International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 05 | May 2021 www.irjet.net p-ISSN: 2395-0072

Sentiment Analysis Using Supervised Machine Learning Techniques

Harinath Yadav[1], Ujjawal Kumar[2], Pramit Karmakar[3], G. Paavai Anand[4]

[1-4]Computer Science and Engineering Department, SRM Institute of Science and Technology Vadapalani, Chennai

---

Abstract - Google's official app store, known as Google Play Store is used by millions of people daily. It can be used to access the contents which include magazines, games, music, movies, television, etc. Users are allowed to rate the apps present in the app store after they download it to share their personal experience of the app, and this works both ways where another user is inspired from the ratings by other users. These experiences given by the users usually describe the usability of the app, its performance, and quite often the problems which one has faced while using it. The aim is to classify the google app reviews based on supervised machine learning techniques (SMLT). The SMLT techniques are used to collect variable identification like sentiments, sentiment polarity, etc. and for building a dataset using these variables, which will then go through the several processes such as data validation/cleaning, visualisation and finally it'll be used to classify the sentiments as positive, neutral and negative.

I. Introduction

Domain overview

Machine learning is a technique which predicts a specific set of events or data and also allows computers to learn without having to be hard coded or specifically programmed. Machine learning is concerned with the building of programmes that can adapt to latest data, as well as the fundamentals of ML, such as the implementation of ML algorithms in Python. Specialized algorithms are used in the training and prediction process. It feeds the training data to an algorithm, which uses it to make predictions about the current test data. ML can be divided into these distinct groups: supervised, unsupervised, and reinforcement learning. A supervised learning algorithm is given both the input data and the accompanying labelling to learn data, which must first be labelled by a person. There are no labels of unsupervised learning. It was made available to the learning algorithm. This algorithm must determine how the input data is clustered. Finally, reinforcement learning communicates with its surroundings in a complex manner and provides constructive or negative feedback in order to enhance output.

To find trends in Python that contribute to actionable observations, data scientists use a variety of machine learning algorithms. These algorithms can be divided into two categories depending on how they "read" about data in order to predict: supervised and unsupervised learning. The process of predicting the class of given data points is known as classification.

Objectives

The aim is to create a machine learning model for classifying Google Play Store feedback that could eventually replace updatable supervised machine learning classification techniques by forecasting outcomes at best accuracy by comparing supervised algorithms.

Problem Description/ Problem Statements

As people experience it, mobile applications play an important role in our everyday lives. In today's world, everybody uses Android apps for various purposes such as messaging, food distribution, exercise, and themes. This internet market offers smartphone users free and paying access to over a million web applications, also known as android apps. Users can select from over a million android applications for different mobile devices on the Google Play store website, but bear in mind that some of the downloaded apps are not useful enough. The app store allows users to browse for, purchase, and install software apps with just a few clicks. This website enables users to post their feedback. App consistency can be improved with the aid of reviews.

II. Proposed System

Exploratory Data Analysis

Machine learning supervised classification algorithms will be used to give the given dataset and extract patterns, which would help in classifying the reviews, thereby helping the apps for making better decisions of their features in the future.

Data Wrangling

We load the files, check for cleanliness and trimming and cleaning is done in the dataset for review in this phase of the report.

Data collection

The data collection gathered for classifying the provided data is divided into two parts: training and testing. In most cases, a 70:30 division is used to divide the dataset into train and test sets. SMLT-created Data Model is then put in to the train set and test set prediction is carried out based on the test outcome accuracy.

The reports are investigated which shows the applicability of ML to segregate the reviews.

Project Goals

- Exploring data analysis of variable identification
  - The given dataset is loaded.
  - Required library packages are imported.
  - Analysing the general properties
  - Finding the missing and duplicate values.
  - Check for count and unique values
- Uni-variate data analysis
  - Rename, add and drop data
  - To specify data type
- Exploration data analysis of bi-variate and multi-variate
  - Plot diagram of pair plot, heat map, bar chart and Histogram
- Method of Outlier detection with feature engineering
  - Pre-processing the given dataset
  - Splitting the training and test data
  - Differentiating the Decision tree and Logistic regression model and random forest etc.
- Comparing the algorithm to predict result
  - Based on best accuracy

IV. MODULES DESCRIPTION

List of Modules:

- Data validation Process EDA
- Exploration data analysis of visualization
- Compare the Algorithm with prediction to get best accuracy result
- Deep Learning RNN with LSTM get best accuracy result
- Output for prediction of sentiment by giving input sentences.

Variable Identification Process/ Data validation process

Machine learning validation methods are utilised for calculating the error rate of the ML algorithm, which is as identical as possible to the actual error rate of the sample. Duplicate the value and the data form definition to locate the missing value, whether it is a float variable or an integer variable. When tuning model hyper parameters, a sample data is used to provide a neutral estimation on the model to fit on the training dataset.

The validation collection is used to validate a model, but it is only used on a regular basis. This data was used by machine learning engineers to regulate the model hyper parameters. Data processing, interpretation, and the exercise of addressing data information, accuracy, and structure can be time-consuming. Understanding the data and its properties is helpful during the data recognition process; this information can help you select which algorithm to prefer to construct your model. For example, to display the data type format of a dataset. Various data cleaning tasks use Python's Pandas library, with an emphasis on the most common data cleaning job, missing values, and the ability to clean data more easily. It prefers to spend more time in analysing and modelling the data and less time in cleaning it. Any of these sources are simply unintentional errors. Other times, there could be a more serious explanation for the lack of results. From a statistical standpoint, it's critical to comprehend the various forms of lost evidence. The category of missing data will affect how missing values are filled in, how missing values are detected, and how simple imputation and precise mathematical approaches are used to deal with missing data. It's crucial to consider the origins of lost data before putting it into code.

Exploration of data analysis and visualisation

In applied analytics and machine learning, data visualisation is a crucial capability. Statistics is concerned with objective data explanations and estimations. Data visualisation is a valuable collection of methods for obtaining a qualitative interpretation of data. This can be useful for detecting trends, corrupt results, outliers, and other things while researching and getting to know a dataset. Data visualisations should be used to articulate and illustrate core associations in graphs and maps that are more visceral and meaningful to clients than indicators of affiliation or importance with a little domain information.

Data can't make sense until it is shown in a graphic format, such as charts and graphs. The ability to image data samples and other objects easily is a valuable skill in both applied statistics and applied machine learning. It will show you how to use the various types of plots available when visualising data in Python to better understand your own data.

Some of the Data Visualisation techniques used are:

Outlier detection process

Outliers in input data can deform and deceive the ML programs training processes, resulting in greater training time and somewhat inaccurate models resulting in bad performance. Earlier predictive models are based on training data, outliers that can trigger misleading representations and, as a result, mislead perceptions of collected data. Outliers will compact the body of data by

biasing the summary distribution of attribute values in predictive statistics like mean and standard deviation, as well as in plots like histograms and scatter plots. Finally, in the case of fraud detection and information protection, outliers may indicate cases of data samples that are significant to the issue, such as anomalies. It was not able to match the model on the training data, and it is thus impossible to predict how well the model would do on real data for which, we make sure that our model extracts the correct patterns from the data and that isn't generating excessive sound. Before proceeding to the actual dataset, we cross-validate it by training our module on a subset of the dataset, which is known as Cross-validation. The transformation of data before any actual application of algorithm on it is knows as pre-processing. After pre-processing, raw data are converted into clean data. In other words, once data is obtained from various sources, it is processed in raw format, which makes interpretation impossible. To get better outcomes from the implemented model of Machine Learning, the data must be organised properly. Any ML models contains data in a certain format, which does not accept any null values. As a result, all the null values are handled from the original raw data collection in order to run the random forest algorithm.

False Positives (FP): An individual who is expected to pay is referred to as a defaulter. When the expected class is yes and the real class is no. For e.g. though the expected class indicates that the passenger would survive, the real class indicates the passenger didn't.

False Negatives (FN): An individual who defaults is likely to be a payer. When the real class is positive but the expected class is negative. For example, if the passenger's real class value implies that he or she survived, but the expected class value indicates that the passenger will die.

True Positives (TP): Defaulter is a term used to describe someone who does not pay their bills. Both accurately predict positive values, displaying the value of the real class is yes, as well as the value of the predicted class. For instance, if both the real and expected class value indicate that this passenger survived.

True Negatives (TN): An individual who defaults is likely to be a payer. Both are accurately estimated negative numbers, indicating that the value of the real class is zero and the value of the predicted class is zero as well. For example, if both the real and the expected class says the passenger did not survive.

Comparing Algorithms: The below 5 different algorithms are compared:

- Decision tree
- Random forest
- Support Vector Machines
- K-Nearest Neighbours
- Logistic Regression

The K-fold cross validation process is used to evaluate every algorithm, which ensures that the same splits to the training data are performed and also checks that each algorithm is being evaluated in the same way. The test and train sets should also be separated. By comparing precision, it is possible to forecast the outcome.

Predicting result by accuracy:

For predicting a value in the logistic regression algorithm, a linear equation with independent predictors is often used. The estimated value ranges from negative infinity to positive infinity. Higher accuracy prediction result is the logistic regression model by comparing the best accuracy.

True Positive Rate (TPR) = TP / (TP + FN)

False Positive rate (FPR) = FP / (FP + TN)

Accuracy calculation:

Accuracy = (TP + TN) / (TP + TN + FP + FN)

The most intuitive efficiency metric is accuracy, which is essentially the number of correctly expected observations to all observations. One might believe that if our model is accurate, it is the best. Precision metric is useful only where the datasets are symmetric and both the values of false positives and negatives are almost equal.

Precision:

Precision = TP / (TP + FP)

Recall:

Recall = TP / (TP + FN)

General Formula:

F- Measure = 2TP / (2TP + FP + FN)

F1-Score Formula:

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

IV. ALGORITHM & TECHNIQUES

Algorithm Explanation

In machine learning and statistics, classification is a supervised learning method in which a computer programme learns from data input and then uses its learning to classify new findings. This data set can be bi-class or multi-class. Some examples include handwriting and speech recognition, biometric identification and other classification problems. Algorithms in Supervised Learning learn from labelled results. The algorithm decides the mark should be assigned to new data based on pattern and associating the patterns to the unlabelled new data after knowing the data.

Logistic Regression

It's a mathematical technique for evaluating a data set in which one or more independent variables influence the result. A dichotomous variable is used to assess the result. The aim of logistic regression is to find the model that better describes the relationship between a set of independent variables and a dichotomous characteristic of interest.

Decision Tree

It is a very efficient and well-known algorithm. The decision-tree algorithm is classified as a supervised learning algorithm. It can be used for both continuous and categorical output variables. Decision tree assumptions: At first, we assume the whole training set to be the root. For knowledge benefit, attributes are assumed to be categorical, but attributes are assumed to be continuous. Records are allocated recursively based on attribute values.

It incrementally breaks down a data set into smaller subsets while simultaneously developing an associated decision tree. A leaf node represents a grouping or judgement, and a decision node has two or three divisions. The top node, also known as root node corresponds to the best predictor. It uses a mutually exclusive and exhaustive if-then rule set for grouping. The rules are learned one by one, based on the outcomes of the preparation. When a rule is taught, the tuples that are covered by the rules are removed. This technique is repeated before the training set's termination condition is met, and it's built in a recursive divide-and-conquer format from the top down.

***K*-Nearest Neighbour (KNN)**

It's a supervised machine learning algorithm that keeps track of all instances in n-dimensional space that match training data points. When an undefined distinct data is obtained, it analyses nearest k number of saved instances and returns the most frequent class as the estimate, while real-valued data returns an average of the k nearest neighbors. It accepts a collection of marked points and uses them to teach itself how to mark more. To mark a new point, it surveys at the named points nearest to it and polls its neighbors; the label with the most votes becomes the new point's label (the "k" refers to the number of neighbors it polls). Using the whole training package, it makes assumptions about the validity set.

Random Forest

It's a supervised algorithm for classification or regression and can also train a large number of decision trees and give an output class that is an average of individual trees. This algorithm is based on ensemble learning system. Ensemble learning is a process of combining various versions of same or similar algorithms to devise a more efficient predictive model.

Support Vector Machines

It's a classifier which categorises a data set by determining the best hyperplane among the data points. This classifier was chosen because it's highly flexible in terms of the amt. of different kernelling functions which can be used, and it can also provide a large predictability score. They became immensely successful when they were first developed in the 1990s, and they remain the go-to tool for a high-performing algorithm that requires minimal tuning today.

Output Screenshots:

Information

```
In [11]: #Checking datatype and information about dataset
         df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 37427 entries, 0 to 64230
Data columns (total 5 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   App                    37427 non-null  object
 1   Translated_Review      37427 non-null  object
 2   Sentiment              37427 non-null  object
 3   Sentiment_Polarity     37427 non-null  float64
 4   Sentiment_Subjectivity 37427 non-null  float64
dtypes: float64(2), object(3)
memory usage: 1.7+ MB
```

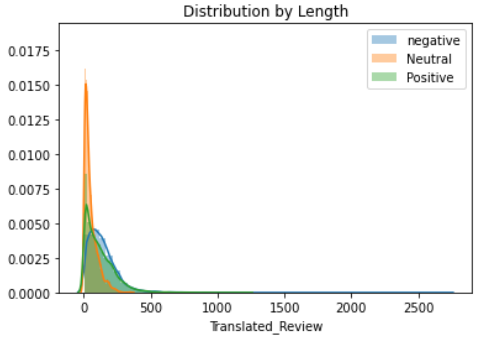
Checking duplicate values of dataframe:

Missing values

```
In [14]: #Checking sum of missing values
         df.isnull().sum()

Out[14]: App                    0
         Translated_Review      0
         Sentiment              0
         Sentiment_Polarity     0
         Sentiment_Subjectivity 0
         dtype: int64
```

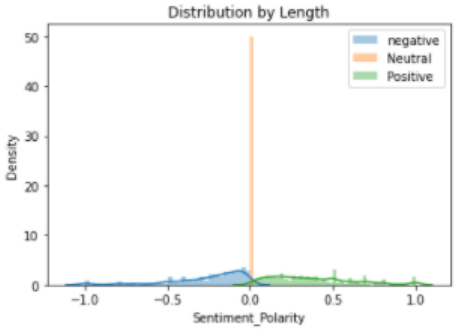
Density plot

```
Out[11]: <matplotlib.legend.Legend at 0x255dcf5ad60>
```



Sentiment Polarity Plot

```
Out[43]: <matplotlib.legend.Legend at 0x18126e7e460>
```



Visualizing the reviews

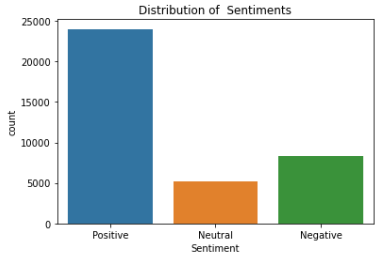
```
plt.title('Sentiment:Pasitive')
Out[21]: Text(0.5, 1.0, 'Sentiment:Pasitive')
```

Build Neural network Summary

```
72]: # constructs the model with 128 LSTM units
model = get_model(tokenizer=tokenizer, lstm_units=128)

Reading GloVe: 400000it [03:25, 1941.77it/s]

Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, 100, 100)          2208500
_________________________________________________________________
lstm_2 (LSTM)                (None, 128)               117248
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 3)                 387
=================================================================
Total params: 2,326,135
Trainable params: 117,635
Non-trainable params: 2,208,500
_________________________________________________________________
```

Visualize Sentiment types on bar plot

```
In [20]: #plotting graph for distribution
import matplotlib.pyplot as plt
import seaborn as sns
sns.countplot(x = "Sentiment", data = df)
df.loc[:, 'Sentiment'].value_counts()
plt.title('Distribution of  Sentiments')

Out[20]: Text(0.5, 1.0, 'Distribution of  Sentiments')
```



KNN Accuracy

```
Accuracy result of K-Nearest Neighbor is: 98.95805503606732


Classification report of K-Nearest Neighbor Results:

              precision    recall  f1-score   support

           0       0.99      0.97      0.98      2481
           1       0.99      1.00      0.99      8748

    accuracy                           0.99     11229
   macro avg       0.99      0.98      0.98     11229
weighted avg       0.99      0.99      0.99     11229

Confusion Matrix result of K-Nearest Neighbor is:
 [[2396   85]
 [  32 8716]]

Sensitivity :  0.9657396211205159

Specificity :  0.9963420210333791
```

Naive Bayes Accuracy

```
Accuracy result of Naive bayes is: 97.80924392198771


Classification report of Naive bayes: Results:

              precision    recall  f1-score   support

           0       1.00      0.90      0.95      2481
           1       0.97      1.00      0.99      8748

    accuracy                           0.98     11229
   macro avg       0.99      0.95      0.97     11229
weighted avg       0.98      0.98      0.98     11229

Confusion Matrix result of Naive bayes: is:
 [[2235  246]
 [   0 8748]]

Sensitivity :  0.9008464328899637

Specificity :  1.0
```

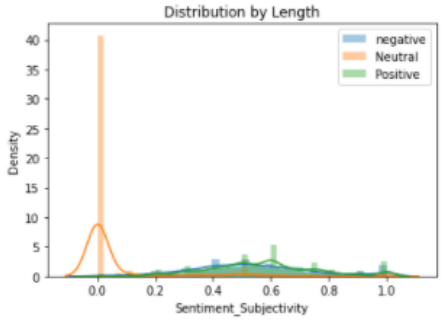
Comparing algorithms

Giving input sentence getting sentiment output:

```
In [66]: text=str(input("enter the statement: "))

         enter the statement: This help eating healthy exercise regular basis

In [67]: #text = "We stayed for a one night getaway with family on a thursday. Triple AAA "
         print(get_predictions(text))

         Positive

In [68]: text = "Language barrier. I understand Korea. I want English."
         print(get_predictions(text))

         Neutral
```

Sentiment Subjectivity Plot

```
Out[44]: <matplotlib.legend.Legend at 0x181290c1d00>
```



REFERENCES

[1] Chi-squared test of independence. http://www.rtutor.

[2] Safwat Hassan, Cor-Paul Bezemer, and Ahmed E. Hassan, “Studying Bad Updates Of Top Free-To-Download Apps in the Google Play Store”

[3] Pan Li, and Alexander Tuzhilin, “Learning Latent Multi-Criteria Ratings from User Reviews for Recommendations”, journal of latex class files, vol. 14, no. 8, December 2019.

[4] T. Donkers, B. Loepp, and J. Ziegler, “Explaining recommendations by means of user reviews,” in Proc. 1st Workshop Explainable Smart Syst.

(ExSS), 2018. [Online]. Available: http://ceur-ws.org/Vol-2068/exss8.pdf

[5] K. Moran, B. Li, C. Bernal-C´ardenas, D. Jelf, and D. Poshyvanyk. Automated reporting of GUI design violations for mobile apps. In Proceedings of the 40th International Conference on Software Engineering, ICSE ’18, 2018.