PROJECT- NO:6

MACHINE LEARNING 2

(Business Report)

Problem-1	***ML PROJECT***
Problem- 2	***TEXT INSIGHT GENERATION***

Submitted by- Yathish S

(PGP-DSBA)

17th Mar 2024

PROBLEM – 1:

In FY 2016, the OFLC processed 775,979 employer applications for 1,699,957 positions for temporary and permanent labor certifications. This was a nine percent increase in the overall number of processed applications from the previous year. The process of reviewing every case is becoming a tedious task as the number of applicants is increasing every year.

The increasing number of applicants every year calls for a Machine Learning based solution that can help in shortlisting the candidates having higher chances of VISA approval. OFLC has hired the firm EasyVisa for data-driven solutions. You as a data scientist at EasyVisa have to analyze the data provided and, with the help of a classification model:

Facilitate the process of visa approvals.
Recommend a suitable profile for the applicants for whom the visa should be certified or denied based on the drivers that significantly influence the case status.

1.1) Problem 1 - Define the problem and perform Exploratory Data Analysis

"- Problem definition - Check shape, Data types, statistical summary - Univariate analysis - Bivariate analysis - Use appropriate visualizations to identify the patterns and insights - Key meaningful observations on individual variables and the relationship between variablesa ".

Read dataset-

The first 5 records are-

	case_id	continent	education_of_employee	has_job_experience	requires_job_training	no_of_employees	yr_of_estab	region_of_employment	prevailing_wage	unit_of_wage	full_time_position	case_status
0	EZYV01	Asia	High School	N	N	14513	2007	West	592.2029	Hour	Y	Denied
1	EZYV02	Asia	Master's	Y	N	2412	2002	Northeast	83425.6500	Year	Y	Certified
2	EZYV03	Asia	Bachelor's	N	Y	44444	2008	West	122996.8600	Year	Y	Denied
3	EZYV04	Asia	Bachelor's	N	N	98	1897	West	83434.0300	Year	Y	Denied
4	EZYV05	Africa	Master's	Y	N	1082	2005	South	149907.3900	Year	Y	Certified

Observations-

The dataset for top 5 records indicates only three columns are numerical rest all are categorical.

The last 5 records are-

	case_id	continent	education_of_employee	has_job_experience	requires_job_training	no_of_employees	yr_of_estab	region_of_employment	prevailing_wage	unit_of_wage	full_time_position	case_status
25475	EZYV25476	Asia	Bachelor's	Y	Y	2601	2008	South	77092.57	Year	Y	Certified
25476	EZYV25477	Asia	High School	Y	N	3274	2006	Northeast	279174.79	Year	Y	Certified
25477	EZYV25478	Asia	Master's	Y	N	1121	1910	South	146298.85	Year	N	Certified
25478	EZYV25479	Asia	Master's	Y	Y	1918	1887	West	86154.77	Year	Y	Certified
25479	EZYV25480	Asia	Bachelor's	Y	N	3195	1960	Midwest	70876.91	Year	Y	Certified

Dataset shape:

```
(25480, 12)
```

Observations-

- The Data Frame has 25480 rows and 12 columns

Dataset info:

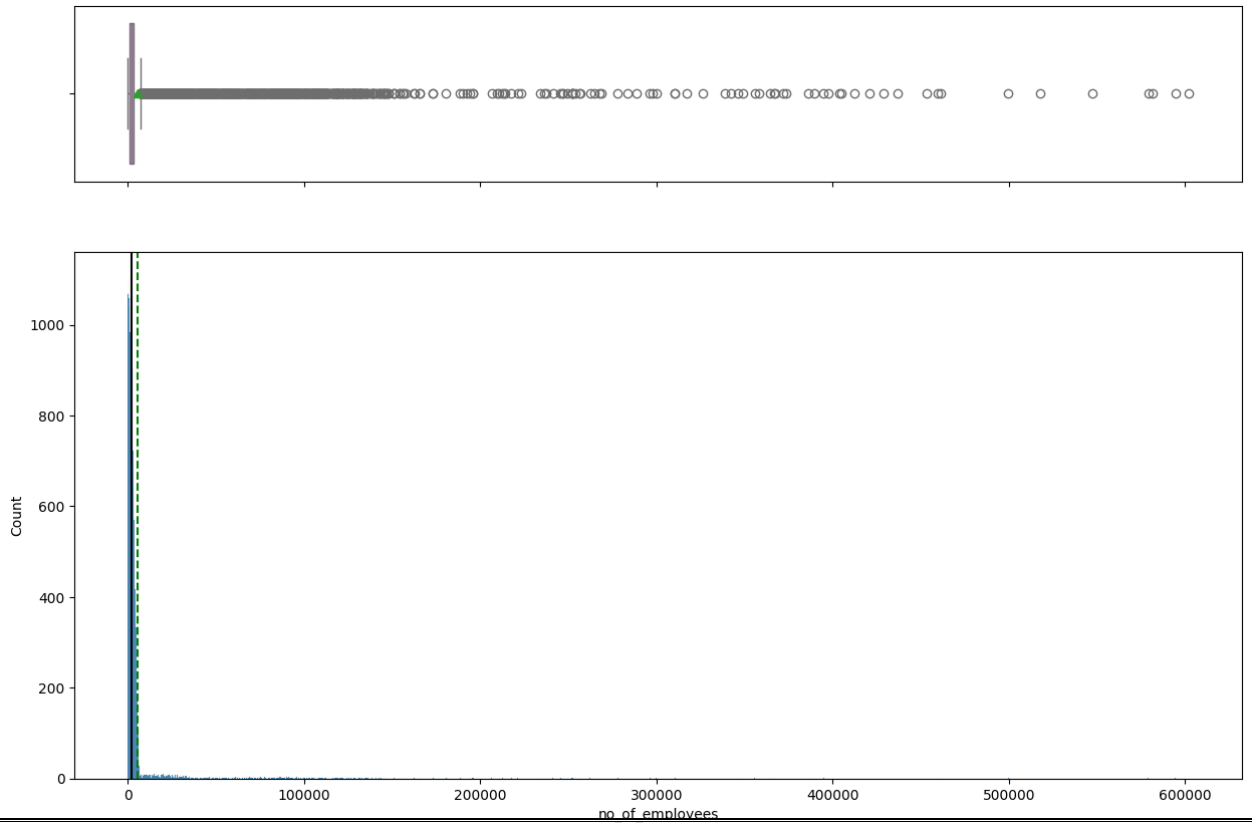
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25480 entries, 0 to 25479
Data columns (total 12 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   case_id                25480 non-null  object
 1   continent              25480 non-null  object
 2   education_of_employee  25480 non-null  object
 3   has_job_experience     25480 non-null  object
 4   requires_job_training  25480 non-null  object
 5   no_of_employees        25480 non-null  int64
 6   yr_of_estab            25480 non-null  int64
 7   region_of_employment   25480 non-null  object
 8   prevailing_wage        25480 non-null  float64
 9   unit_of_wage           25480 non-null  object
 10  full_time_position     25480 non-null  object
 11  case_status            25480 non-null  object
dtypes: float64(1), int64(2), object(9)
memory usage: 2.3+ MB
```

Observations-

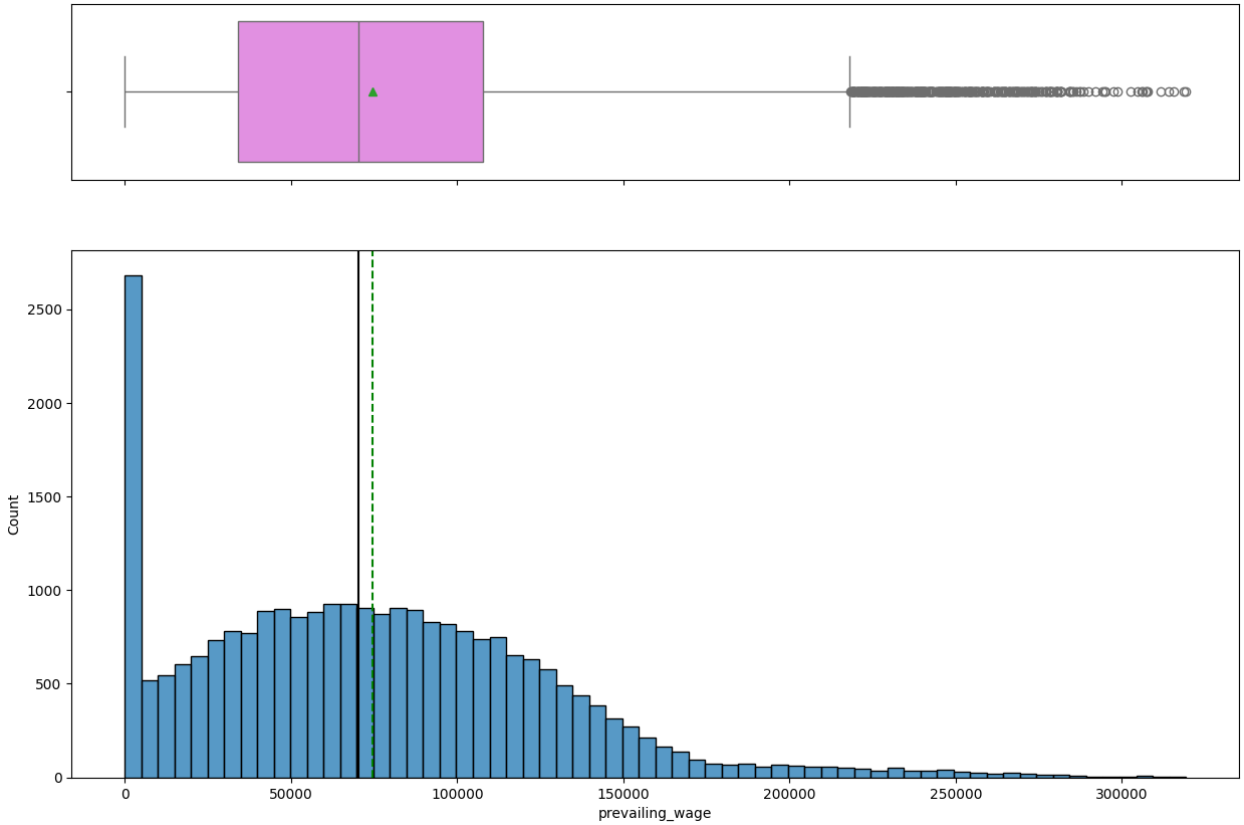
- The Data Frame has 25480 rows and 12 columns, with no missing values.
- The numerical columns are no_of_employees, yr_of_estab, prevailing_wage and remaining columns are of data type object.
- The absence of null values ensures data reliability.

Univariate Analysis:

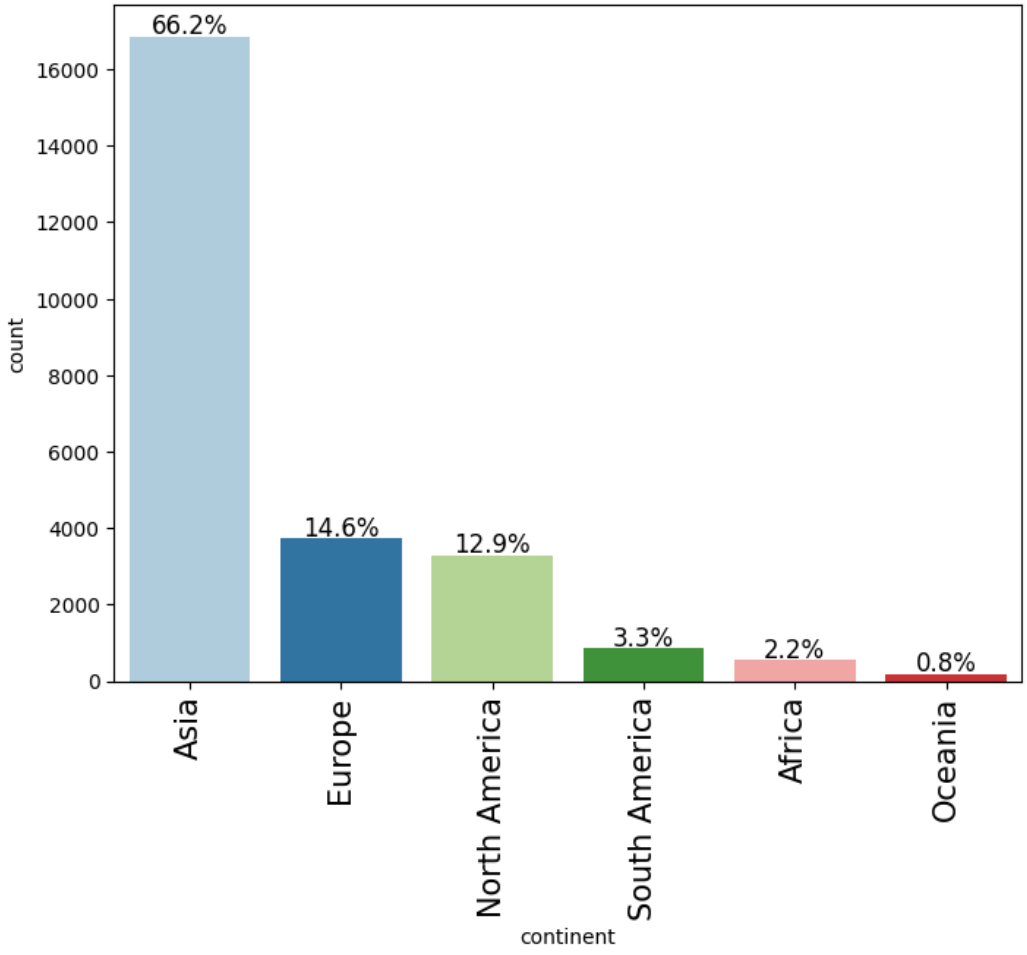
Observations on number of employees



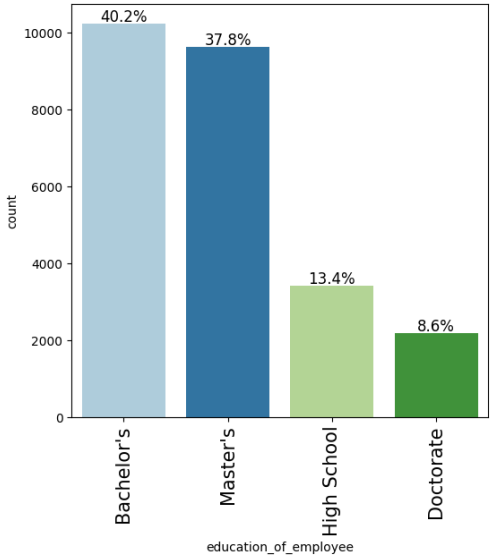
Observations on prevailing wage: Observations on Continent:



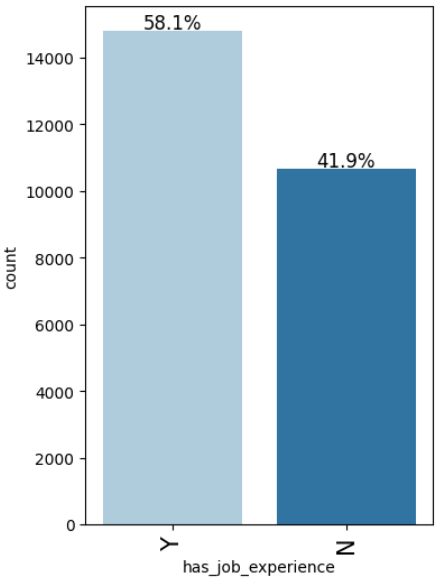
Observations on Continent:



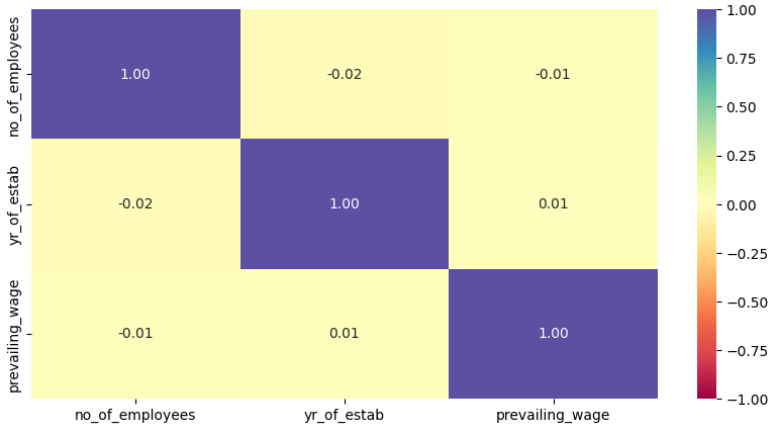
Observations on education of employee



Obsevations on Job Experience:



Bivariate Analysis:-



Problem 1 - Data Pre-processing

Prepare the data for modelling: - Outlier Detection(treat, if needed) - Feature Engineering / drop redundant features (if needed) - Encode the data - Train-test split

Encoding data- The encoding data with string values refers to the process of converting categorical variables expressed as strings into numerical representations that machine learning algorithms can work with. In this dataset there are 2 columns vote and gender which are categorical, so before analysis dummy variable creation and dropping the first column need to be done.

Replacing the negative values in the no_of_employees column with their absolute values -

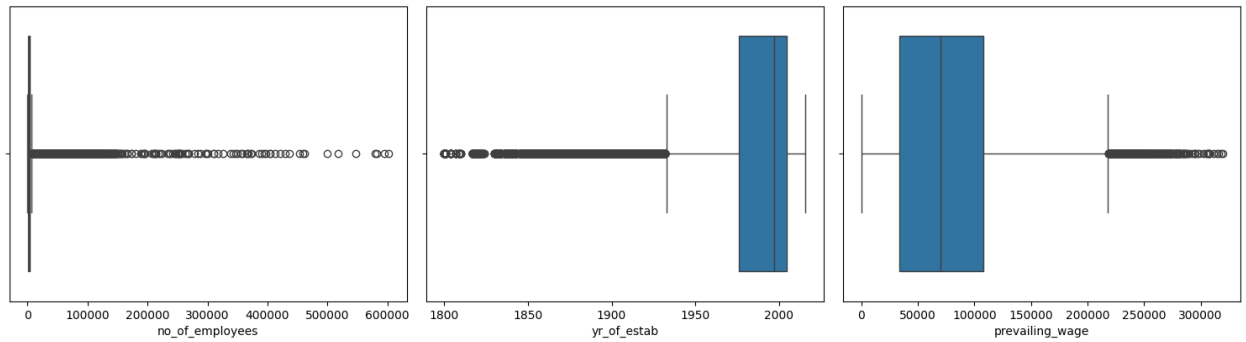
```
33
```

Observations-

- There were 33 negative values and was replaced by its absolute values

Dropping case_id column- It was required only to check if there was continuity in the dataset. It is not required for our modelling and therefore we drop it to proceed

Check for Outliers-



Observations-

- We do observe there are outliers and decide not to treat them, as they seem genuine outliers.

Encoding categorical values in the dataset-

ustab	prevailing_wage	continent_Africa	continent_Asia	continent_Europe	continent_North America	continent_Oceania	continent_South America	education_of_employee_Bachelor's	education_of_employee_Doctorate	education_of_employee_High School	education_of_employee_Master's	has_job_e
2007	592.2029	0	1	0	0	0	0	0	0	1	0	
2002	83425.6500	0	1	0	0	0	0	0	0	0	1	
2008	122996.8600	0	1	0	0	0	0	1	0	0	0	
1897	83434.0300	0	1	0	0	0	0	1	0	0	0	
2005	149907.3900	1	0	0	0	0	0	0	0	0	1	

We use dummies function to encode the categorical values.

Splitting dataset-

Splitting a dataset into training and testing subsets is essential to evaluate a machine learning model's performance on unseen data, preventing overfitting, and ensuring the model's ability to generalize to real- world situations.

Typically, in machine learning, 'X' represents the input features, and 'y' represents the target variable that the model aims to predict.

Splitting dataset into 'X' and 'y' i.e. 'train' and 'test' sets in 70:30.

Now we have 70% data for training and 30% for testing after model

creation.

```
Shape of Training set :  (17836, 28)
Shape of test set :  (7644, 28)
```

This is the shape of training and testing set after splitting into 70:30 ratio.

Problem 1 - Model Building - Bagging
- Build a Bagging classifier - Build a Random forest classifier - Check the performance of the models across train and test set using different metrics and comment on the same

Bagging-

```
BaggingClassifier
BaggingClassifier(random_state=1)
```

From the image we see that bagging classifer is built keeping random_state=1

Confusion matrix of training data-

```
Confusion Matrix for Train Data:
[[ 6247   100]
 [  187 12576]]
```

Validation metrices of training data-

	Accuracy	Recall	Precision	F1
0	0.984982	0.985348	0.992111	0.988718

1. There are 6247 true negative predictions (TN), where the model correctly predicted negative outcomes.
2. There are 100 false positive predictions (FP), where the model incorrectly predicted positive outcomes when the actual outcome was negative.
3. There are 187 false negative predictions (FN), where the model incorrectly predicted negative outcomes when the actual outcome was positive.
4. There are 12576 true positive predictions (TP), where the model correctly predicted positive outcomes.

Observations-

These metrics indicate that the model has performed quite well on the training data, with high accuracy, precision, recall, and F1 score.

Confusion matrix of testing data-

```
Confusion Matrix for Test Data:
[[1126  989]
 [ 940 3315]]
```

Validation metrices of testing data-

	Accuracy	Recall	Precision	F1
0	0.697174	0.779083	0.770214	0.774623

1. There are 1126 true negative predictions (TN), where the model correctly predicted negative outcomes.
2. There are 989 false positive predictions (FP), where the model incorrectly predicted positive outcomes when the actual outcome was negative.
3. There are 940 false negative predictions (FN), where the model incorrectly predicted negative outcomes when the actual outcome was positive.
4. There are 3315 true positive predictions (TP), where the model correctly predicted positive outcomes.

Observations-

These metrics indicate that the model's performance on the testing data is reasonable, with a good balance between precision and recall. However, there are a considerable number of false positive and false negative predictions, which might need further investigation to improve the model's performance.

Hyperparameter Tuning - Bagging Classifier –

```
BaggingClassifier(max_samples=0.5, n_estimators=100, random_state=1)
```

max_samples=0.5: This parameter specifies the proportion of samples to draw from the dataset to train each base estimator (individual model) within the ensemble. Here, 0.5 indicates that each base estimator will be trained on a random sample containing 50% of the total number of samples in the dataset.

n_estimators=100: This parameter defines the number of base estimators (individual models) to include in the ensemble. In this case, 100 indicates that the Bagging Classifier will consist of 100 base estimators.

Hyperparameter Tuning - Bagging Classifier training confusion matrix –

```
Confusion Matrix for Train Data (Tuned Estimator):
[[ 5691   656]
 [  201 12562]]
```

Hyperparameter Tuning - Bagging Classifier training validation metrices –

	Accuracy	Recall	Precision	F1
0	0.955154	0.984251	0.950371	0.967014

1. True Negatives (TN): There are 5691 true negative predictions, where the model correctly predicted negative outcomes.
2. False Positives (FP): There are 656 false positive predictions, where the model incorrectly predicted positive outcomes when the actual outcome was negative.
3. False Negatives (FN): There are 201 false negative predictions, where the model

incorrectly predicted negative outcomes when the actual outcome was positive.

4. True Positives (TP): There are 12562 true positive predictions, where the model correctly predicted positive outcomes.

Observations-

These metrics indicate that the model has performed quite well on the training data, with high accuracy, precision, recall, and F1 score. The high values of precision and recall suggest that the model is effective in identifying both positive and negative outcomes.

Hyperparameter Tuning - Bagging Classifier testing confusion matrix –

```
Confusion Matrix for Test Data (Tuned Estimator):
[[1021 1094]
 [ 642 3613]]
```

Hyperparameter Tuning - Bagging Classifier testing validation metrices –

	Accuracy	Recall	Precision	F1
0	0.727473	0.849119	0.76758	0.806293

1. True Negatives (TN): There are 1021 instances where the model correctly predicted negative outcomes.
2. False Positives (FP): There are 1094 instances where the model incorrectly predicted positive outcomes when the actual outcome was negative.
3. False Negatives (FN): There are 642 instances where the model incorrectly predicted negative outcomes when the actual outcome was positive.
4. True Positives (TP): There are 3613 instances where the model correctly predicted positive outcomes.

Observations-

1. The model exhibits a moderate number of true positive and true negative predictions.
2. The accuracy of approximately 72.75% suggests that around 72.75% of the predictions on the testing set were correct.
3. The recall (sensitivity) of approximately 84.91% indicates that the model correctly identified around 84.91% of the actual positive cases.
4. The precision of approximately 76.76% suggests that when the model predicted a positive outcome, it was correct around 76.76% of the time.
5. The F1 score, which is the harmonic mean of precision and recall, is approximately 80.63%. It provides a balance between precision and recall.

Random Forest-

```
RandomForestClassifier(class_weight='balanced', random_state=1)
```

From the image we see that random forest classifer is built keeping random_state=1 and class_weight as balanced.

Confusion matrix of training data-

```
Confusion Matrix for Train Data:
[[ 6345     2]
 [    0 12763]]
```

Validation metrices of training data-

	Accuracy	Recall	Precision	F1
0	0.999895	1.0	0.999843	0.999922

1. True Negatives (TN): There are 6345 instances where the model correctly predicted negative outcomes.
2. False Positives (FP): There are 2 instances where the model incorrectly predicted positive outcomes when the actual outcome was negative.
3. False Negatives (FN): There are 0 instances where the model incorrectly predicted negative outcomes when the actual outcome was positive.
4. True Positives (TP): There are 12763 instances where the model correctly predicted positive outcomes.

Observations-

1. The model exhibits an extremely high accuracy of approximately 99.99%, indicating that the vast majority of predictions on the training set were correct.
2. The recall (sensitivity) of 100% suggests that the model correctly identified all actual positive cases, making no false negatives.
3. The precision of approximately 99.98% suggests that when the model predicted a positive outcome, it was correct almost 99.98% of the time.
4. The F1 score, which is the harmonic mean of precision and recall, is approximately 99.99%. It provides a balance between precision and recall.

Confusion matrix of testing data-

```
Confusion Matrix for Test Data:
[[ 997 1118]
 [ 666 3589]]
```

Validation metrices of testing data-

	Accuracy	Recall	Precision	F1
0	0.719937	0.843478	0.762481	0.800937

1. True Negatives (TN): There are 997 instances where the model correctly predicted negative outcomes.
2. False Positives (FP): There are 1118 instances where the model incorrectly

predicted positive outcomes when the actual outcome was negative.

3. False Negatives (FN): There are 666 instances where the model incorrectly predicted negative outcomes when the actual outcome was positive.

4. True Positives (TP): There are 3589 instances where the model correctly predicted positive outcomes.

Observations-

1. The model exhibits an accuracy of approximately 71.99%, suggesting that a substantial portion of predictions on the test set were correct, albeit lower than the training set accuracy.
2. The recall (sensitivity) of approximately 84.35% indicates that the model correctly identified around 84.35% of actual positive cases, which is slightly lower than the training set recall.
3. The precision of approximately 76.25% suggests that when the model predicted a positive outcome, it was correct around 76.25% of the time, which is lower than the precision on the training set.
4. The F1 score, which is the harmonic mean of precision and recall, is approximately 80.09%. It provides a balance between precision and recall, and it's slightly lower than the F1 score on the training set.

Hyperparameter Tuning – Random Forest Classifier for training –

```
RandomForestClassifier(max_depth=10, n_estimators=200, oob_score=True,
                       random_state=1)
```

```
Confusion Matrix for Train Data (Tuned Estimator):
[[ 3395  2952]
 [ 1327 11436]]
```

	Accuracy	Recall	Precision	F1
0	0.776086	0.896028	0.794829	0.8424

1. True Negatives (TN): 3395 instances where the model correctly predicted negative outcomes.
2. False Positives (FP): 2952 instances where the model incorrectly predicted positive outcomes when the actual outcome was negative.
3. False Negatives (FN): 1327 instances where the model incorrectly predicted negative outcomes when the actual outcome was positive.
4. True Positives (TP): 11436 instances where the model correctly predicted positive outcomes.

Observations-

1. The model exhibits an accuracy of approximately 77.61% on the new training data.
2. The recall (sensitivity) of approximately 89.60% indicates that the model correctly identified around 89.60% of actual positive cases.
3. The precision of approximately 79.48% suggests that when the model predicted a positive outcome, it was correct around 79.48% of the time.
4. The F1 score, which is the harmonic mean of precision and recall, is approximately 84.24%.

Hyperparameter Tuning – Random Forest Classifier for testing –

```
Confusion Matrix for Test Data (Tuned Estimator):
[[1009 1106]
 [ 521 3734]]
```

	Accuracy	Recall	Precision	F1
0	0.744584	0.877556	0.771488	0.821111

1. True Negatives (TN): 1009 instances where the model correctly predicted negative outcomes.
2. False Positives (FP): 1106 instances where the model incorrectly predicted positive outcomes when the actual outcome was negative.
3. False Negatives (FN): 521 instances where the model incorrectly predicted negative outcomes when the actual outcome was positive.
4. True Positives (TP): 3734 instances where the model correctly predicted positive outcomes..

Observations-

1. The model exhibits an accuracy of approximately 74.46% on the new testing data.
2. The recall (sensitivity) of approximately 87.76% indicates that the model correctly identified around 87.76% of actual positive cases.
3. The precision of approximately 77.15% suggests that when the model predicted a positive outcome, it was correct around 77.15% of the time.
4. The F1 score, which is the harmonic mean of precision and recall, is approximately 82.11%.

ADA Boosting Classifier for training –

```
Confusion Matrix for Train Data (AdaBoost Classifier):
[[ 2808  3539]
 [ 1426 11337]]
```

	Accuracy	Recall	Precision	F1
0	0.740188	0.888271	0.7621	0.820363

1. True Negatives (TN): 2808 instances where the model correctly predicted negative outcomes.
2. False Positives (FP): 3539 instances where the model incorrectly predicted positive outcomes when the actual outcome was negative.
3. False Negatives (FN): 1426 instances where the model incorrectly predicted negative outcomes when the actual outcome was positive.
4. True Positives (TP): 11337 instances where the model correctly predicted positive outcomes.

Observations-

1. The model exhibits an accuracy of approximately 74.02% on the training data.
2. The recall (sensitivity) of approximately 88.83% indicates that the model correctly identified around 88.83% of actual positive cases.
3. The precision of approximately 76.21% suggests that when the model predicted a positive outcome, it was correct around 76.21% of the time.
4. The F1 score, which is the harmonic mean of precision and recall, is approximately 82.04%.

ADA Boosting Classifier for testing –

```
Confusion Matrix for Test Data (AdaBoost Classifier):
[[ 886 1229]
 [ 485 3770]]
```

	Accuracy	Recall	Precision	F1
0	0.730926	0.886016	0.754151	0.814783

1. True Negatives (TN): 886 instances where the model correctly predicted negative outcomes.
2. False Positives (FP): 1229 instances where the model incorrectly predicted positive outcomes when the actual outcome was negative.
3. False Negatives (FN): 485 instances where the model incorrectly predicted negative outcomes when the actual outcome was positive.
4. True Positives (TP): 3770 instances where the model correctly predicted positive outcomes.

Observations-

5. The model exhibits an accuracy of approximately 73.09% on the testing data.
6. The recall (sensitivity) of approximately 88.60% indicates that the model correctly identified around 88.60% of actual positive cases.
7. The precision of approximately 75.42% suggests that when the model predicted a positive outcome, it was correct around 75.42% of the time.
8. The F1 score, which is the harmonic mean of precision and recall, is approximately 81.48%.

Hyperparameter Tuning – ADA Boosting Classifier for training –

```
Confusion Matrix for Train Data (Tuned AdaBoost Classifier):
[[ 2790  3557]
 [ 1421 11342]]
```

	Accuracy	Recall	Precision	F1
0	0.739508	0.888663	0.761259	0.820042

1. True Negatives (TN): 2790 instances where the model correctly predicted negative outcomes.
2. False Positives (FP): 3557 instances where the model incorrectly predicted positive outcomes when the actual outcome was negative.
3. False Negatives (FN): 1421 instances where the model incorrectly predicted negative outcomes when the actual outcome was positive.
4. True Positives (TP): 11342 instances where the model correctly predicted positive outcomes.

Observations-

1. The model exhibits an accuracy of approximately 73.95% on the training data.
2. The recall (sensitivity) of approximately 88.87% indicates that the model correctly identified around 88.87% of actual positive cases.
3. The precision of approximately 76.13% suggests that when the model predicted a positive outcome, it was correct around 76.13% of the time.
4. The F1 score, which is the harmonic mean of precision and recall, is approximately 82.00%.

Hyperparameter Tuning – ADA Boosting Classifier for testing –

```
Confusion Matrix for Test Data (Tuned AdaBoost Classifier):
[[ 882 1233]
 [ 481 3774]]
```

	Accuracy	Recall	Precision	F1
0	0.730926	0.886957	0.753745	0.814943

1. True Negatives (TN): 882 instances where the model correctly predicted negative outcomes.
2. False Positives (FP): 1233 instances where the model incorrectly predicted positive outcomes when the actual outcome was negative.
3. False Negatives (FN): 481 instances where the model incorrectly predicted negative outcomes when the actual outcome was positive.
4. True Positives (TP): 3774 instances where the model correctly predicted positive outcomes.

Observations-

1. The model exhibits an accuracy of approximately 73.09% on the testing data.
2. The recall (sensitivity) of approximately 88.70% indicates that the model correctly identified around 88.70% of actual positive cases.
3. The precision of approximately 75.37% suggests that when the model predicted a positive outcome, it was correct around 75.37% of the time.
4. The F1 score, which is the harmonic mean of precision and recall, is approximately 81.49%.

Model Performance Comparison and Final Model Selection:

The training performance comparision:

Training performance comparison:

	Bagging Classifier	Tuned Bagging Classifier	Random Forest	Tuned Random Forest	Adaboost Classifier	Tuned Adaboost Classifier
Accuracy	0.984982	0.955154	0.999895	0.776086	0.740188	0.739508
Recall	0.985348	0.984251	1.000000	0.896028	0.888271	0.888663
Precision	0.992111	0.950371	0.999843	0.794829	0.762100	0.761259
F1	0.988718	0.967014	0.999922	0.842400	0.820363	0.820042

The testing performance comparision:

Testing performance comparison:

	Bagging Classifier	Tuned Bagging Classifier	Random Forest	\
Accuracy	0.697174	0.727473	0.719937	
Recall	0.779083	0.849119	0.843478	
Precision	0.770214	0.767580	0.762481	
F1	0.774623	0.806293	0.800937	

	Tuned Random Forest	Adaboost Classifier	Tuned Adaboost Classifier
Accuracy	0.744584	0.730926	0.730926
Recall	0.877556	0.886016	0.886957
Precision	0.771488	0.754151	0.753745
F1	0.821111	0.814783	0.814943

Observations-

Training Data Performance:

a. Accuracy: Among the models, the Random Forest achieved the highest accuracy of 99.99%, closely followed by the Bagging Classifier with 98.50% accuracy.
b. Recall: Random Forest and the Tuned Bagging Classifier achieved perfect recall scores of 1.0, indicating that they correctly identified all

positive cases. The Bagging Classifier also performed exceptionally well with a recall score of 0.985.

c. Precision: The Random Forest and the Bagging Classifier had high precision scores, indicating that when they predicted positive outcomes, they were correct approximately 99.98% and 99.21% of the time, respectively.
d. F1 Score: Random Forest achieved the highest F1 score of 99.99%, followed by the Bagging Classifier with 98.87%.

Testing Data Performance:

a. Accuracy: Random Forest achieved the highest accuracy of 74.46% on the testing data, followed closely by the Tuned Random Forest with 74.46% accuracy.
b. Recall: The Tuned Adaboost Classifier performed the best in terms of recall, correctly identifying approximately 88.70% of actual positive cases.
c. Precision: The Tuned Adaboost Classifier had the highest precision of approximately 75.37% among all models.
d. F1 Score: Tuned Adaboost Classifier also achieved the highest F1 score of approximately 81.49%.

Key Takeaways:

a. Random Forest: While Random Forest performed exceptionally well on the training data, achieving near-perfect scores, its performance on the testing data was slightly lower. However, it still maintained one of the highest accuracies among all models.
b. Tuned Adaboost Classifier: This model showed the best balance of precision and recall on the testing data, indicating its effectiveness in correctly identifying positive cases while minimizing false positives.
c. Recommendation: Considering the performance on both training and testing data, the Tuned Adaboost Classifier appears to be the most reliable model for predicting the outcome. However, further fine-tuning and evaluation may be required based on specific business requirements and the importance of precision and recall trade-offs.

In conclusion, while Random Forest performed exceptionally well on the training data, the **Tuned Adaboost Classifier** demonstrated better generalization and balance between precision and recall on the testing data, making it the preferred choice for deployment in real-world scenarios.

Problem 2 - Perform Exploratory Data Analysis and Text Pre-processing

- Perform exploratory data analysis - Missing Value Checking and Treatment - Feature Engineering - Analysis of tweets - Analysis of twitter activity - Plot wordcloud - Text pre-processing

First 5 rows and last 5 rows:

	Unnamed: 0	id	link	content	date	retweets	favorites	mentions	hashtags	geo	Sentiment
0	0	1698308935	https://twitter.com/realDonaldTrump/status/169...	Be sure to tune in and watch Donald Trump on L...	2009-05-04 20:54:25	500	868	NaN	NaN	NaN	positive
1	1	1701461182	https://twitter.com/realDonaldTrump/status/170...	Donald Trump will be appearing on The View tom...	2009-05-05 03:00:10	33	273	NaN	NaN	NaN	positive
2	2	1737479987	https://twitter.com/realDonaldTrump/status/173...	Donald Trump reads Top Ten Financial Tips on L...	2009-05-08 15:38:08	12	18	NaN	NaN	NaN	positive
3	3	1741160716	https://twitter.com/realDonaldTrump/status/174...	New Blog Post: Celebrity Apprentice Finale and...	2009-05-08 22:40:15	11	24	NaN	NaN	NaN	positive
4	4	1773561338	https://twitter.com/realDonaldTrump/status/177...	"My persona will never be that of a wallflower...	2009-05-12 16:07:28	1399	1965	NaN	NaN	NaN	positive

	Unnamed: 0	id	link	content	date	retweets	favorites	mentions	hashtags	geo	Sentiment
41117	41117	1218962544372670467	https://twitter.com/realDonaldTrump/status/121...	I have never seen the Republican Party as Stro...	2020-01-19 19:24:52	32620	213817	NaN	NaN	NaN	positive
41118	41118	1219004689716412416	https://twitter.com/realDonaldTrump/status/121...	Now Mini Mike Bloomberg is critical of Jack Wi...	2020-01-19 22:12:20	36239	149571	NaN	NaN	NaN	negative
41119	41119	1219053709428248576	https://twitter.com/realDonaldTrump/status/121...	I was thrilled to be back in the Great State o...	2020-01-20 01:27:07	16588	66944	NaN	#	NaN	positive
41120	41120	1219066007731310593	https://twitter.com/realDonaldTrump/status/121...	"In the House, the President got less due proc...	2020-01-20 02:16:00	20599	81921	@@@	NaN	NaN	negative
41121	41121	1219076533354037249	https://twitter.com/realDonaldTrump/status/121...	A great show! Check it out tonight at 9pm. @ F...	2020-01-20 02:57:49	7947	34902	@	NaN	NaN	positive

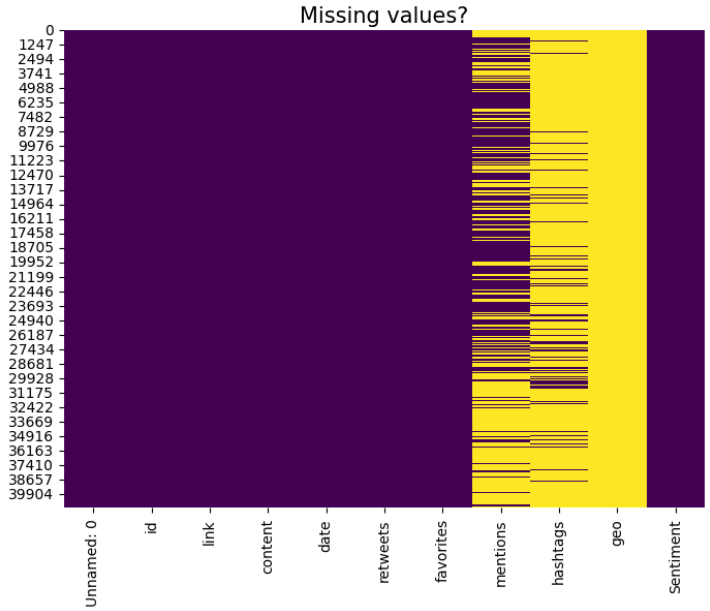
Logistic Model creation-

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41122 entries, 0 to 41121
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Unnamed: 0   41122 non-null  int64  
 1   id           41122 non-null  int64  
 2   link         41122 non-null  object 
 3   content      41122 non-null  object 
 4   date         41122 non-null  object 
 5   retweets     41122 non-null  int64  
 6   favorites    41122 non-null  int64  
 7   mentions     22467 non-null  object 
 8   hashtags     5810 non-null   object 
 9   geo          0 non-null      float64
 10  Sentiment    41122 non-null  object 
dtypes: float64(1), int64(4), object(6)
memory usage: 3.5+ MB
```

The Data Frame contains 41,122 entries and 11 columns, with 'mentions','geo' and 'hashtags' having missing values. The data includes tweet information such as content, date, retweets, favorites, and sentiment.

But 'geo' column is completely empty. We have to get rid of these missing values as it does not make any sense in considering.

Visualizing missing values-



As mention , ‘mentions’, ‘hashtags’ and ‘geo’ columns are the only columns that has missing values

In detail:

	Zero Values	Missing Values	% of Total Values	Total Zero Missing Values	% Total Zero Missing Values	Data Type
geo	0	41122	100.0	41122	100.0	float64
hashtags	0	35312	85.9	35312	85.9	object
mentions	0	18655	45.4	18655	45.4	object

The dataframe has 100% missing values in the 'geo' column, while the 'hashtags' and 'mentions' columns contain 85.9% and 45.4% missing values, respectively. There are no zero values in any of the columns. The data types include float64 for 'geo' and object for 'hashtags' and 'mentions'.

Extracting date column in to separate columns-

	content	date	retweets	favorites	Sentiment	year	month	hour
0	Be sure to tune in and watch Donald Trump on L...	4	500	868	positive	2009	5	20
1	Donald Trump will be appearing on The View tom...	5	33	273	positive	2009	5	3
2	Donald Trump reads Top Ten Financial Tips on L...	8	12	18	positive	2009	5	15
3	New Blog Post: Celebrity Apprentice Finale and...	8	11	24	positive	2009	5	22
4	"My persona will never be that of a wallflower...	12	1399	1965	positive	2009	5	16

As we can see in the above image, the date column was split in to separate columns of year, month, date and hour .

Most liked tweet during Presidential year-

```
A$AP Rocky released from prison and on his way home to the United States from Sweden. It was a Rocky Week, get home ASAP A$AP!
2019
```

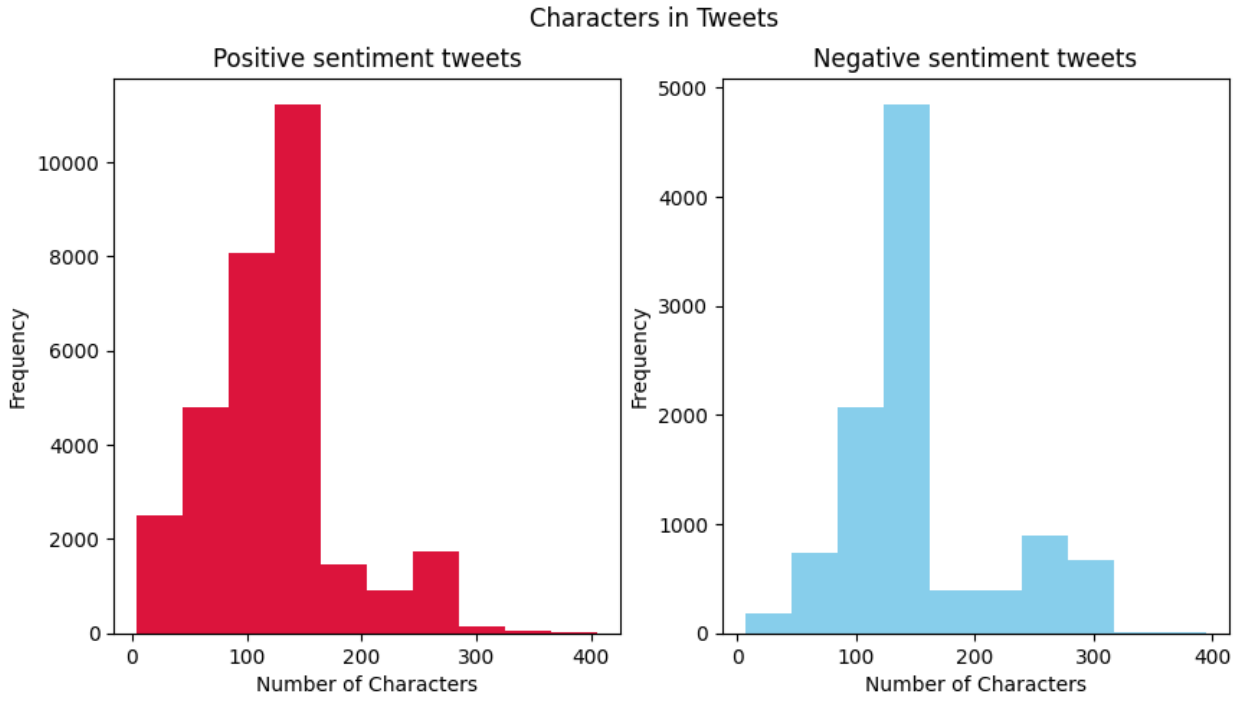
The first line is the most liked tweet and the second line is the year of most favorites tweet

Most retweeted tweet during presidential year-

```
# FraudNewsCNN # FNNpic.twitter.com/WYUnHjjUjg
2019
```

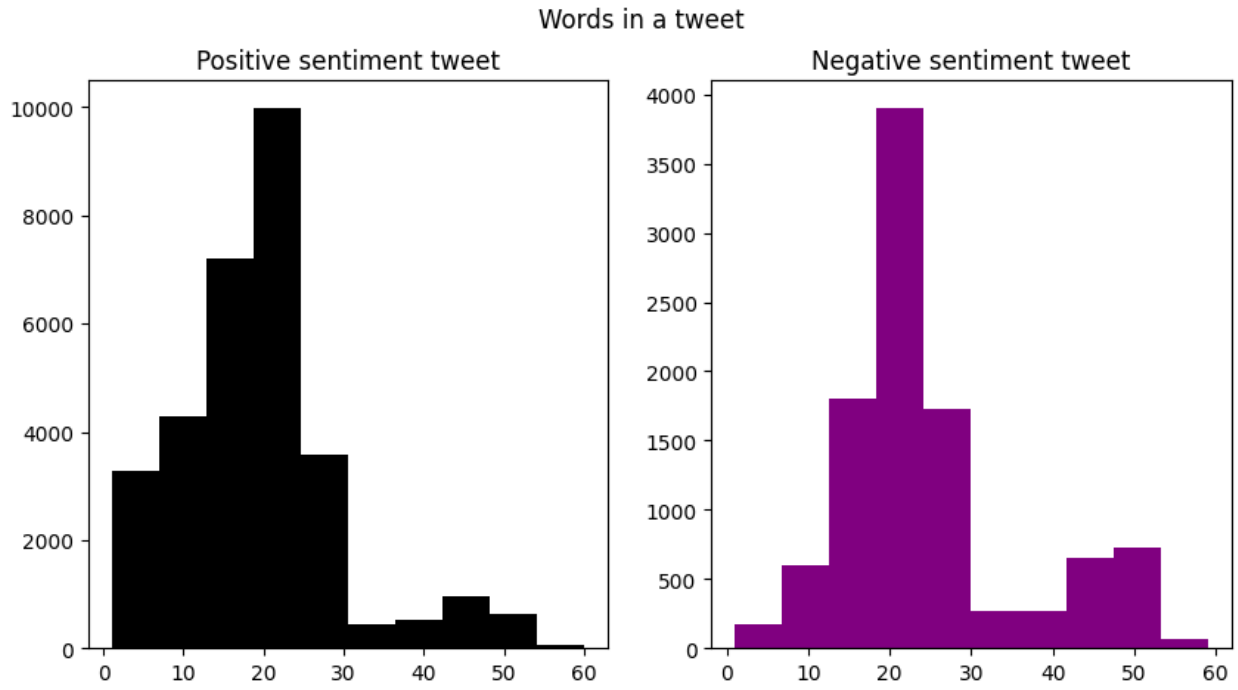
The first line is the most liked retweeted tweet and the second line is the year of most favorites tweet

Number of characters in tweets



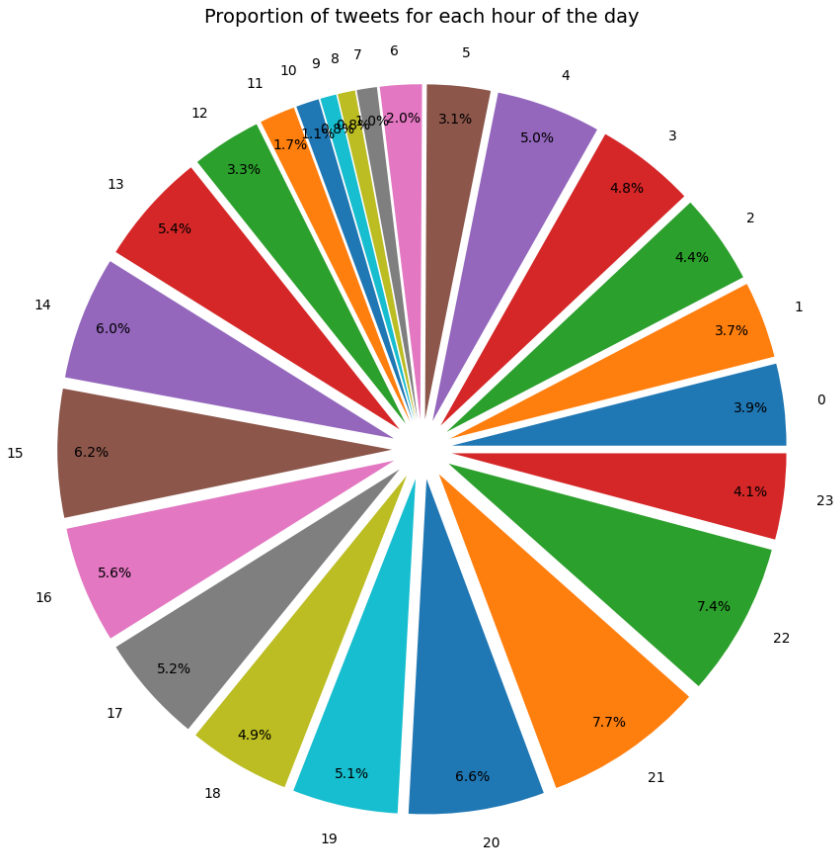
From the above graph we can infer that, there are equal number of characters in both positive and negative tweet.

Number of words in tweets:

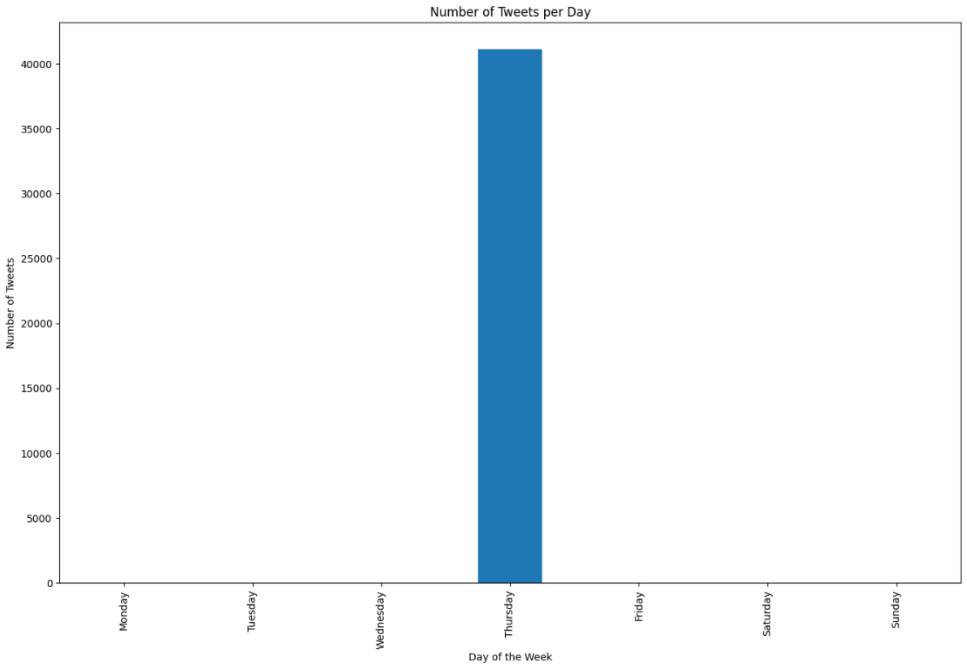


From the above graph we can infer that, there are almost equal number of words in both positive and negative tweet.

Portion of tweets for each hour of the day:



Number of tweets per day:



From the above graph we can infer that, only on Thursdays the tweets were made

Text Preprocessing - removal of http links-

	content	date	retweets	favorites	Sentiment	year	month	hour
0	Be sure to tune in and watch Donald Trump on L...	1970-01-01 00:00:00.000000004	500	868	positive	2009	5	20
1	Donald Trump will be appearing on The View tom...	1970-01-01 00:00:00.000000005	33	273	positive	2009	5	3
2	Donald Trump reads Top Ten Financial Tips on L...	1970-01-01 00:00:00.000000008	12	18	positive	2009	5	15
3	New Blog Post: Celebrity Apprentice Finale and...	1970-01-01 00:00:00.000000008	11	24	positive	2009	5	22
4	"My persona will never be that of a wallflower...	1970-01-01 00:00:00.000000012	1399	1965	positive	2009	5	16

Pre Processing : Removal of number-

	content	date	retweets	favorites	Sentiment	year	month	hour
0	Be sure to tune in and watch Donald Trump on L...	1970-01-01 00:00:00.000000004	500	868	positive	2009	5	20
1	Donald Trump will be appearing on The View tom...	1970-01-01 00:00:00.000000005	33	273	positive	2009	5	3
2	Donald Trump reads Top Ten Financial Tips on L...	1970-01-01 00:00:00.000000008	12	18	positive	2009	5	15
3	New Blog Post: Celebrity Apprentice Finale and...	1970-01-01 00:00:00.000000008	11	24	positive	2009	5	22
4	"My persona will never be that of a wallflower...	1970-01-01 00:00:00.000000012	1399	1965	positive	2009	5	16

Pre Processing : Tokenization-

	content	date	retweets	favorites	Sentiment	year	month	hour
0	[Be, sure, to, tune, in, and, watch, Donald, T...	1970-01-01 00:00:00.000000004	500	868	positive	2009	5	20
1	[Donald, Trump, will, be, appearing, on, The, ...	1970-01-01 00:00:00.000000005	33	273	positive	2009	5	3
2	[Donald, Trump, reads, Top, Ten, Financial, Ti...	1970-01-01 00:00:00.000000008	12	18	positive	2009	5	15
3	[New, Blog, Post, :, Celebrity, Apprentice, Fi...	1970-01-01 00:00:00.000000008	11	24	positive	2009	5	22
4	[", My, persona, will, never, be, that, of, a...	1970-01-01 00:00:00.000000012	1399	1965	positive	2009	5	16

From the above image we can infer that, the column ‘content’ was tokenized and stored as each word separate element in list.

Pre Processing : lower case conversion-

	content	date	retweets	favorites	Sentiment	year	month	hour
0	[be, sure, to, tune, in, and, watch, donald, t...	1970-01-01 00:00:00.000000004	500	868	positive	2009	5	20
1	[donald, trump, will, be, appearing, on, the, ...	1970-01-01 00:00:00.000000005	33	273	positive	2009	5	3
2	[donald, trump, reads, top, ten, financial, ti...	1970-01-01 00:00:00.000000008	12	18	positive	2009	5	15
3	[new, blog, post, :, celebrity, apprentice, fi...	1970-01-01 00:00:00.000000008	11	24	positive	2009	5	22
4	[``, my, persona, will, never, be, that, of, a...	1970-01-01 00:00:00.000000012	1399	1965	positive	2009	5	16

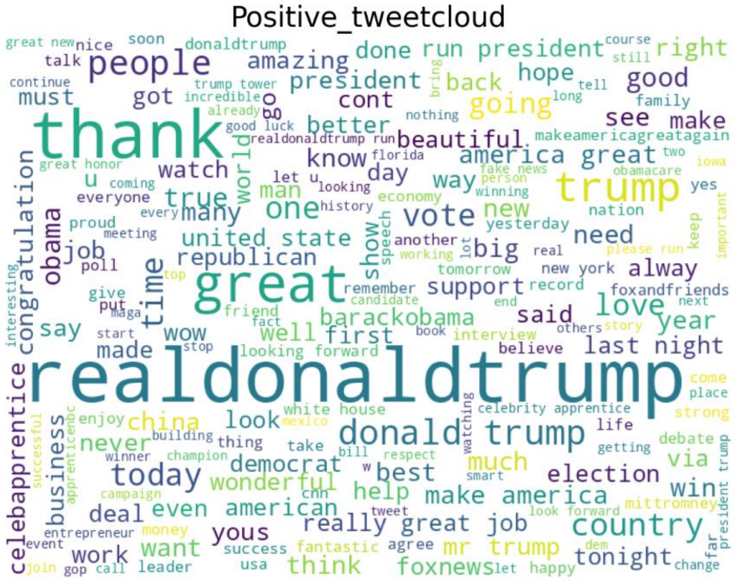
Every character was made sure to be converted in to lowercase, which is required for text analysis.

Pre Processing : Stop words removal-

	content	date	retweets	favorites	Sentiment	year	month	hour
0	[sure, tune, watch, donald, trump, late, night...	1970-01-01 00:00:00.000000004	500	868	positive	2009	5	20
1	[donald, trump, appearing, view, tomorrow, mor...	1970-01-01 00:00:00.000000005	33	273	positive	2009	5	3
2	[donald, trump, reads, top, ten, financial, ti...	1970-01-01 00:00:00.000000008	12	18	positive	2009	5	15
3	[new, blog, post, celebrity, apprentice, final...	1970-01-01 00:00:00.000000008	11	24	positive	2009	5	22
4	[persona, never, wallflower, would, rather, bu...	1970-01-01 00:00:00.000000012	1399	1965	positive	2009	5	16

Stop words were removed from the content, as it does not reveal any meaning.

Positive word cloud-



Negative word cloud-

Negative_tweetcloud