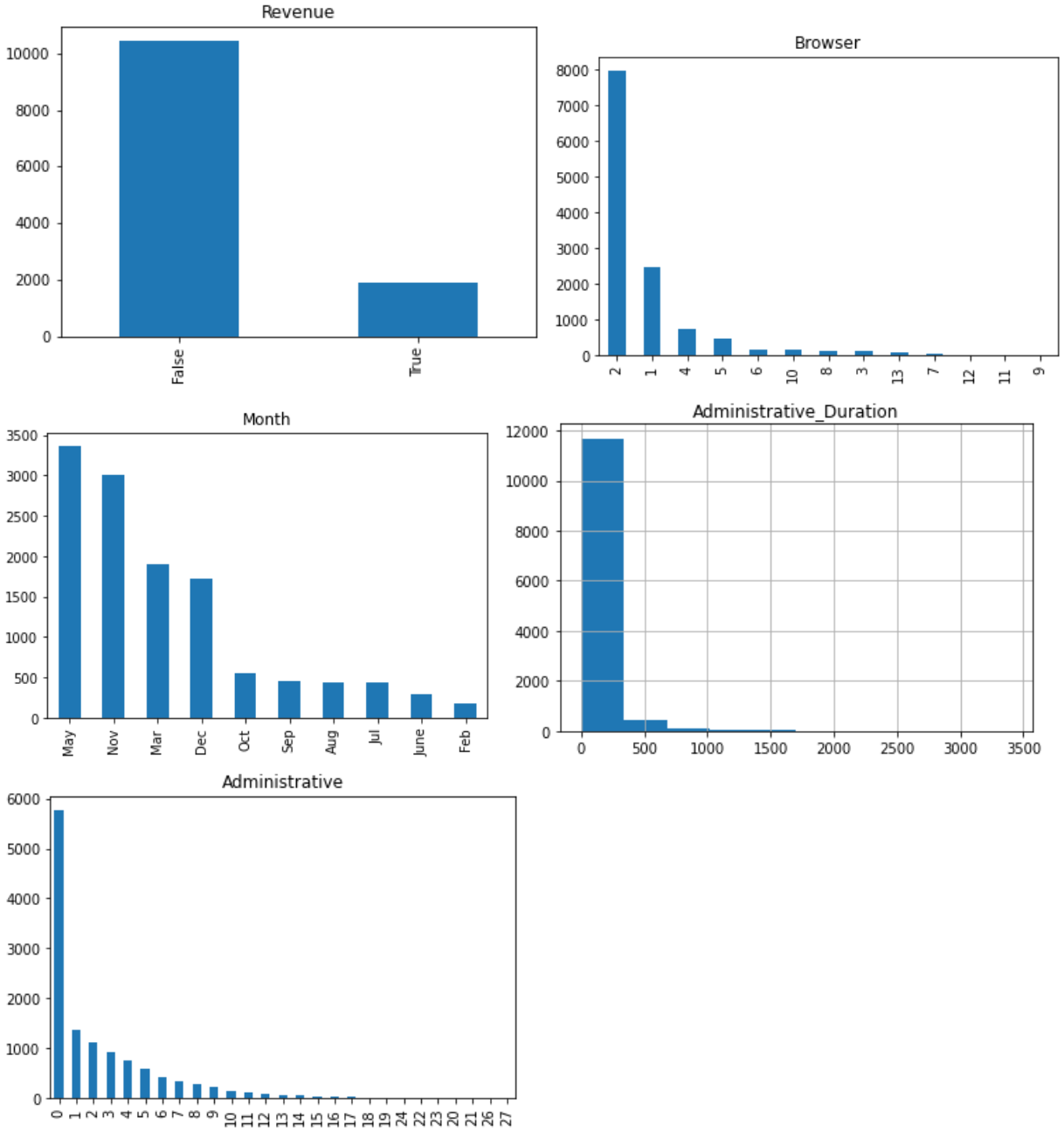


This histogram shows that the ‘False value is more frequent than ‘True, within the target variable, ‘Revenue’.

This validated using the skewness and kurtosis values:

```
Skewness: 1.909509
Kurtosis: 1.646493
```

To view both the numerical and categorical variables, a count plot was used. Below shows the frequencies of the values for each variable:

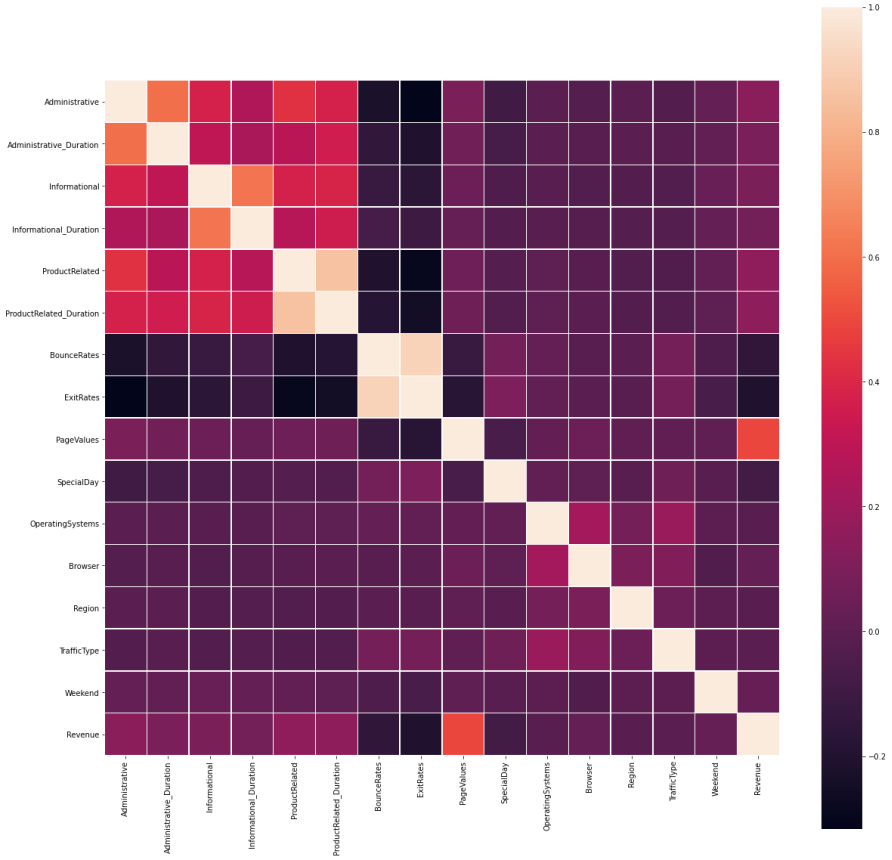


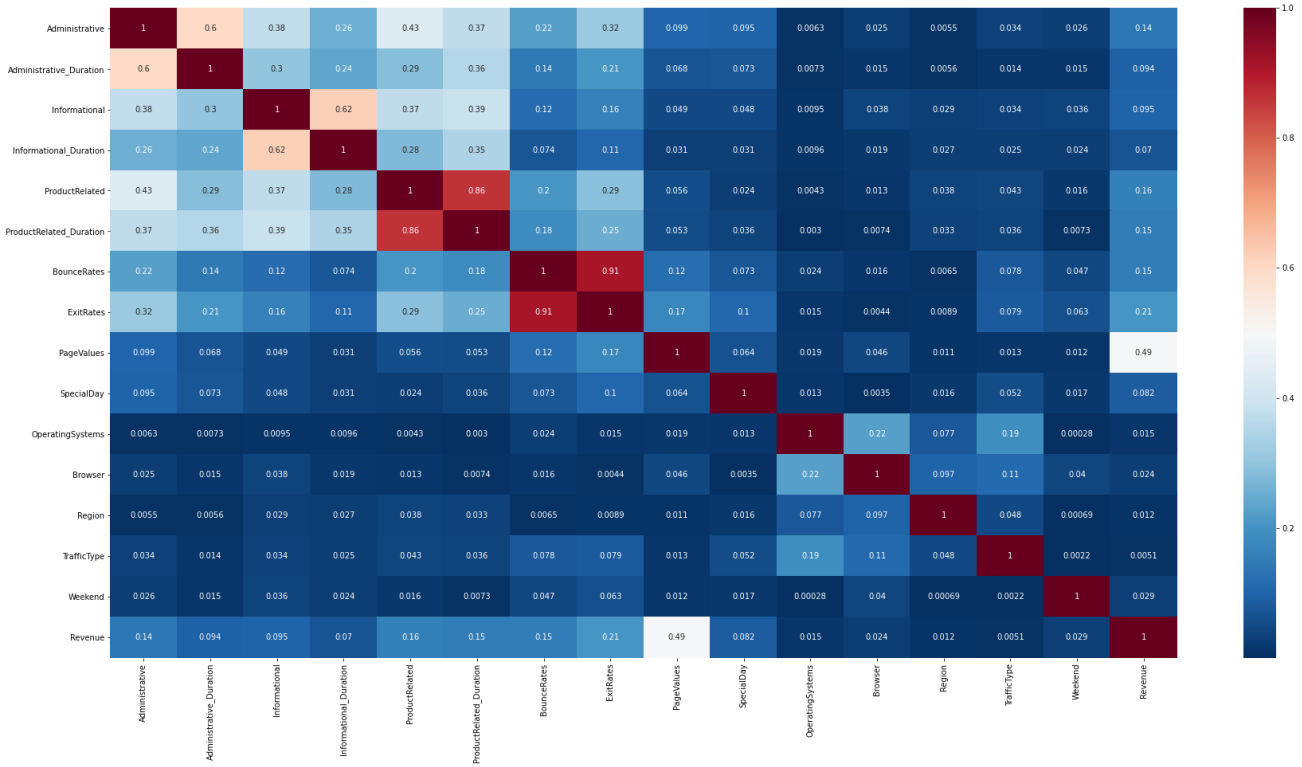
The values are skewed, so that is why removing the outliers is needed (later on).

After checking the correlation for the one-hot vectors, the one-hot vectors are close to zero with each other. So, the first column of the one-hot vector will be compared. For example, Administration's

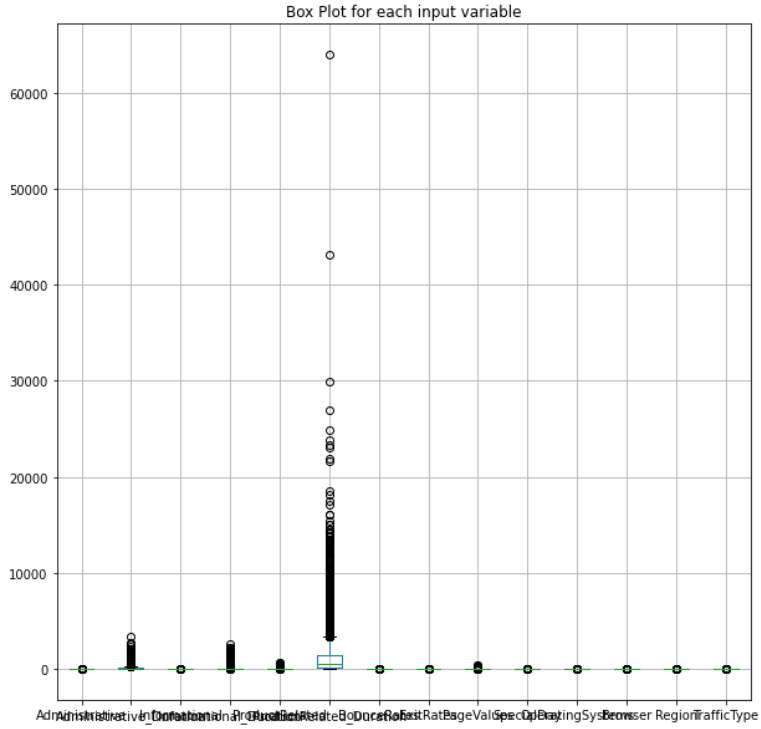
correlation will be compared with “Admin 1”. If one of the other columns have a high correlation, then the other columns would be important as well.

Below show the correlations between each pair of features using heatmap:

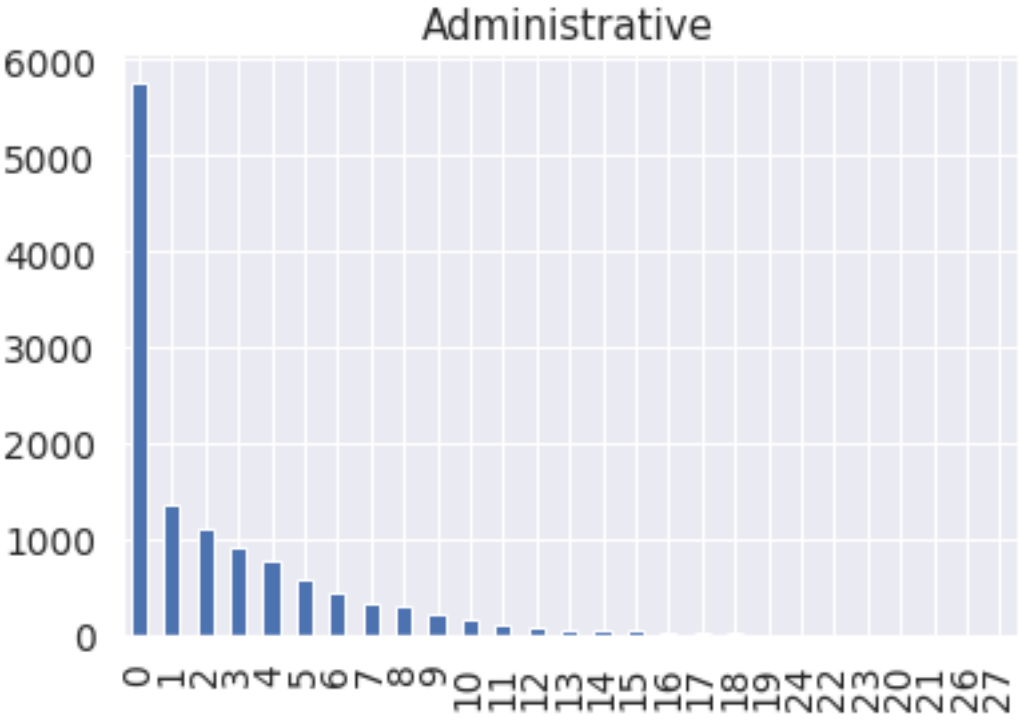


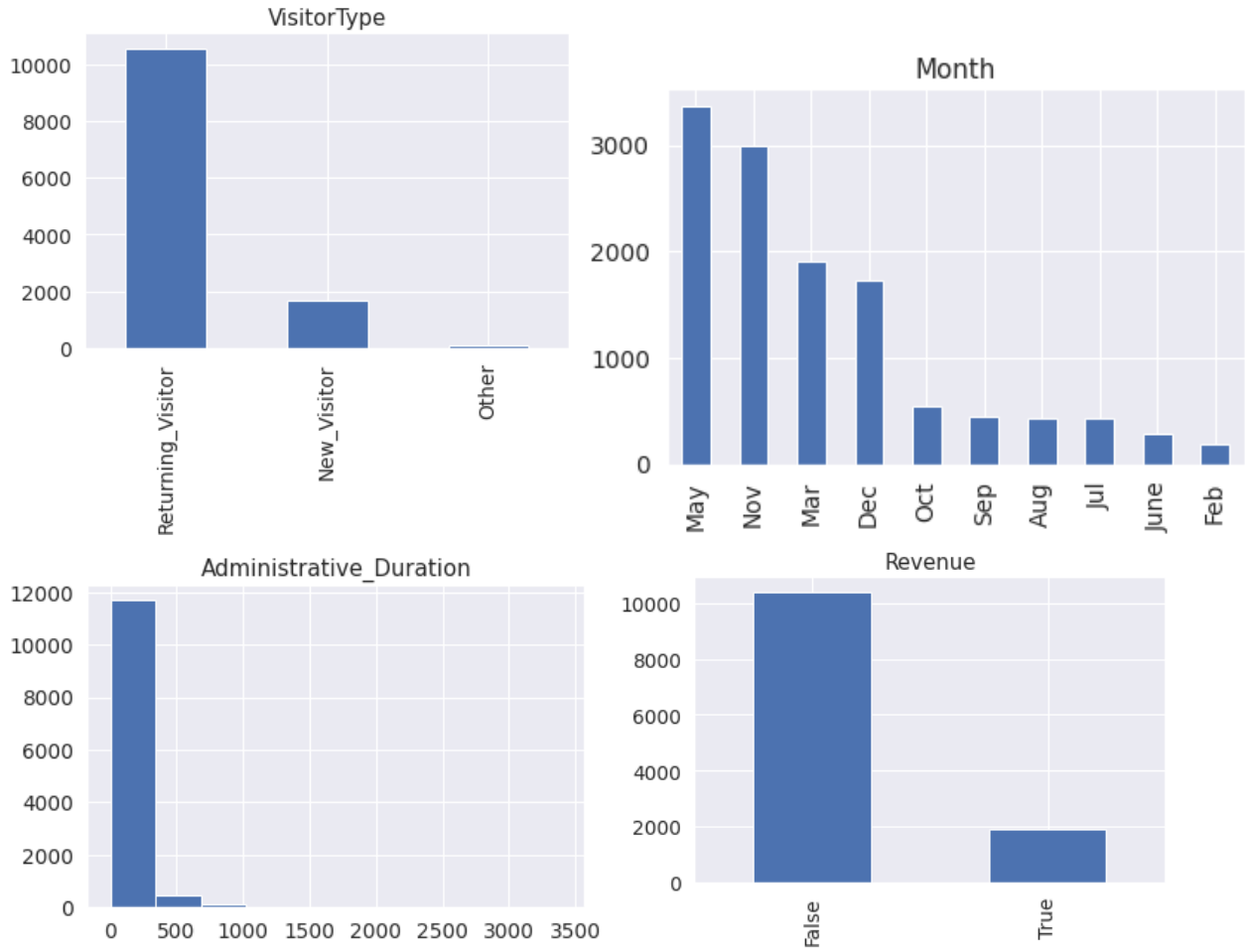


Below is the boxplot:



Below is the count plots:





There are many skewed variables, where the frequencies of one (or a few) values are greatly larger than the other values.

Scatterplot cannot be used because it can only be used on numerical values.

One Hot Vector Encoding

Many machine learning algorithms cannot operate on label data directly. They require all input variables and output variables to be numeric. In general, this is mostly a constraint of the efficient implementation of machine learning algorithms rather than hard limitations on the algorithms themselves.

This means that categorical data must be converted to a numerical form. If the categorical variable is an output variable, you may also want to convert predictions by the model back into a categorical form in order to present them or use them in some applications.

For categorical variables where no such ordinal relationship exists. It is not good to label things using 1, 2, etc. So, a one-hot encoding can be applied to the integer representation. A new binary variable is added for each unique integer value.

One example of one hot vector encoding is:

Admin 1	Admin 2	Admin 3	Admin 4	Admin 5	Admin 6	Admin 7	Admin 8	Admin 9	Admin 10	Admin 11	Admin 12	Admin 13	Admin 14	Admin 15	Admin 16	Admin 17	Admin 18	Admin 19	Admin 20	Admin 21	Admin 22	Admin 23	Admin 24	Admin 25	Admin 26	Admin 27
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

The columns were concatenated to include all of the various categories within the feature. For the non-numerical categorical features need label encoding first, so it can be converted into one-hot vector. For example, month would require this:

Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Month 7	Month 8	Month 9	Month 10
0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...
0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0

Weekend and Revenue were converted into numerical variables. True would be 1 while False would -1, so this would perform better when using the absolute values.

After completing this, 106 columns are added.

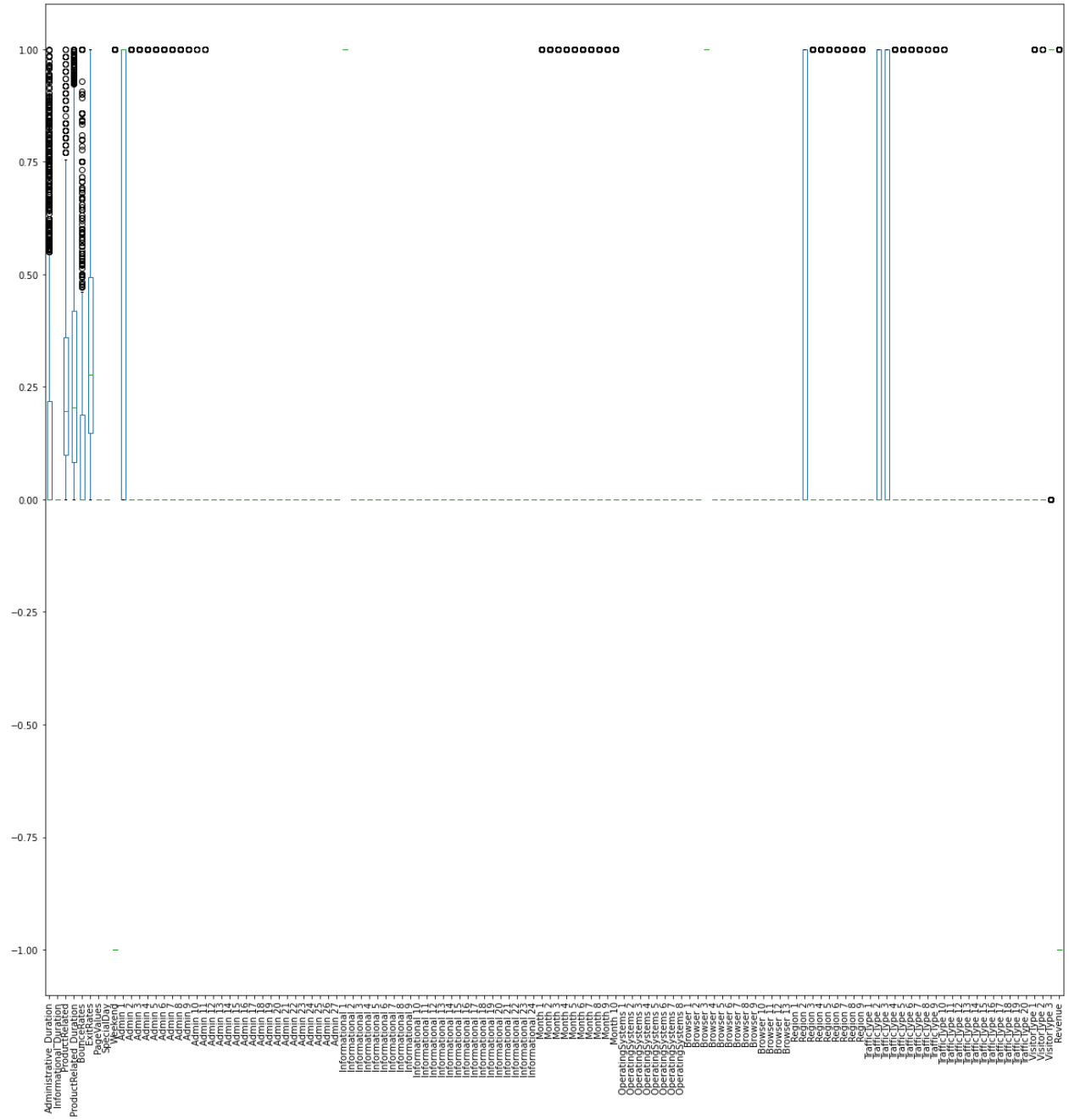
Remove Outliers

For removing the outliers, each numerical variable has the 25 and 75 percentiles removed only, with the IQR method. Adding the z-score removes all the values. 2543 rows are left after conducting this.

Normalization

Only the numerical variables are normalized, where the values range from -1 and 1. If the True/False or one hot vector are normalized, then it would not make sense, since each type not greater or less than another type.

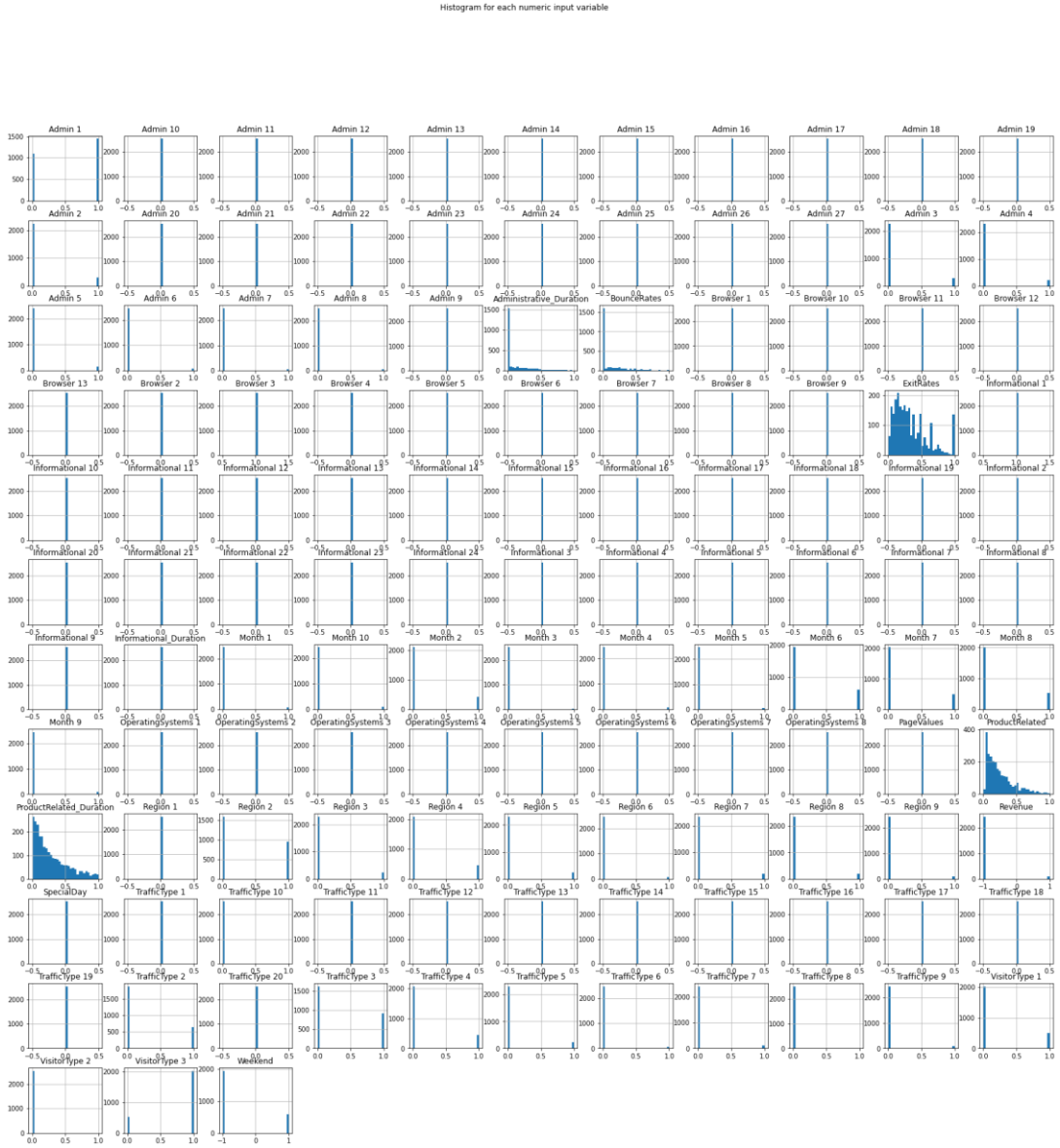
Below shows the boxplots of all the features:



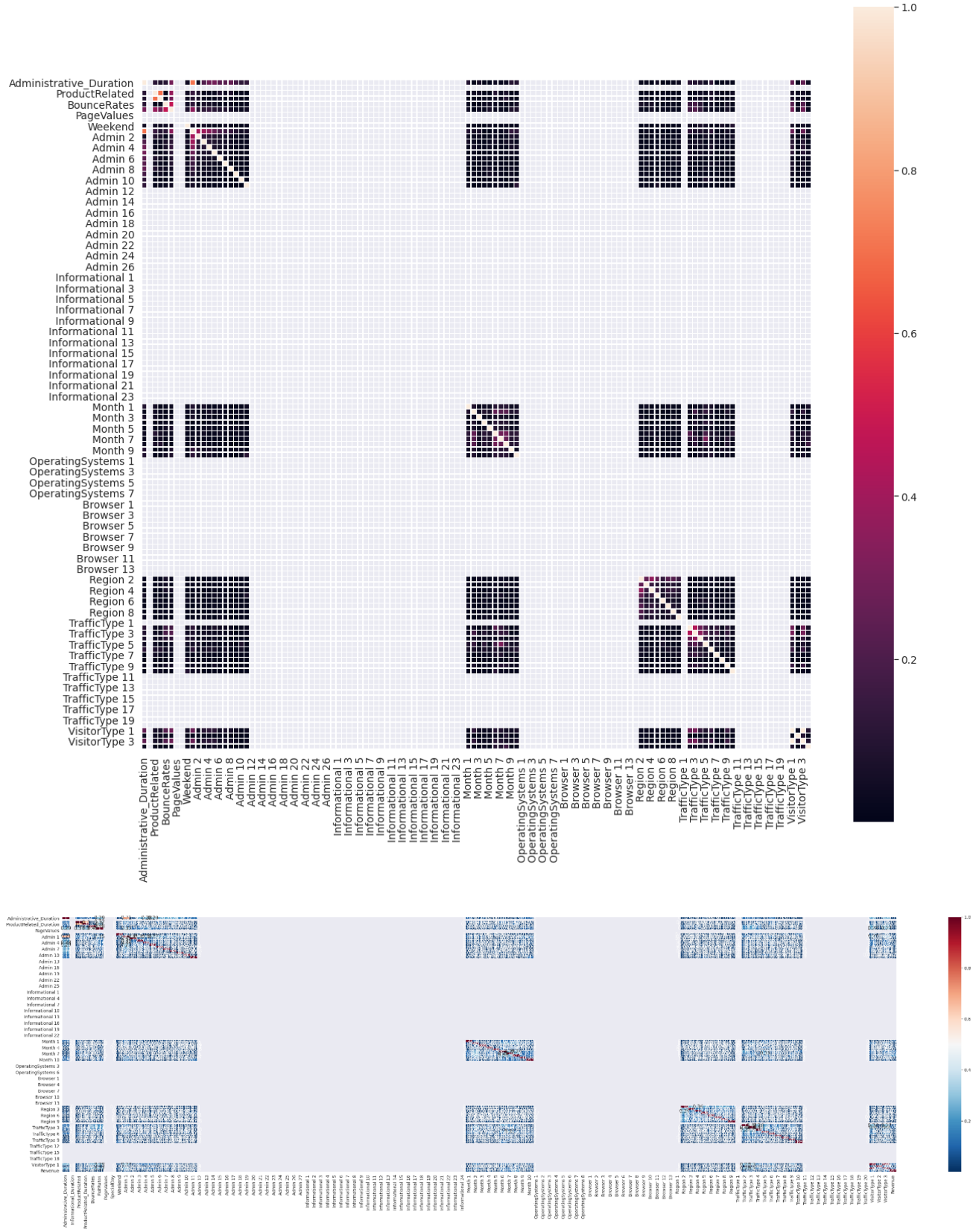
Since most of the data are apart of the one-hot vector, there are not many outliers.

This shows that there are still a lot of outliers in the dataset, so that is why feature selection will be used to narrow down the most unbias data.

Below shows the histograms:

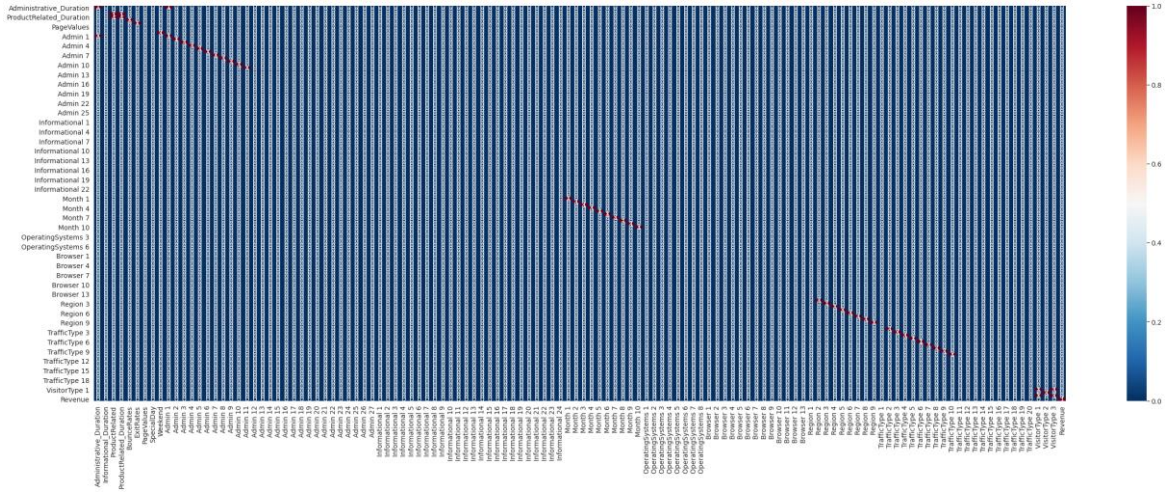


Below is the heatmap:



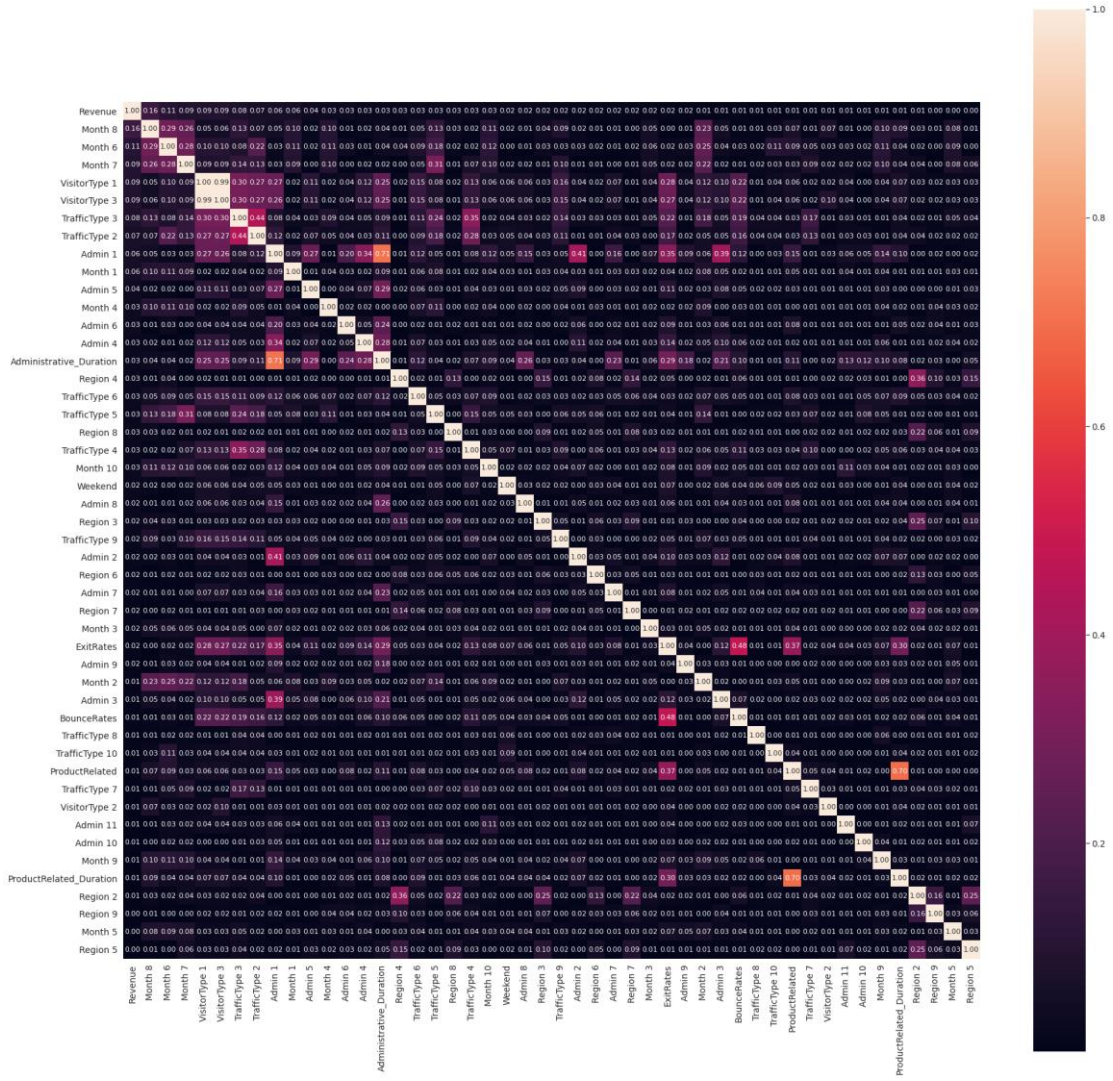
The gray and white spaces means that there is no correlation (NaN), while the dark spaces means there is low correlation. The lighter colours show there is high correlation.

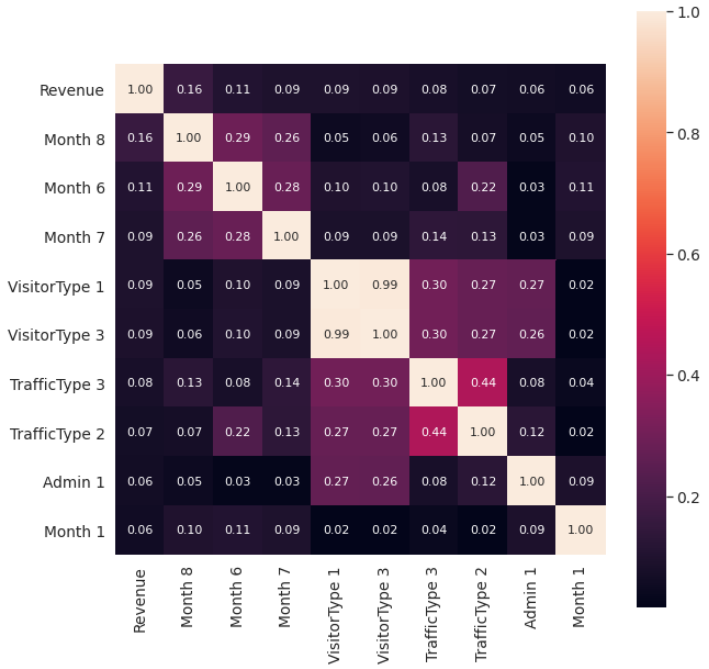
This shows the correlation larger than 0.5:



Feature Selection

Similar to the news dataset, threshold filtering is used. The following are attempted in order to improve performance. Filtering the top 50 and 10 features with high correlation with target variable.





Then, the features with high correlation with other features were removed.

	Revenue	Month 8	Month 6	Month 7	VisitorType 1	TrafficType 3	TrafficType 2	Admin 1	Month 1
1	-1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0
5	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
14	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
18	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
22	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
...	...	...	...	...	...	...	...	...	...
12310	-1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12320	-1.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0
12322	-1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
12326	-1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
12329	-1.0	1.0	0.0	0.0	1.0	1.0	0.0	1.0	0.0

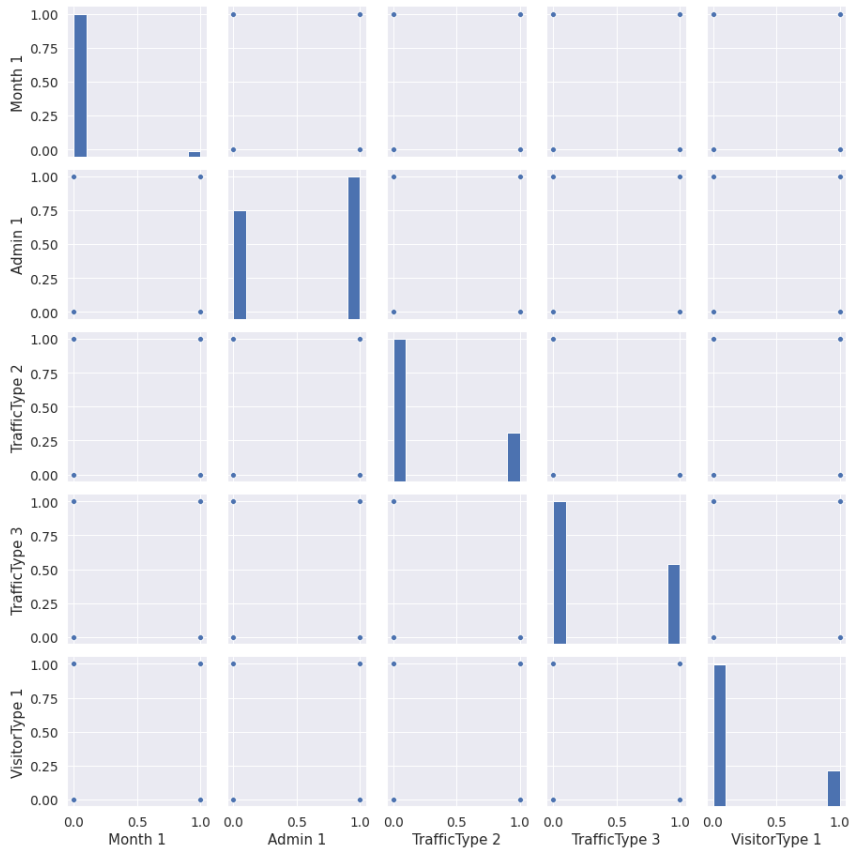
2543 rows × 9 columns

Filtering out the high information gains only:

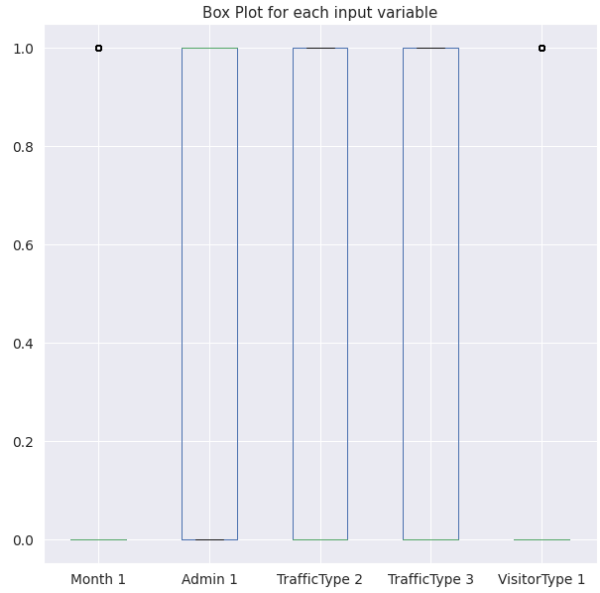
	Month 1	Admin 1	TrafficType 2	TrafficType 3	VisitorType 1
1	0.0	1.0	0.0	1.0	0.0
5	0.0	1.0	0.0	0.0	0.0
14	0.0	1.0	0.0	0.0	0.0
18	0.0	1.0	0.0	0.0	0.0
22	0.0	1.0	0.0	0.0	0.0
...	...	...	...	...	...
12310	0.0	0.0	0.0	0.0	0.0
12320	0.0	1.0	1.0	0.0	0.0
12322	0.0	0.0	0.0	1.0	0.0
12326	0.0	1.0	0.0	0.0	0.0
12329	0.0	1.0	0.0	1.0	1.0

2543 rows × 5 columns

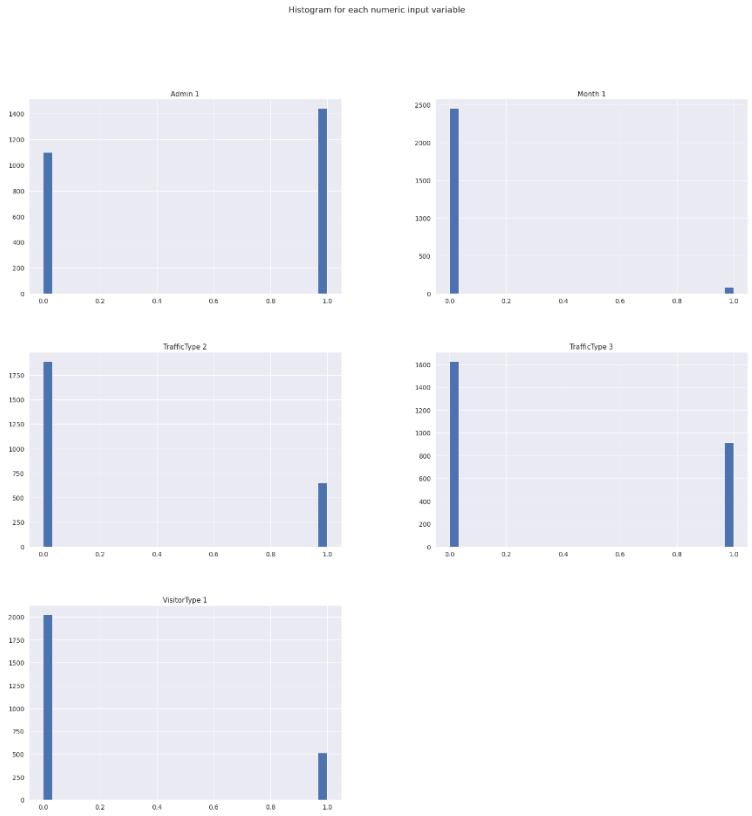
The pair plots below show that there is no correlation between the features, which means the model will not have redundancy or bias.



Most of the outliers are removed from the boxplot below, so there won't be as much bias in the model.



The histogram below shows the relevant variables, even within the one hot vector. It is assumed that when one variable (within a one hot vector) has a relationship or correlation with another feature, this would be reflected with the rest of the one hot encoded columns. For example, TrafficType shows up twice, which means that traffic is one of the most relevant features for this dataset. Or, Admin 1 shows up, which means the Administrative feature is important, even if the other "Admin" columns are not selected.



Overall, this feature selection process will help improve performance.

Learning Methods

a) Decision Tree

Decision trees are useful for both categorical and numerical features. The models produced are generally quite understandable, since each decision is represented by one node of the tree. However, this approach is only useful for low dimension feature vectors.

With many features, building decision trees with many levels leads to overfitting, where below the top levels, the decisions are based on peculiarities of small fractions of the training set, rather than fundamental properties of the data. However, if a decision tree has few levels, then it cannot even mention more than a small number of features.

Thus, this method is used as it is very common and can be used after feature selection.

b) K-Nearest neighbour (k-NN)

The model is the training set itself, so we expect it to be intuitively understandable. The approach can deal with multidimensional data, although the larger the number of dimensions, the sparser the training set will be. Therefore, the less likely it is that we find a training point very close to the point we need to classify.

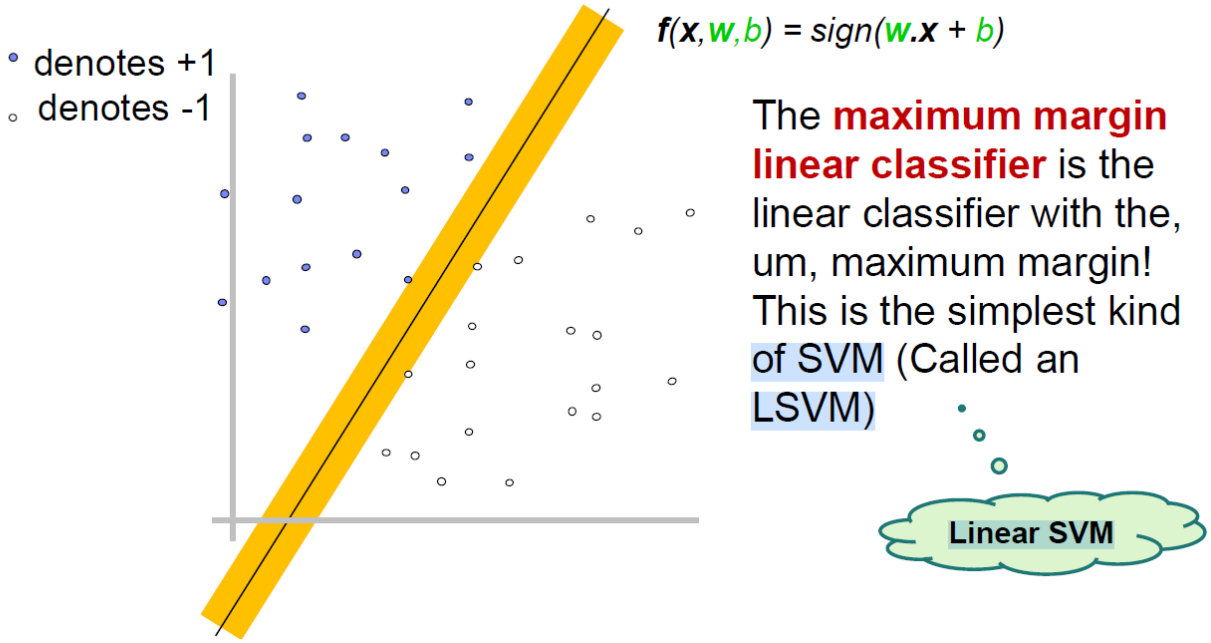
The “curse of dimensionality” makes nearest neighbour methods not very useful in high dimensions. It is mainly useful when we deal with numerical features. This method has some parameters to set:

- The distance measure we use (e.g., cosine or Euclidean)
- The number of neighbours to choose (i.e., k)
- In many cases, it is not obvious which choices yield the best results

Thus, this method is useful after feature selection and converted to numerical values (one-hot encoding).

c) SVM

The maximum margin linear classifier is the linear classifier with the maximum margin. This is the simplest kind of SVM (also known as LSVM).



Support vectors are those data points that the margin pushes up again. The maximum margin will minimize the chances of misclassification and is immune to removal of any non-support-vector data points. Empirically, the model is very good. It is theoretically good at handling large feature spaces and overfitting is controlled by margin. However, it is very sensitive to noise, decreasing performance.

Thus, SVM is used for classification problems and performs well, especially after feature selection.

When using data mining algorithms, there are limitations for each algorithm. Some algorithms are not able to handle symbolic attributes or may perform poorly when data contains symbolic attributes. Some other algorithms may have this problem with continuous attributes. Also, some algorithms may not be able to handle missing values. In this case, missing values should be predicted in a preprocessing step before passing the data to the algorithm. Another option is to remove data points that contain missing values.

Evaluation

All the performance measures you used should be described first and then draw figures that represent results and discuss the results and compare the methods.

News

Mean absolute error, mean squared error, and root mean squared error are used as a metric since the dataset is regression.

Normalized

Method	Accuracy (Train) (%)	Accuracy (Test) (%)	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
Decision Tree	100	-461/341	0.0038	0.0015	0.0121
k-NN	37	-43	0.0038	0.0015	0.0121
SVM	-23.83	-51.92	0.0724	0.0054	0.0734

Remove Outliers

Method	Accuracy (Train) (%)	Accuracy (Test) (%)	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
Decision Tree	100	-87	0.1302	0.0462	0.2149
k-NN	42	-16	0.1058	0.0287	0.1693
SVM	28	7	0.1033	0.0229	0.1512

Top 10 Correlation

Method	Accuracy (Train) (%)	Accuracy (Test) (%)	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
Decision Tree	100	-87	0.1311	0.0461	0.2148
k-NN	40	-21	0.1105	0.0298	0.1726
SVM	6	5	0.1016	0.0234	0.1529

Top 50 Correlation

Method	Accuracy (Train) (%)	Accuracy (Test) (%)	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
Decision Tree	100	-94	0.1333	0.0479	0.2188
k-NN	43	-17	0.1076	0.0289	0.1701
SVM	13	6	0.1016	0.0231	0.1520

Top 20 Information Gain

Method	Accuracy (Train) (%)	Accuracy (Test) (%)	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
Decision Tree	100	-96	0.1343	0.0483	0.2130
k-NN	43	-17	0.1077	0.0289	0.1701
SVM	13	6	0.1016	0.0231	0.1520

Top 5 Information Gain

Method	Accuracy (Train) (%)	Accuracy (Test) (%)	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
Decision Tree	100	-87	0.1313	0.0460	0.2145
k-NN	40	-21	0.1105	0.0298	0.1726
SVM	6	5	0.1016	0.0234	0.1529

Normalization with Classification

**** This is based on the weighted average***

Method	Accuracy (Train) (%)	Accuracy (Test) (%)	Precision (%)*	Recall (%)*	F1 Score (%)*
Decision Tree	100	2	55	55	55
k-NN	23	2	58	58	58
SVM	N/A	N/A	N/A	N/A	N/A

SVM's runtime is extremely slow. This is most likely due to the outliers and noise of the dataset. This cannot be executed.

Top 10 Correlation with Classification

Method	Accuracy (Train) (%)	Accuracy (Test) (%)	Precision (%)*	Recall (%)*	F1 Score (%)*
Decision Tree	100	56	56	56	56
k-NN	73	58	58	58	58
SVM	63	62	62	62	62

The test accuracy increased by a lot after feature selection. The other performance metrics are fairly similar to the last one. SVM runs much faster.

Top 5 Information Gain with Classification

Method	Accuracy (Train) (%)	Accuracy (Test) (%)	Precision (%)*	Recall (%)*	F1 Score (%)*
Decision Tree	100	55	56	55	55
k-NN	73	58	58	58	58
SVM	63	62	62	62	62

It seems that SVM has the best accuracy. However, there is overfitting despite the filtering.

Overall, SVM and top 10 correlation perform the best.

Shopping

**** This is based on the weighted average***

Normalized

Method	Accuracy (Train) (%)	Accuracy (Test) (%)	Precision (%)*	Recall (%)*	F1 Score (%)*
Decision Tree	100	86	86	86	86
k-NN	88	83	80	83	80
SVM	86	84	85	84	78

The normalized methods are relatively accurate and perform moderately overall. However, there is overfitting, as the test accuracy is less than the training accuracy.

Removed Outliers

Method	Accuracy (Train) (%)	Accuracy (Test) (%)	Precision (%)*	Recall (%)*	F1 Score (%)*
Decision Tree	100	94	95	94	95
k-NN	97	97	94	97	95
SVM	96	97	94	97	95

After removing the outliers, the performance increased by around 10%. Precision and F1 score are similar amongst the different methods. However, the recall is slightly better in k-NN and SVM. There is still overfitting with the decision tree and SVM. k-NN has the same accuracy in training and test.

Top 50 Correlation

Method	Accuracy (Train) (%)	Accuracy (Test) (%)	Precision (%)*	Recall (%)*	F1 Score (%)*
Decision Tree	100	94	95	94	95
k-NN	97	97	94	97	95

SVM	96	97	94	97	95

This does not increase accuracy, which means some of the features are creating some noise and bias.

Top 5 Correlation

Method	Accuracy (Train) (%)	Accuracy (Test) (%)	Precision (%)*	Recall (%)*	F1 Score (%)*
Decision Tree	96	97	94	97	95
k-NN	96	97	94	97	95
SVM	96	97	94	97	95

The performance metrics are all the same. The performances improved slightly.

Top 20 Information Gain

Method	Accuracy (Train) (%)	Accuracy (Test) (%)	Precision (%)*	Recall (%)*	F1 Score (%)*
Decision Tree	100	94	95	94	94
k-NN	97	97	94	97	95
SVM	96	97	94	97	95

The recall is slightly lower. There is overfitting with the decision tree, but the k-NN and SVM have good accuracies.

Top 5 Information Gain

Method	Accuracy (Train) (%)	Accuracy (Test) (%)	Precision (%)*	Recall (%)*	F1 Score (%)*
Decision Tree	96	97	94	97	95
k-NN	96	97	94	97	95
SVM	96	97	94	97	95

This is not required, since it does not change the performance metrics, compared to the top 5 correlated features.

Overall, top 5 correlation and information gain have the best performance. SVM and k-NN performed well consistently and had less overfitting than the decision tree.

Discussion

News

a) Decision Tree

   The model has bad accuracy and problems with overfitting.

b) K-Nearest neighbour (k-NN)

   This had an even lower accuracy compared to the decision tree. This also had overfitting.

c) SVM

   The model was not able to classify the data, even when it was converted into a classification problem. This had a very long runtime.

SVM performed the best overall, with k-NN close behind. It can perform well with classification and regression, which shows the resilient and adaptability of SVM.

Regression models should be used instead because this is a regression problem. However, it would be more difficult find the accuracy of a regression problem. That is why the target variables were converted into a classifier.

Shopping

a) Decision Tree

   This has high accuracy for test data, compared to training data. There is no overfitting. Precision, recall, F1 score, did not change between the different criteria.

b) K-Nearest neighbor (k-NN)

   This has high accuracy for test data, compared to training data. The performance was consistent during various feature selection techniques.

c) SVM

   This has high accuracy for test data, compared to training data. The performance was consistent during various feature selection techniques.

This could have been improved by:

- Pruning the decision tree;
- Use other pre-processing techniques;
- Use other normalization techniques; and
- Experiment with various k values.

Final Results

The following models are the best for each dataset:

Dataset	Evaluation criteria	Before feature selection	After feature selection
News	*Number of features (columns)*	59	6*
	Size of dataset (rows)	39644	16776
	Accuracy (%)	-461	63
Shopping	*Number of features (columns)*	17	5
	Size of dataset (rows)	12330	2543
	Accuracy (%)	84**	97

After high correlated features were filtered.
****** Based on SVM model for top 5 information gain***

It seems that SVM performs well for both datasets. SVM has a gradual change in performance throughout the process, which makes it a good model. Overall, feature selection significantly improved all the models.

Conclusion

In summary, apply exploratory analysis on the two datasets, “Online News Popularity” and “Online Shoppers Purchasing Intention”, and three classification algorithms, SVM, k-Nearest Neighbour, and Decision Tree.

This report will be presenting the following:

1. Data exploration including correlation analysis
2. Data cleaning (feature selection) and transformation based on the exploratory analysis
3. Visual models from the three learning methods
4. Evaluation of each method
5. Discussion of the evaluations

As we explored the two datasets, we found:
Dataset 1: SVM is the best model to use for regression and classification.
Dataset 2: SVM is the best model.

This shows that SVM can work well with different types of problems and datasets.

We found redundant features or attributes within each dataset, so we had to clean the data. Feature selection is important because it filters out irrelevant and redundant features. It also ensured the model had better performance across all the models.

I could have improved the model by:

- Remove outliers from 10 and 90 percentile to check how that changes accuracy;
- Use cross-validation for splitting the data;
- Deriving new variable(s) from existing variables (also known as Feature Creation);
- Conduct research on domain to understand what variables would have the largest impact;
- Reduce the features of training data through factor analysis, low variance, backward/ forward feature selection, etc.;
- Remove skewness of variables by using methods such as log, square root or inverse of the values
- Use other algorithms such as random forest; and
- Prune the decision tree. [9]

[9] https://www.analyticsvidhya.com/blog/2015/12/improve-machine-learning-results/