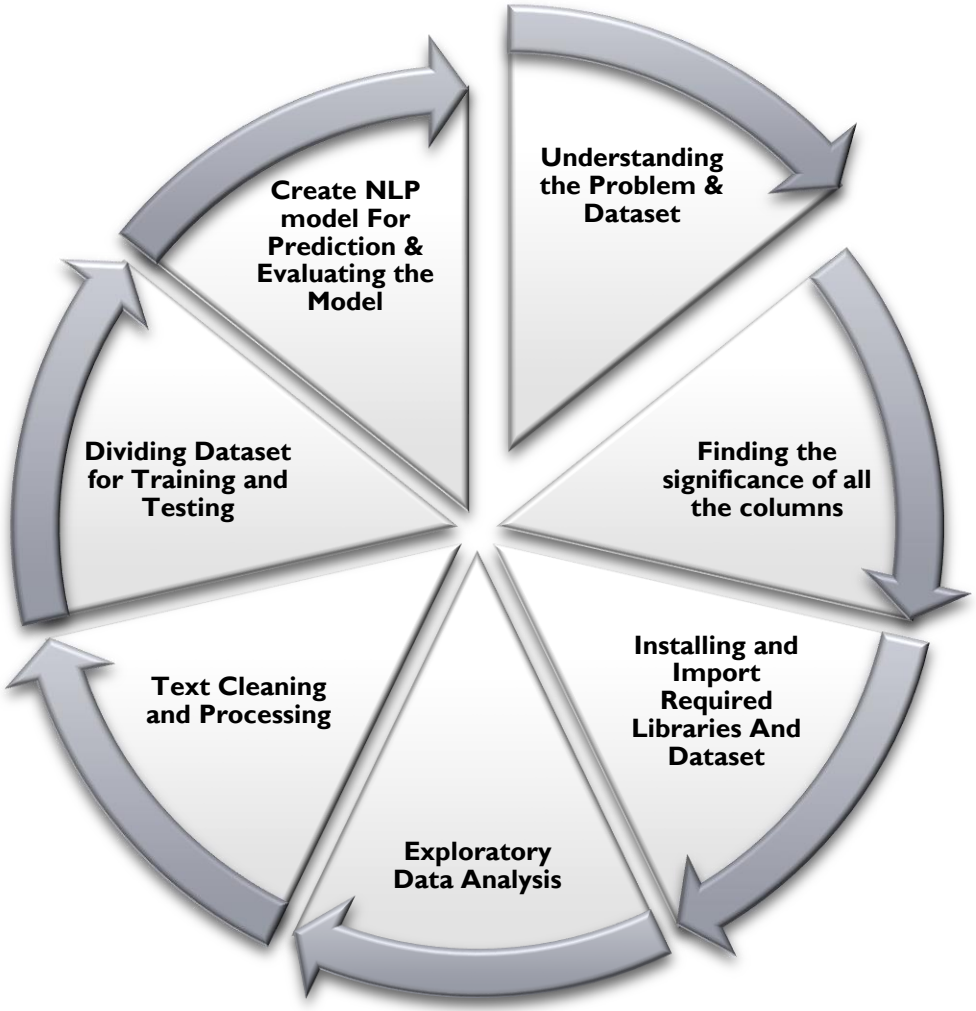
Descriptive Analysis And NLP Techniques

REVIEWS OF MUSICAL INSTRUMENTS.

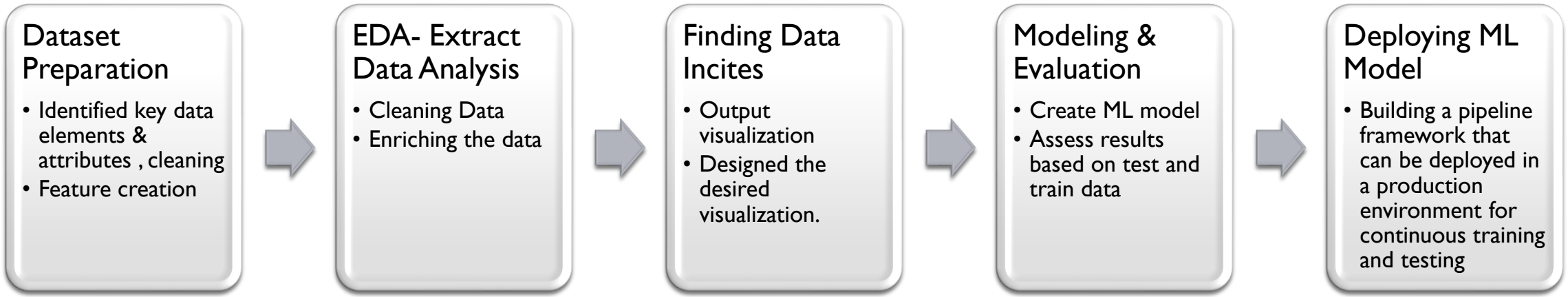
ROADMAP

1. Overview of the Solution Developed
2. The Approach And Design
3. Key considerations
4. Analytical methodology
5. Outcomes
6. Proposed Next Steps

OVERVIEW OF THE SOLUTION DEVELOPED



DESIGN APPROACH



THE APPROACH AND DESIGN

- **Data Source:**

  Data Set Shared by Bank of Ireland for submitting application for the Lead Data Analyst position we have available within our Group Internal Audit team.

- **Approch:**

  The data set provided contain lots of text data which can be process with the help of Natural language processing and the creating ML models on top of the data processed. Hence we will be going with the NLPML implementations for the problem shared for solving

- **Understanding the Problem**
  - Aim of the problem is extract and present key insights and recommendations from the dataset (including text attributes) using a combination of descriptive and natural language processing techniques.
  - Features available are:
    1. reviewerID: Instrument reviewerId
    2. asin: Instrument ID to identify the instruments uniquely
    3. reviewerName: Number of person who has revived the Instrument
    4. helpful: How helpful the relieves were
    5. overall: The rating provided by the reviewer for the instrument
    6. summary : The description about the product as provided by the reviewer
    7. unixReviewTime: Unix time of the review
    8. reviewTime: Date of the review

KEY CONSIDERATIONS

- Columns : reviewerID , asin , reviewerName acts as identifiers and can be ignored
- Column : helpful columns being a list value cannot be used by the ML model directly so need to find alternative way to save the value.(Here we take the ratio of the values present in the column)
- Column : reviewText and summary have same level of information i.e if reviews bad summary wont be good so we can combine those column so that ML model wont have domination of similar features.
- Column : reviewTime being a date column ML models cannot use them directly so need to convert them into separate columns
- Columns : overall have 5 distinct values , we can either consider solving the problem with multiclassification or we can convert it to 2 distinct values as the reviews will be then considered as Positive And Negative based on the algorithm designed
- We have used the concept of collection of text documents to a matrix of token counts - **CountVectorizer**

DESCRIPTIVE ANALYSIS AND ANALYTICS

- We have used descriptive analysis in the ML Code that is summited to gain the incites of the data set

Understand the data types of the columns for data manipulations

```
5]: # Check the dataframe info for datatypes

    df.info()
```

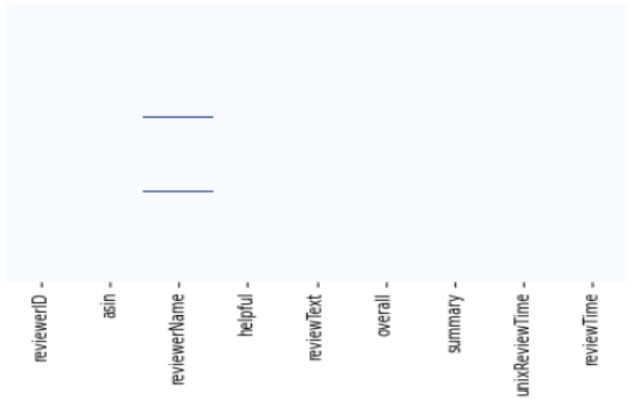
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10261 entries, 0 to 10260
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   reviewerID      10261 non-null  object
 1   asin            10261 non-null  object
 2   reviewerName    10234 non-null  object
 3   helpful         10261 non-null  object
 4   reviewText      10261 non-null  object
 5   overall         10261 non-null  int64
 6   summary         10261 non-null  object
 7   unixReviewTime  10261 non-null  int64
 8   reviewTime      10261 non-null  object
dtypes: int64(2), object(7)
memory usage: 721.6+ KB
```

Finding then null values present in data so as to determine the fixes

```
# check if there are any Null values
print(df.isnull().sum())
sns.heatmap(df.isnull(), yticklabels = False, cbar = False, cmap="Blues")
```

```
reviewerID          0
asin                0
reviewerName       27
helpful             0
reviewText          0
overall             0
summary             0
unixReviewTime      0
reviewTime          0
dtype: int64
```

```
<AxesSubplot:>
```



DESCRIPTIVE ANALYSIS AND ANALYTICS

Understand the data present in the columns

```
print('Data Set size ----' , df.shape)
print()
print('Unique Instruments Reviewed       : ' , len(df.asin.unique()))
print('Unique Reviewers who reviewed     : ' , len(df.reviewerID.unique()))
print('Unique Reviewers Name             : ' , len(df.reviewerName.unique()))
#print('Unique helpful : ' , len(df.helpful.unique()))
print('Unique Messages Shared by reviewer: ' , len(df.reviewText.unique()))
print('Unique Summary Shared by Reviewer : ' , len(df.summary.unique()))
print('Unique unixReviewTime             : ' , len(df.unixReviewTime.unique()))
print('Unique reviewTime                 : ' , len(df.reviewTime.unique()))
print('Unique Overall Rating             : ' , len(df.overall.unique()))
print('Unique Overall Ratings            : ' , df.overall.unique())
```

```
Data Set size ---- (10261, 9)

Unique Instruments Reviewed       :  900
Unique Reviewers who reviewed     :  1429
Unique Reviewers Name             :  1398
Unique Messages Shared by reviewer:  10255
Unique Summary Shared by Reviewer :  8852
Unique unixReviewTime             :  1570
Unique reviewTime                 :  1570
Unique Overall Rating             :  5
Unique Overall Ratings            :  [5 3 4 2 1]
```

Understand the relations between dependent and independent variables

```
df.overall.value_counts()
```

```
5    6938
4    2084
3     772
2     250
1     217
Name: overall, dtype: int64
```

```
df.groupby('overall').describe()
```

overall	helpful															
	count	mean	std	min	25%	50%	75%	max	count	mean	std	min	25%	50%	75%	
1	217.0	0.330553	0.391282	0.0	0.0	0.0	0.6700	1.0	217.0	1.363610e+09	3.693997e+07	1.141344e+09	1.347926e+09	1.370995e+09	1.390349e+09	
2	250.0	0.287600	0.380346	0.0	0.0	0.0	0.5000	1.0	250.0	1.361242e+09	3.770940e+07	1.190678e+09	1.342116e+09	1.369872e+09	1.389506e+09	
3	772.0	0.275687	0.402314	0.0	0.0	0.0	0.6700	1.0	772.0	1.361718e+09	3.633831e+07	1.161389e+09	1.343282e+09	1.369008e+09	1.389053e+09	
4	2084.0	0.282970	0.432186	0.0	0.0	0.0	0.8025	1.0	2084.0	1.359799e+09	3.914760e+07	1.095466e+09	1.342915e+09	1.369138e+09	1.388707e+09	
5	6938.0	0.253758	0.420095	0.0	0.0	0.0	0.6700	1.0	6938.0	1.360608e+09	3.757515e+07	1.096416e+09	1.343606e+09	1.367971e+09	1.388945e+09	

Understand data distribution of dependent and independent variables

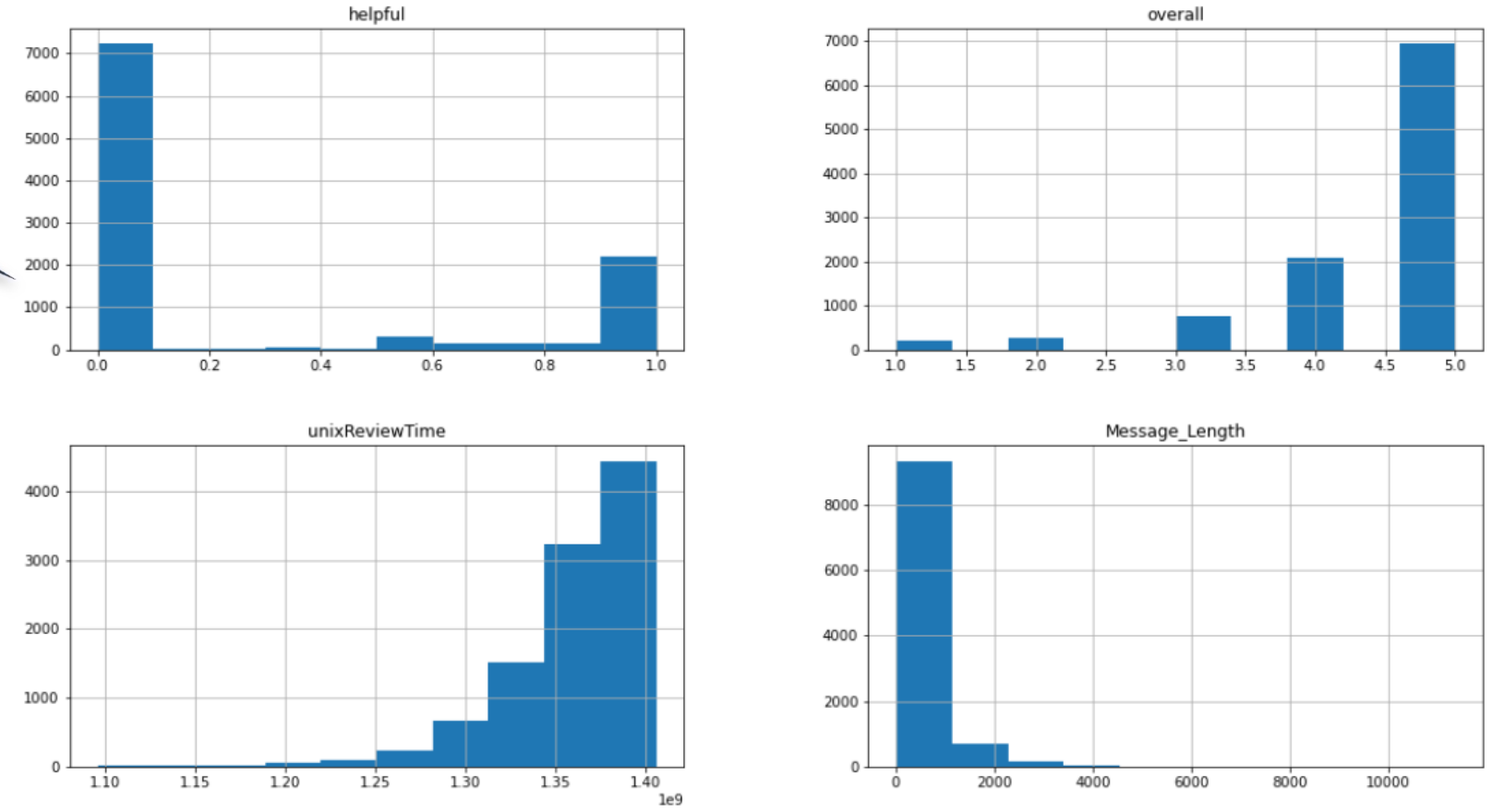
```
df.describe()
```

	helpful	overall	unixReviewTime	Message_Length
count	10261.000000	10261.000000	1.026100e+04	10261.000000
mean	0.263789	4.488744	1.360606e+09	511.277458
std	0.420002	0.894642	3.779735e+07	618.354038
min	0.000000	1.000000	1.095466e+09	15.000000
25%	0.000000	4.000000	1.343434e+09	185.000000
50%	0.000000	5.000000	1.368490e+09	309.000000
75%	0.670000	5.000000	1.388966e+09	581.000000
max	1.000000	5.000000	1.405987e+09	11345.000000

DESCRIPTIVE ANALYSIS AND VISUALIZATIONS

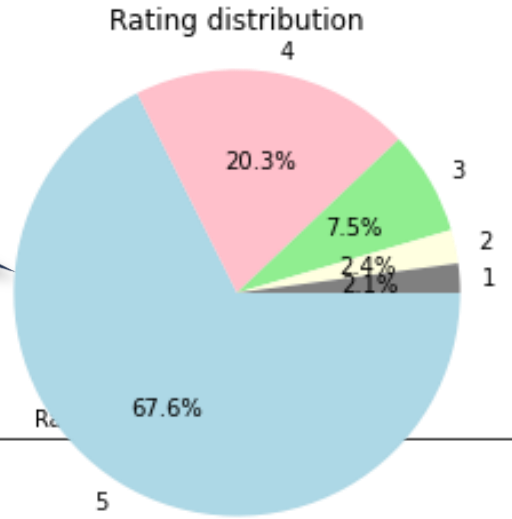
Histogram gives you the distribution view of the data in the respective column in dataset

```
df.hist(figsize=(18,10))
plt.show()
```

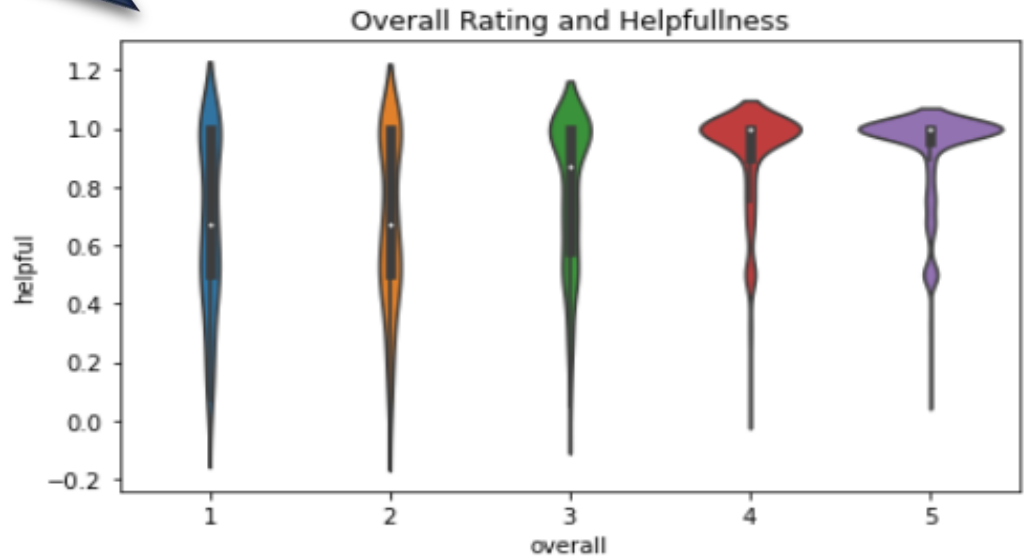


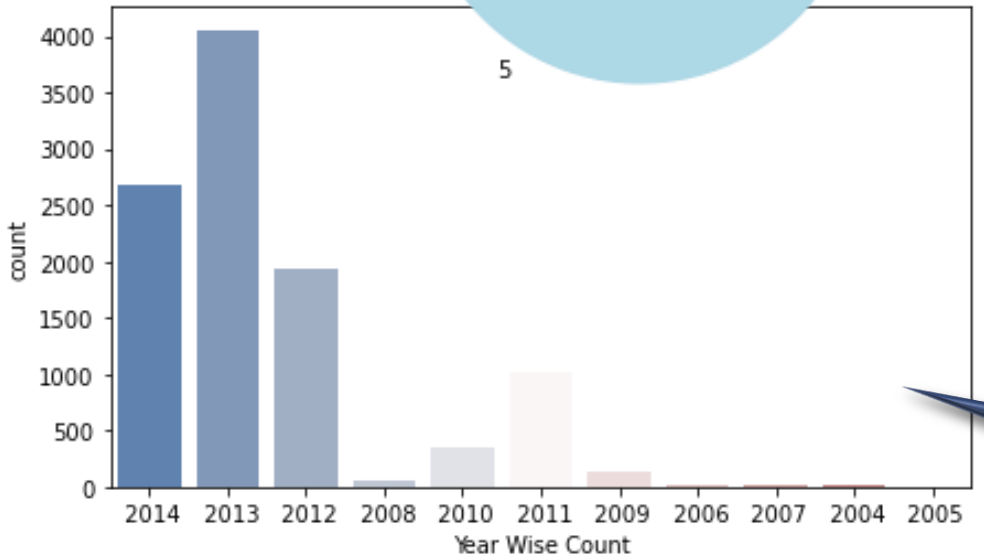
ANALYTICAL METHODOLOGY

Understand the dependent variable distribution in data set



This fig below helps us understand the relation between helpful column and the rating column , determining that more the good reviews can be decided based on higher helpful rate





The figure help us understand the rating count started to rise with the years

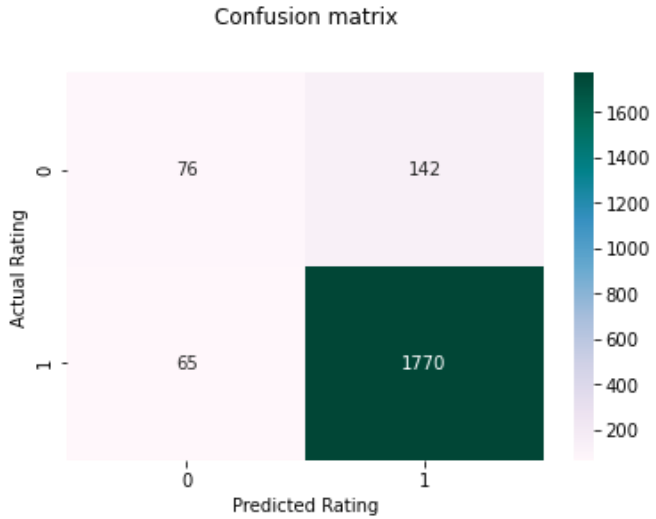
BUILDING ML MODEL

- I have implemented 2 models for the given dataset , so as to compare the final resultset and choose the best of them for further predictions.
  - Model 1 : Logistic Regression Model
  - Model 2 : ANN Classifier Model
- While training these models, various factors were considered and the model have been tunned with different hyperparameters so as to choose the best hyperparameter and get the maximum possible accuracy
- Different Model Evaluation metrics are used like Confusion Metrix , ROC-AUC Curve , Accuracy Score, Precision Score , Recall Score and F1 Score.

MODEL EVALUATIONS

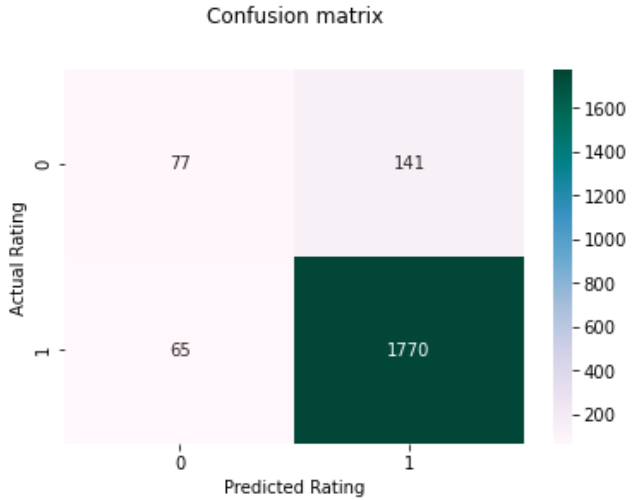
Model 1 : Logistic Regression Model

```
Accuracy Score : 0.899171943497321
Precision Score : 0.9257322175732218
Recall Score : 0.9645776566757494
F1 Score : 0.944755804643715
Confusion Matrix :
[[  76  142]
 [  65 1770]]
```



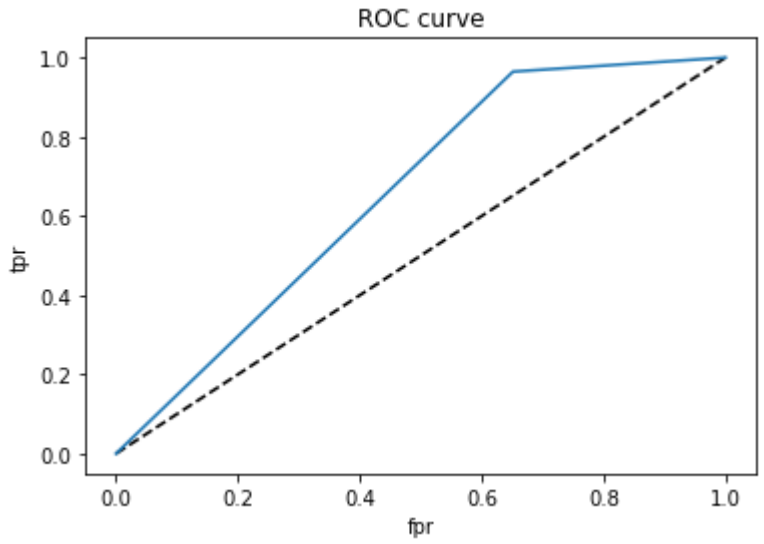
Model 2 :ANN Classifier Model

```
Accuracy Score : 0.8996590355577204
Precision Score : 0.9262166405023547
Recall Score : 0.9645776566757494
F1 Score : 0.9450080085424454
Confusion Matrix :
[[  77  141]
 [  65 1770]]
```



MODEL EVALUATIONS

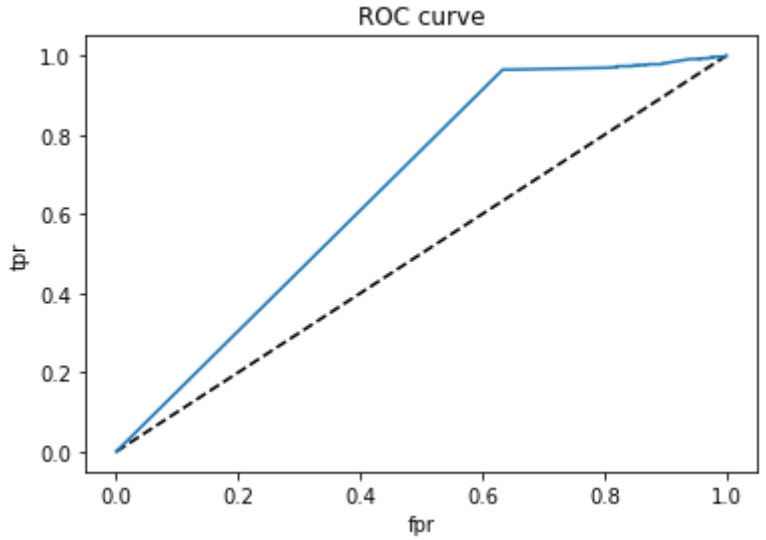
Model 1 : Logistic Regression Model




```
#Area under ROC curve
from sklearn.metrics import roc_auc_score
print('Area under ROC curve',roc_auc_score(y_test,y_pred))
```

```
Area under ROC curve 0.6566007549433792
```

Model 2 :ANN Classifier Model




```
#Area under ROC curve
from sklearn.metrics import roc_auc_score
print("Area under ROC curve", roc_auc_score(y_test,y_pred_proba))
```

```
Area under ROC curve 0.6631215158863085
```

PROPOSED NEXT STEPS

Data Set

- This Model prepared is on textual data that is provided in the data set, we can include more data like reviewer demographics data (Age, Gender, Occupation) as well as other information like Purchased Products number of visits to the site etc…

EDA- Extract Data Analysis

- Data transformations or more NLP techniques like TFIDF can be used
- More Visualization's around the dataset can be included

Modeling & Evaluation

- In the project is build with 2 ML model , more models can be implemented and comparison can be done with them

Deploying ML Model

- Building a pipeline framework that can be deployed in a production environment for continuous training and testing.For eg. MLFlow can be used to deploy these models , where maintenance becomes easy job

- Sukruti Admuthe