DATA SCIENCE – ASSIGNMENT 3

0.1 Problem Statement

A worldwide hotel brand wants to comprehend the customer feedback they have collected. The feedback comprises textual reviews from different sources, such as online websites, surveys, and social media. They want to find out how happy or unhappy the customers are with the hotel services. We require you to examine the feedback and evaluate the customer satisfaction level. Design and code a solution using Python and machine learning. Please provide well-documented code, a model if any, and clear instructions to run the code.

SOLUTION

Our process will start with Exploratory Data Analysis, followed by Model Development employing both Machine Learning and Deep Learning methodologies.

1 EXPLORATORY DATA ANALYSIS

Now I have the customer feedback data, It's time to perform Exploratory Data Analysis (EDA) to gain insights and understand the data before further proceeding.

1.1 Dataset

yelp_ratingsċsv is customer feedback data having columns text,stars and sentiment.

1.2 Observations

- The dataset contains a total of 44,530 records.
- No null values are present within the dataset.
- The dataset comprises three columns: ”text,” ”stars,” and ”sentiment.”
- The ”stars” column encompasses four categories: 1.0, 2.0, 4.0, and 5.0
- The ”sentiment” column consists of two categories: 0 and 1, where 0 signifies negative sentiment and 1 represents positive sentiment.
- Stars rated 1.0 and 2.0 are classified as negative sentiment (0), while stars rated 4.0 and 5.0 are categorized as positive sentiments.

1.3 Distribution

1.3.1 Distribution of stars

The following diagram shows the distribution of stars in the dataset.

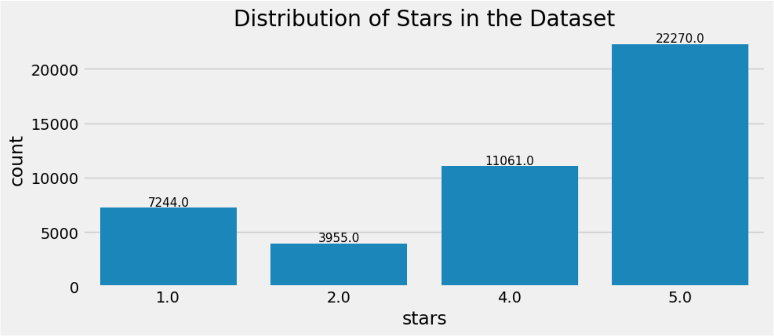


Figure 1. Distribution of Stars in the Dataset

1.3.2 Distribution of Sentiment

The following diagram shows the distribution of sentiment in the dataset. The dataset is imbalance as we can see from the below figure.

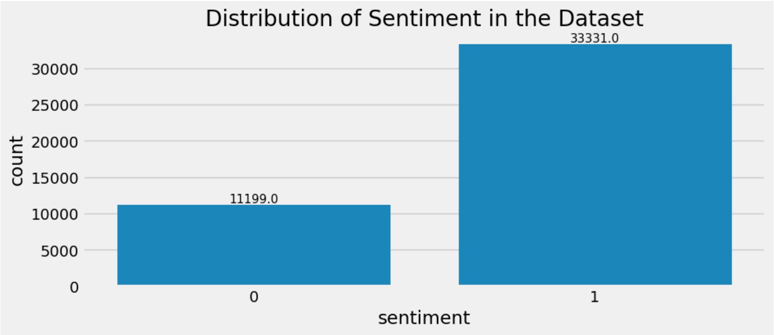


Figure 2. Distribution of Stars in the Dataset

2 DATA PREPORCESSING(FEATURE ENGINEERING) AND DATA CLEANING

2.1 Data Cleaning

The subsequent actions are taken in this step:

- Eliminating punctuation marks from the reviews.
- Removing Numbers from the reviews
- Removing accented characters from the reviews.
- Removing special characters from the reviews.
- Removing stop words and applying lemmatization

2.2 Data Preporcessing(Feature Engineering)

2.2.1 Text Subjectivity

In natural language, subjectivity refers to expression of opinions, evaluations, feelings, and speculations and thus incorporates sentiment. Subjective text is further classified with sentiment or polarity.We calculate the subjectvity of the Reviews by using TextBlob.

2.2.2 Most Frequently Occuring Words - Top 20

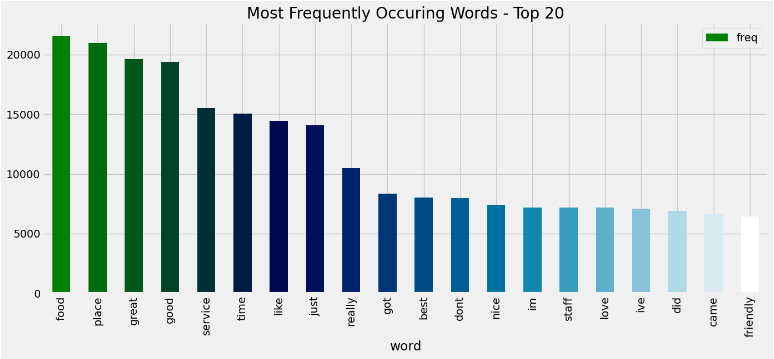


Figure 3. Most Frequently Occuring Words - Top 20

3 APPROACH

I prepare the training data from EDA step and the train data is used to develop the model.Initially, we adhere to the traditional Machine learning algorithmic approach, and subsequently, we employ deep learning techniques.

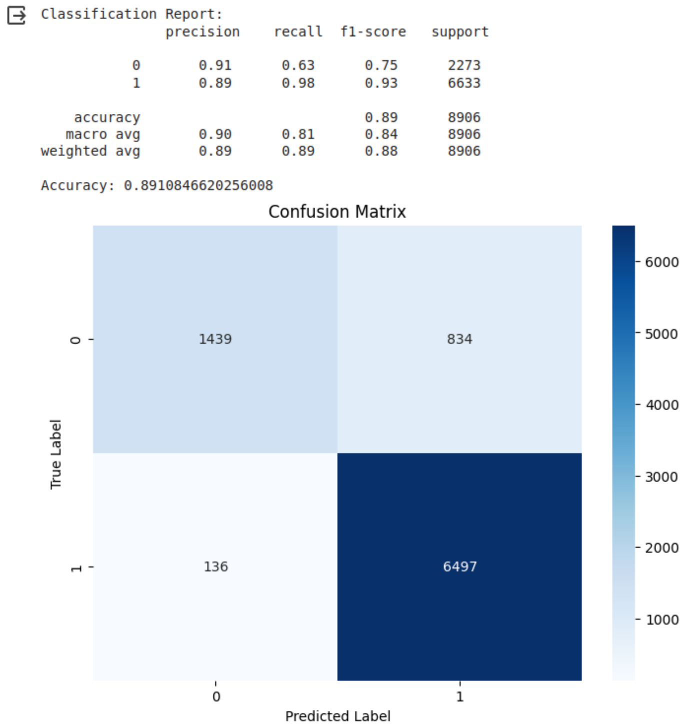
3.1 Machine Learning Approach

In this approach , we used three algorithms:

- Naive Bayes Classifier
- Randomforest Classifier
- Xgboost Classifier

3.1.1 Naive Bayes Classifier

Result & Metrics: The following diagram shows the confusion metrics for Naive Bayes algorithm



```
Classification Report:
               precision    recall  f1-score   support

           0       0.91      0.63      0.75      2273
           1       0.89      0.98      0.93      6633

    accuracy                           0.89      8906
   macro avg       0.90      0.81      0.84      8906
weighted avg       0.89      0.89      0.88      8906

Accuracy: 0.8910846620256008
```

Figure 4. Classification Report by using -Naive Bayes

Metrics Explanation :
Term Explanation

- **Precision**: The proportion of correctly identified instances among those labeled as positive. It measures how often the model is correct when it predicts a positive result.
- **Recall**: The proportion of actual positive instances that were correctly identified. It measures how often the model correctly identifies all positive cases.
- **F1-score:** The harmonic mean of precision and recall, providing a balanced measure of accuracy that considers both. Higher F1-scores indicate better overall performance.
- **Support**: The number of instances for each class in the dataset. It shows how much data was available for training and evaluation.

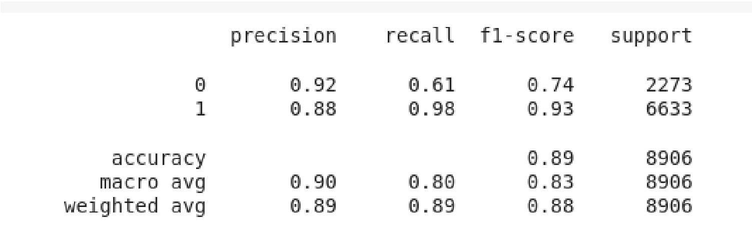
Overall Performance:

- **Accuracy**: 0.89(how often the model makes correct predictions)
- **Precision:** How many of the model's positive predictions were actually correct.
  - Precision for positive sentiment: 0.91 (out of 2273 positive sentiment predictions, 91% were correct)

- Precision for negative sentiment: 0.89 (out of 6633 negative sentiment predictions, 89% were correct)

- **Recall**: How many of the actual positive sentiment cases did the model predict correctly.
  - Recall for positive sentiment: 0.63 (out of 2273 actual positive sentiment cases, the model predicted 63% correctly)
  - Recall for negative sentiment: 0.98 (out of 6633 actual negative sentiment cases, the model predicted 98% correctly)
- **F1-Score**:Harmonic mean between precision and recall.
  - F1-score for positive sentiment: 0.75
  - F1-score for negative sentiment: 0.93

3.1.2 Radomforest Classifier



```
              precision    recall  f1-score   support

           0       0.92      0.61      0.74      2273
           1       0.88      0.98      0.93      6633

    accuracy                           0.89      8906
   macro avg       0.90      0.80      0.83      8906
weighted avg       0.89      0.89      0.88      8906
```

Figure 5. Classification Report by using -Randomforest Classifier

Class-Specific Metrics :

- **Accuracy:** 0.89 (how often the model makes correct predictions)
- **Precision:**How many of the model's positive predictions were actually correct.
  - Precision for positive sentiment: 0.91 (out of 2273 positive sentiment predictions, 91% were correct)
  - Precision for negative sentiment: 0.89 (out of 6633 negative sentiment predictions, 89% were correct)
- **Recall:**How many of the actual positive sentiment cases did the model predict correctly.
  - Recall for positive sentiment: 0.63 (out of 2273 actual positive sentiment cases, the model predicted 63% correctly)
  - Recall for negative sentiment: 0.98 (out of 6633 actual negative sentiment cases, the model predicted 98% correctly)
- **F1-Score**:Harmonic mean between precision and recall.
  - F1-score for positive sentiment: 0.75
  - F1-score for negative sentiment: 0.93

3.1.3 Xgboost Classifier

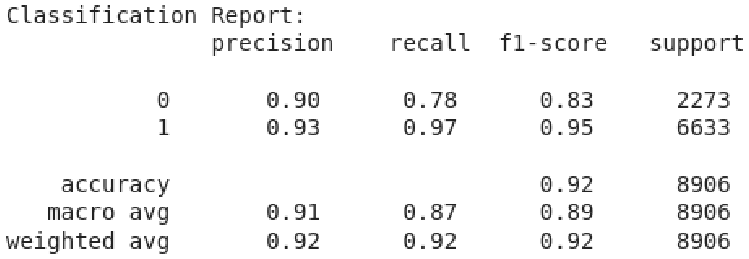
Class-Specific Metrics :

- **Accuracy:** 0.92 (how often the model makes correct predictions)
- **Precision:**How many of the model's positive predictions were actually correct.
  - Precision for positive sentiment: 0.93 (out of 2273 positive sentiment predictions, 93% were correct)

  - Precision for negative sentiment: 0.90 (out of 6633 negative sentiment predictions, 90% were correct)

- **Recall:**How many of the actual positive sentiment cases did the model predict correctly.
  - Recall for positive sentiment: 0.97 (out of 2273 actual positive sentiment cases, the model predicted 97% correctly)
  - Recall for negative sentiment: 0.78 (out of 6633 actual negative sentiment cases, the model predicted 78% correctly)

- **F1-Score**:Harmonic mean between precision and recall.
  - F1-score for positive sentiment: 0.95
  - F1-score for negative sentiment: 0.83



```
Classification Report:
               precision    recall  f1-score   support

           0       0.90      0.78      0.83      2273
           1       0.93      0.97      0.95      6633

    accuracy                           0.92      8906
   macro avg       0.91      0.87      0.89      8906
weighted avg       0.92      0.92      0.92      8906
```

Figure 6. Classification Report by using -Xgboost Classifier

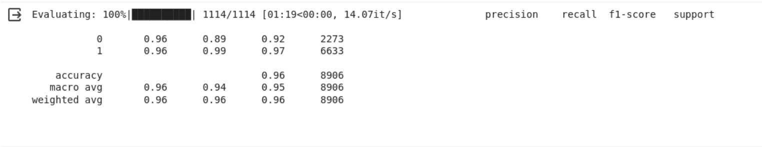
3.2 Deep Learning Approach

In deep learning approach, I experimented with a transformer-based model called BERT-base-uncased to assess its performance.

3.2.1 Implementation

I employed the PyTorch framework to create a custom class, utilizing the DataLoader from PyTorch. Additionally, a pre-trained model from Hugging Face was invoked.

Metrics Explanation:



```
Evaluating: 100%|██████████| 1114/1114 [01:19<00:00, 14.07it/s]              precision    recall  f1-score   support

           0       0.96      0.89      0.92      2273
           1       0.96      0.99      0.97      6633

    accuracy                           0.96      8906
   macro avg       0.96      0.94      0.95      8906
weighted avg       0.96      0.96      0.96      8906
```

Figure 7. Classification Report by using -BERT

Class-Specific Metrics::

- **Accuracy:** 0.96 (how often the model makes correct predictions)

- **Precision:**How many of the model's positive predictions were actually correct.
  - Precision for positive sentiment: 0.96 (out of 2273 positive sentiment predictions, 96% were correct)
  - Precision for negative sentiment: 0.96 (out of 6633 negative sentiment predictions, 96% were correct)

- **Recall:**How many of the actual positive sentiment cases did the model predict correctly.

  - Recall for positive sentiment: 0.99 (out of 2273 actual positive sentiment cases, the model predicted 99% correctly)
  - Recall for negative sentiment: 0.89 (out of 6633 actual negative sentiment cases, the model predicted 89% correctly)

- **F1-Score**:Harmonic mean between precision and recall.
  - F1-score for positive sentiment: 0.97
  - F1-score for negative sentiment: 0.92

Obervation :

- BERT shows promise with higher over all accuracy, precision and F1-scores

4 CODE

- **EDA.ipynb**: Please refer this notebook for Exploratory Data Analysis and data preparation step.After executing this code, we will get the data for training the model.The data name will be :train_data.csv
- **ML_approachipynb**: Please refer this notebook for Machine Learning Approach.
- **DL_approachipynb**: Please refer this notebook for deep Learning Approach.

MY PORTFOLIO

Please feel free to explore my projects:

- Github