MACHINE LEARNING
(CSC8111)

TASK – FARS, TWITTER SENTIMENT ANALYSIS

NAME: KAILASH BALACHANDIRAN
STUDENT ID: 220243160

FARS

1. Introduction

The objective is to evaluate and forecast the "FARS" and "Twitter sentiment datasets" using machine learning techniques. The research is conducted using a machine learning pipeline structure that starts with data pre-processing and moves on to data cleaning, training, and prediction. A variety of machine learning models are fitted to the dataset, and the performance of each model is examined to select the most successful model with noticeably higher accuracy.

The FARS dataset is a census of fatal traffic accidents that happened inside the boundaries of all 50 states, the District of Columbia, and Puerto Rico and that resulted in at least one person's death (either a driver or a passenger) within 30 days after the accident. Several classifiers are used to predict the classes of a certain record in the FARS dataset to compare the performance of the models. The tweets in the Twitter sentiment dataset can be classified as positive, negative, or neutral depending on the content of the message. FARS dataset.

2. Exploratory Data Analysis

The target variable, which is likewise encoded as factors, is found in the last column of the dataset's 30 rows. There are 8 classes in the target variable.

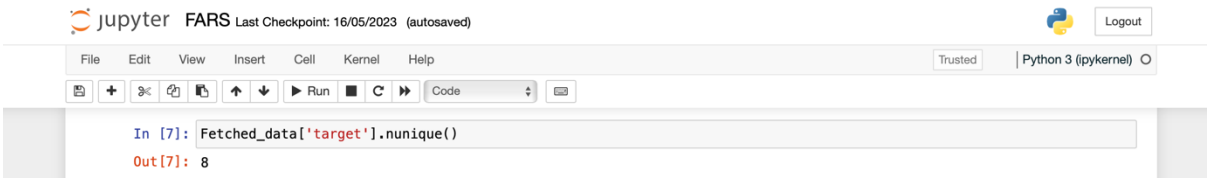


Figure 1

The pre-processed dataset's first five samples are listed below.

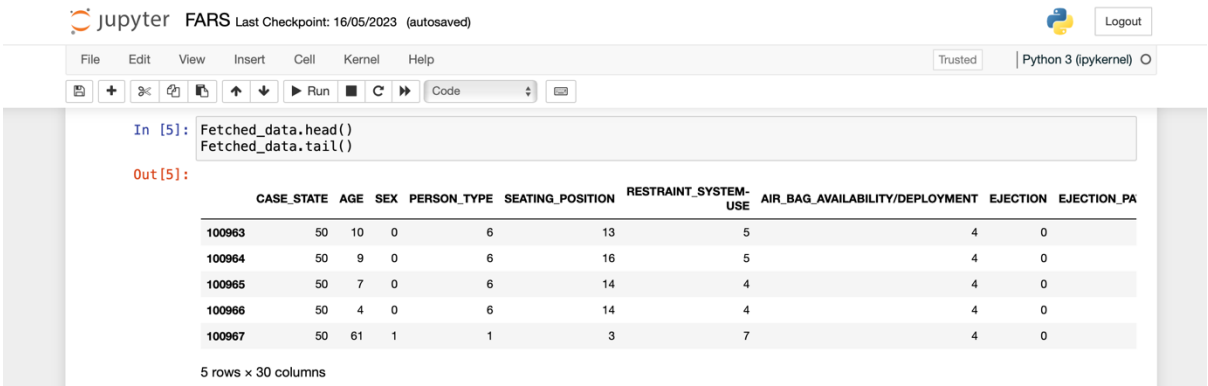


Figure 2

The data's values are as follows and don't contain any null values.

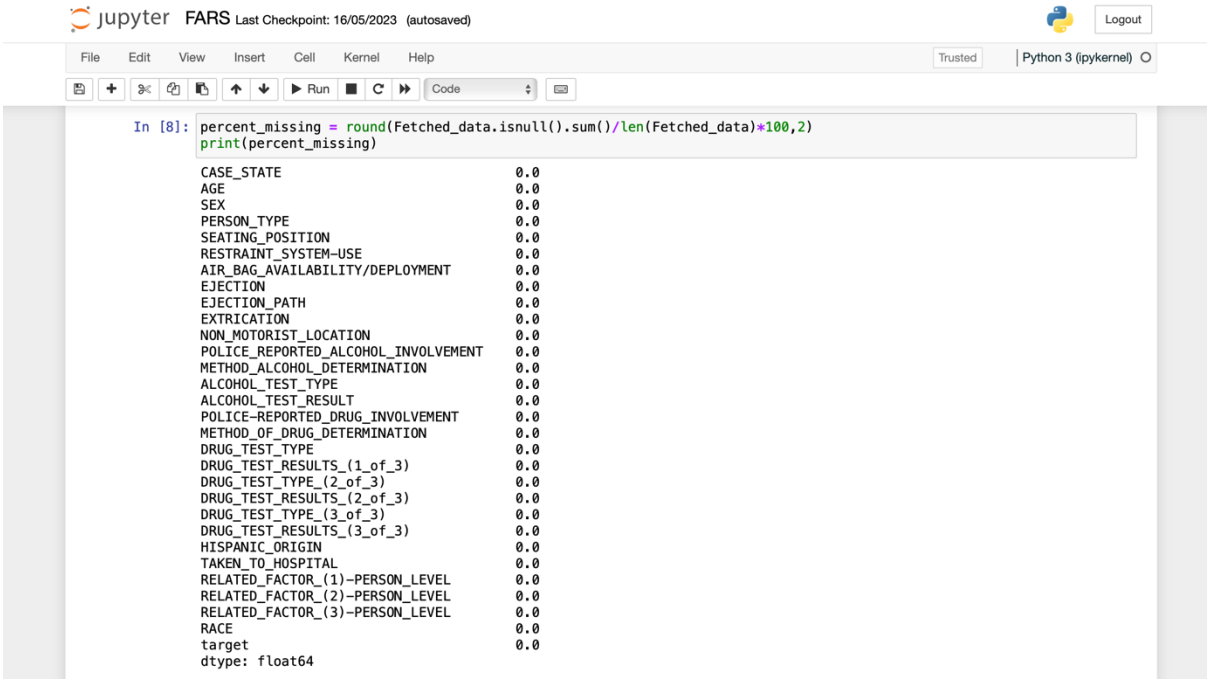


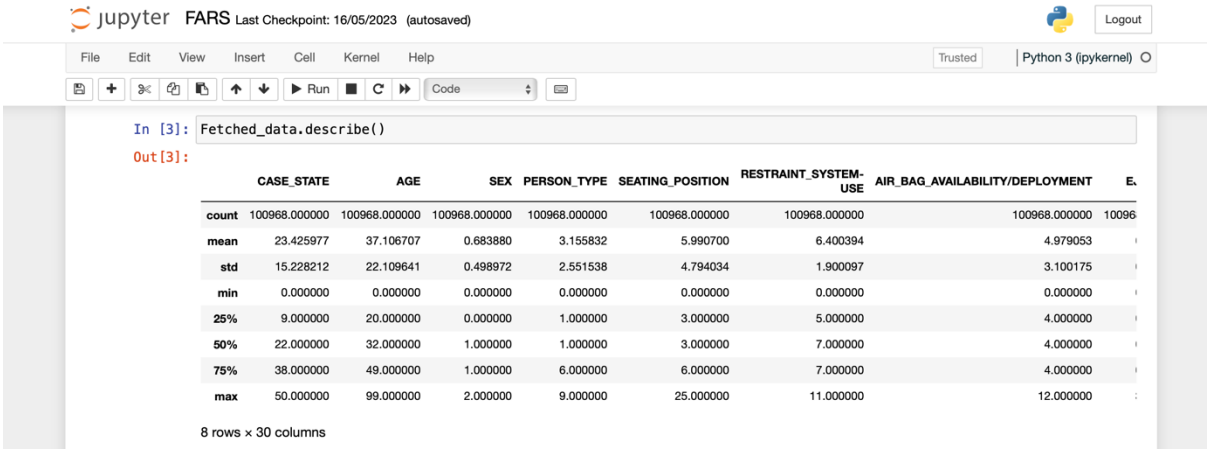
Figure 3

The data collection consists of 30 features and 100968 observations. 29 independent variables and 1 dependent variable make up the 30 characteristics.

Figure 4

3. Numerical data analysis

The screenshots below illustrate that the only discrete value is age and that there is little variation between the 75th percentile and the median. Outliers are therefore hard to come by. Most of the traits are hence symmetrically distributed with low skewness. The maximum age is 99, which is peculiar. Most of the other features in the data set are categorical, and there won't be any outliers. Each component's differences act as a reminder of the importance of homogeneity.



jupyter FARS Last Checkpoint: 16/05/2023 (autosaved) Logout

File Edit View Insert Cell Kernel Help Trusted Python 3 (ipykernel)

Run Code

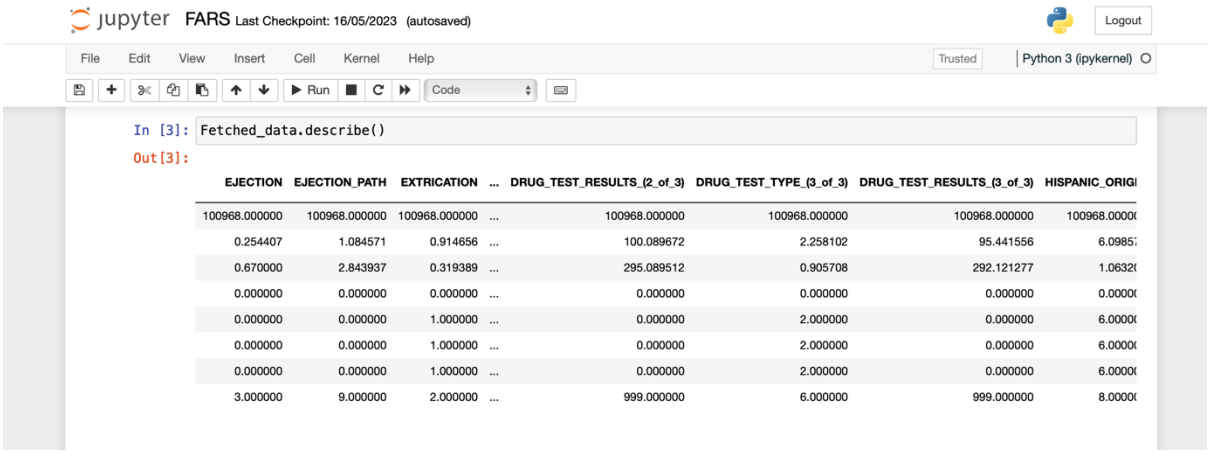
```
In [3]: Fetched_data.describe()
```

Out[3]:

	CASE_STATE	AGE	SEX	PERSON_TYPE	SEATING_POSITION	RESTRAINT_SYSTEM-USE	AIR_BAG_AVAILABILITY/DEPLOYMENT	E.
count	100968.000000	100968.000000	100968.000000	100968.000000	100968.000000	100968.000000	100968.000000	10096
mean	23.425977	37.106707	0.683880	3.155832	5.990700	6.400394	4.979053	
std	15.228212	22.109641	0.498972	2.551538	4.794034	1.900097	3.100175	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	9.000000	20.000000	0.000000	1.000000	3.000000	5.000000	4.000000	
50%	22.000000	32.000000	1.000000	1.000000	3.000000	7.000000	4.000000	
75%	38.000000	49.000000	1.000000	6.000000	6.000000	7.000000	4.000000	
max	50.000000	99.000000	2.000000	9.000000	25.000000	11.000000	12.000000	

8 rows × 30 columns

Figure 5



jupyter FARS Last Checkpoint: 16/05/2023 (autosaved) Logout

File Edit View Insert Cell Kernel Help Trusted Python 3 (ipykernel)

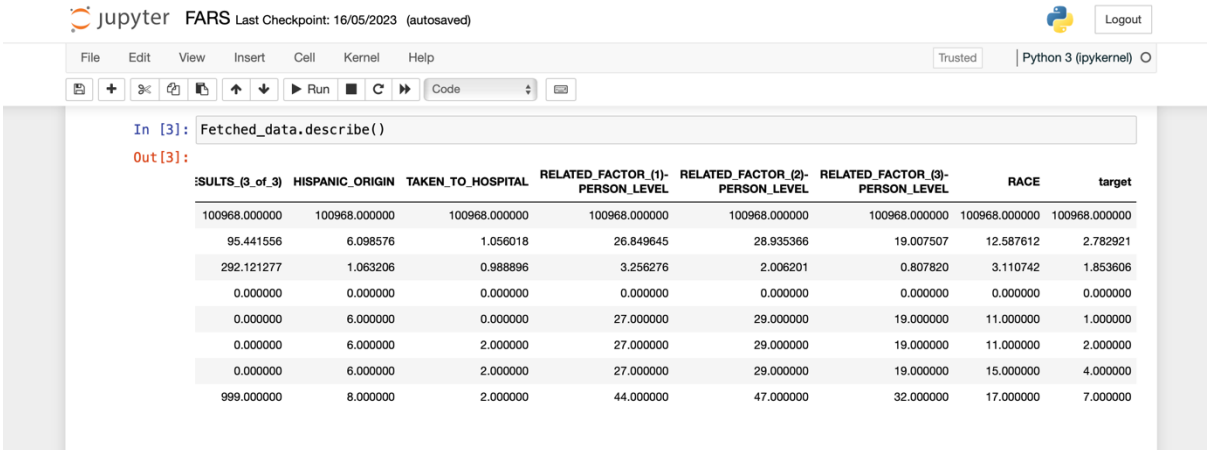
Run Code

```
In [3]: Fetched_data.describe()
```

Out[3]:

EJECTION	EJECTION_PATH	EXTRICATION	...	DRUG_TEST_RESULTS_(2_of_3)	DRUG_TEST_TYPE_(3_of_3)	DRUG_TEST_RESULTS_(3_of_3)	HISPANIC_ORIGI
100968.000000	100968.000000	100968.000000	...	100968.000000	100968.000000	100968.000000	100968.0000(
0.254407	1.084571	0.914656	...	100.089672	2.258102	95.441556	6.0985
0.670000	2.843937	0.319389	...	295.089512	0.905708	292.121277	1.0632(
0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.0000(
0.000000	0.000000	1.000000	...	0.000000	2.000000	0.000000	6.0000(
0.000000	0.000000	1.000000	...	0.000000	2.000000	0.000000	6.0000(
0.000000	0.000000	1.000000	...	0.000000	2.000000	0.000000	6.0000(
3.000000	9.000000	2.000000	...	999.000000	6.000000	999.000000	8.0000(

Figure 6



jupyter FARS Last Checkpoint: 16/05/2023 (autosaved) Logout

File Edit View Insert Cell Kernel Help Trusted Python 3 (ipykernel)

Run Code

```
In [3]: Fetched_data.describe()
```

Out[3]:

:SULTS_(3_of_3)	HISPANIC_ORIGIN	TAKEN_TO_HOSPITAL	RELATED_FACTOR_(1)-PERSON_LEVEL	RELATED_FACTOR_(2)-PERSON_LEVEL	RELATED_FACTOR_(3)-PERSON_LEVEL	RACE	target
100968.000000	100968.000000	100968.000000	100968.000000	100968.000000	100968.000000	100968.000000	100968.000000
95.441556	6.098576	1.056018	26.849645	28.935366	19.007507	12.587612	2.782921
292.121277	1.063206	0.988896	3.256276	2.006201	0.807820	3.110742	1.853606
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	6.000000	0.000000	27.000000	29.000000	19.000000	11.000000	1.000000
0.000000	6.000000	2.000000	27.000000	29.000000	19.000000	11.000000	2.000000
0.000000	6.000000	2.000000	27.000000	29.000000	19.000000	15.000000	4.000000
999.000000	8.000000	2.000000	44.000000	47.000000	32.000000	17.000000	7.000000

Figure 7

4. Data Pre-processing and cleaning

There are 28 discrete and category features in the data that was acquired. Pre-processing of the data has already taken place, including the transformation of categorical text data into categorical numeric data. Numerical features are used to represent category data

because machine learning models can only work with numbers to find patterns. In the dataset, there are 100968 samples.

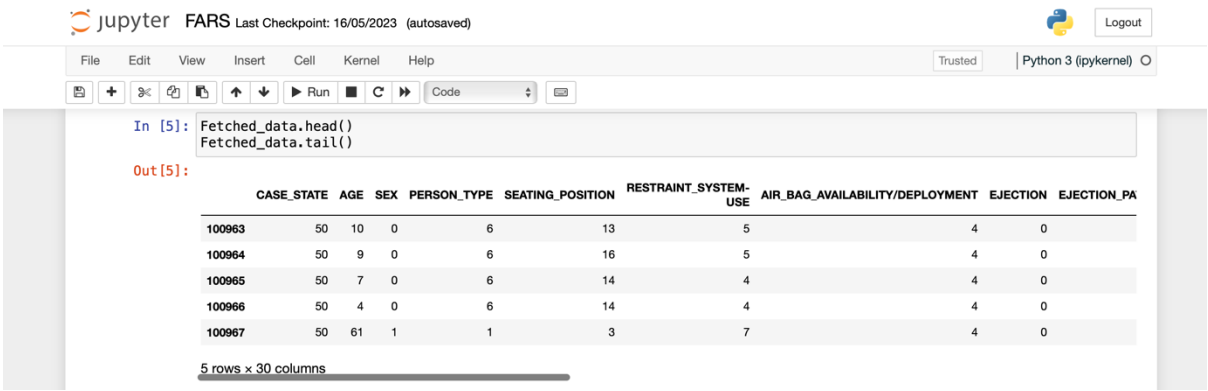


Figure 8

The data set's columns are as follows:

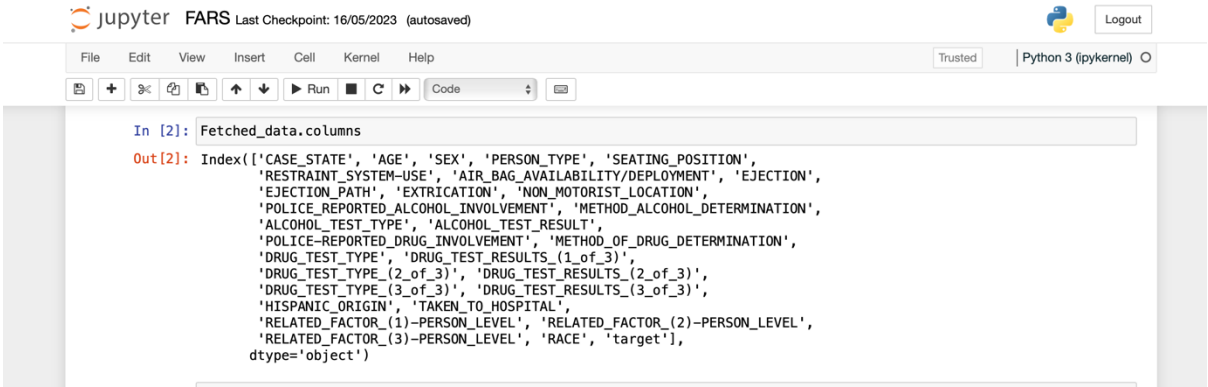


Figure 9

There are data samples in the dataset that are not balanced with respect to the objective variable. The model may be trained to predict data from the majority class but perform badly on data from the minority class as a result. SMOTE, a sampling method, is used to effectively address this and improve productivity. SMOTE is used to enrich the minority class samples, enhancing model recall, precision, and accuracy.

The number of samples for each class is displayed in the screenshot below. There are only 9 samples in minority class samples, such as "0." It is balanced with the majority class using the oversampling method.

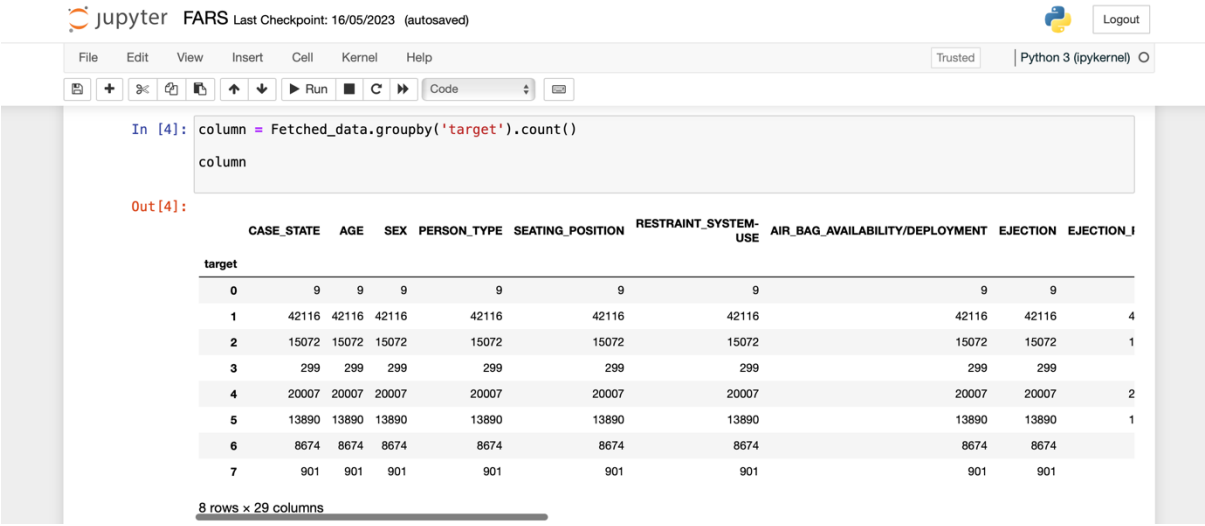


Figure 10

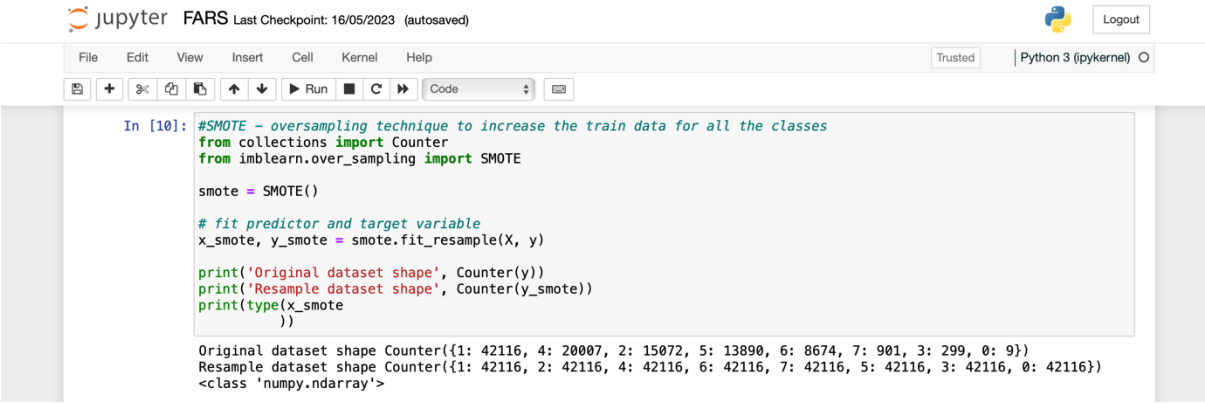


Figure 11

The variance of the characteristics is normalised using the usual scaler method. A standard scaler is needed since the dataset values differ on a larger scale. Age spans from 40 to 80, but most of the category entries have values of 1 or 0. The performance of the machine learning model could be harmed by this change. Following standardisation, the scaled values are as follows:

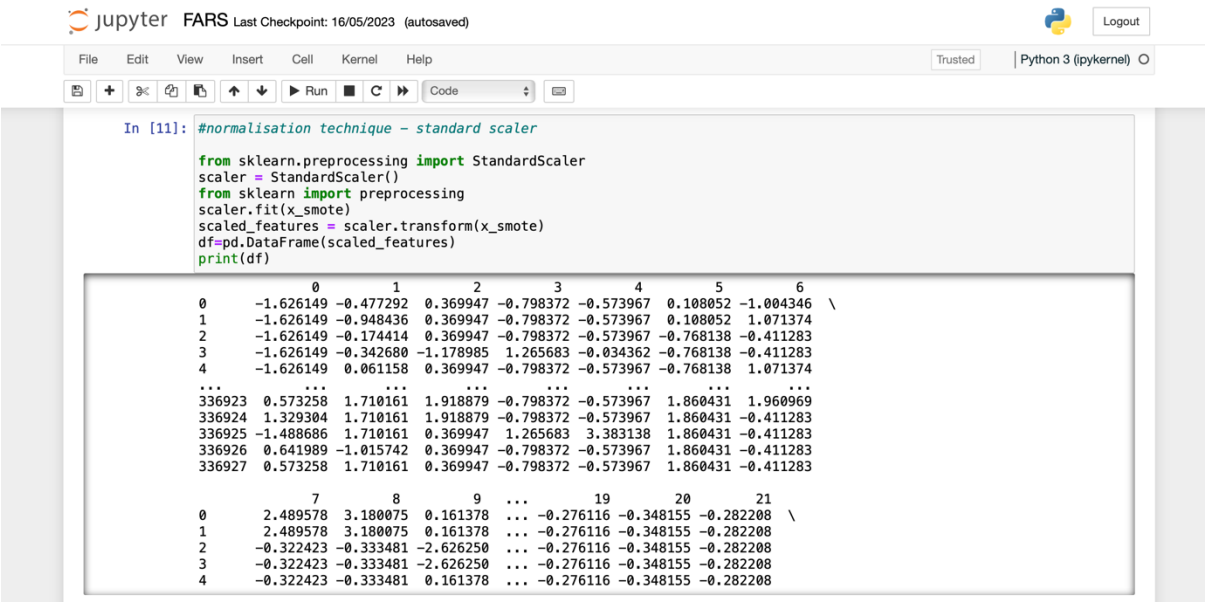


Figure 12

5. Logistic regression

Logistic regression is not the ideal method when there are more than two groups to divide the data sample into. The following classification problem, which entails dividing the data into eight different groups, can be resolved using logistic regression, though. The multi class classification approach or logistic regression is used to categorise the data samples. Each class will be handled as a binary classification problem, with 1 denoting that the sample belongs to a certain class and 0 denoting that it does not. The one vs. all method is the name given to the approach detailed below. A sigmoid function is used in the logistic regression method, and it produces a result of 1 if its return value exceeds 0.5 and a result of 0 if it is less than 0.5.

$$h = \frac{1}{1+e^{-z}}$$

where z is the input features multiplied by a theta value that has been initialised randomly.

The regression model receives the standardised data as input. The train/test split technique from sklearn is then used to divide the input data into test and training sets. After that, the model is trained using training data, and test data are used to calculate its correctness. The Predict function is used to calculate accuracy.

The accuracy and confusion matrix are displayed below.

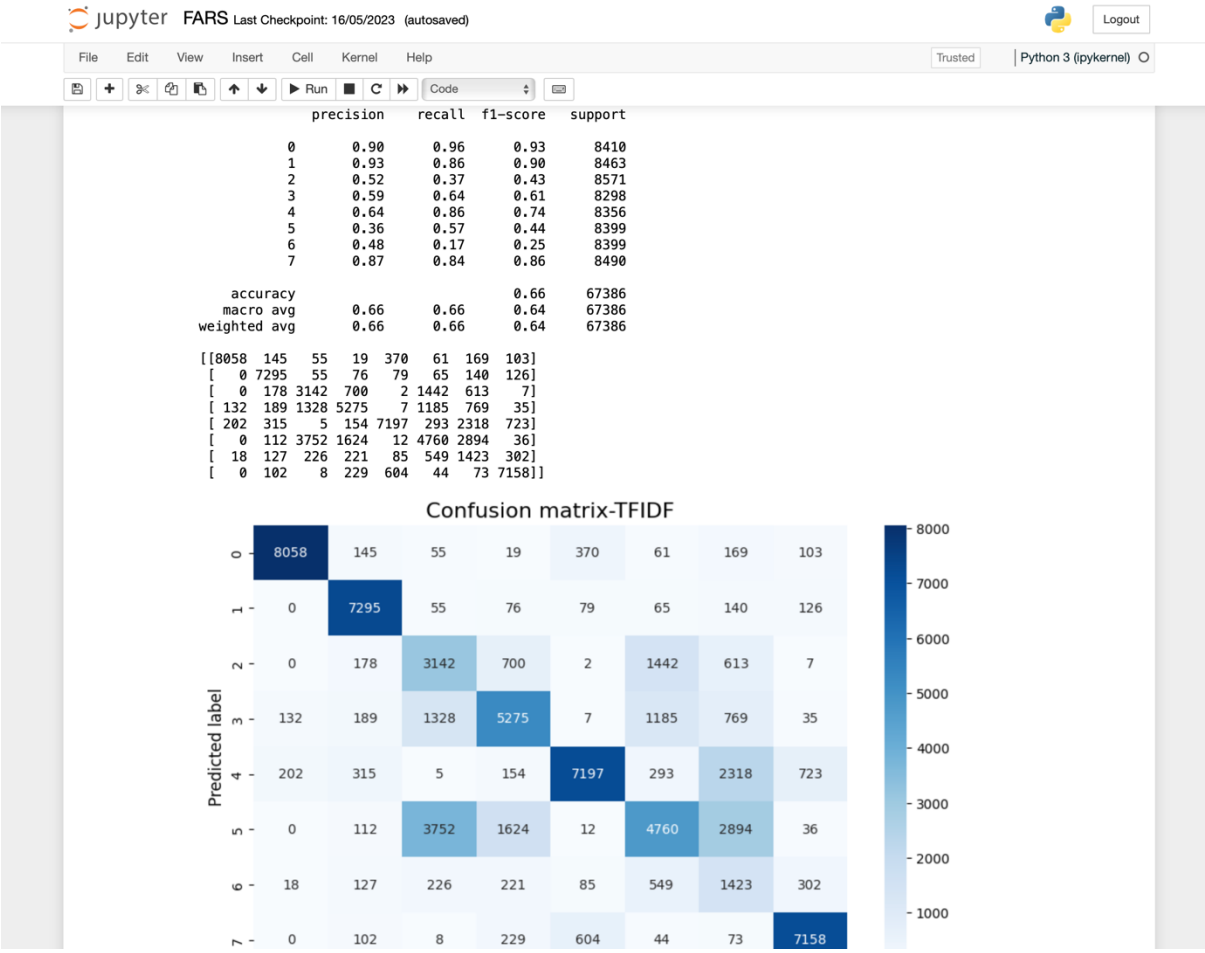


Figure 13

The accuracy is 66, which is lower than we could have anticipated, and we can see that there are a few false negative readings.

6. K Nearest Neighbour

Although KNN is not the best solution when dealing with a huge dataset, it can produce very high classification accuracy scores. KNN only functions with numerical data, and standardisation aids in enhancing accuracy. The KNN model receives the standardised value as input and trains.

The nearest annotated point is used in KNN to classify the data points. The Euclidean, Manhattan, Hamming, and Minkowski distances are used to determine how far apart two points are from one another.

KNN with SMOTE

The accuracy is 88, and the KNN prediction results following SMOTE are as follows:

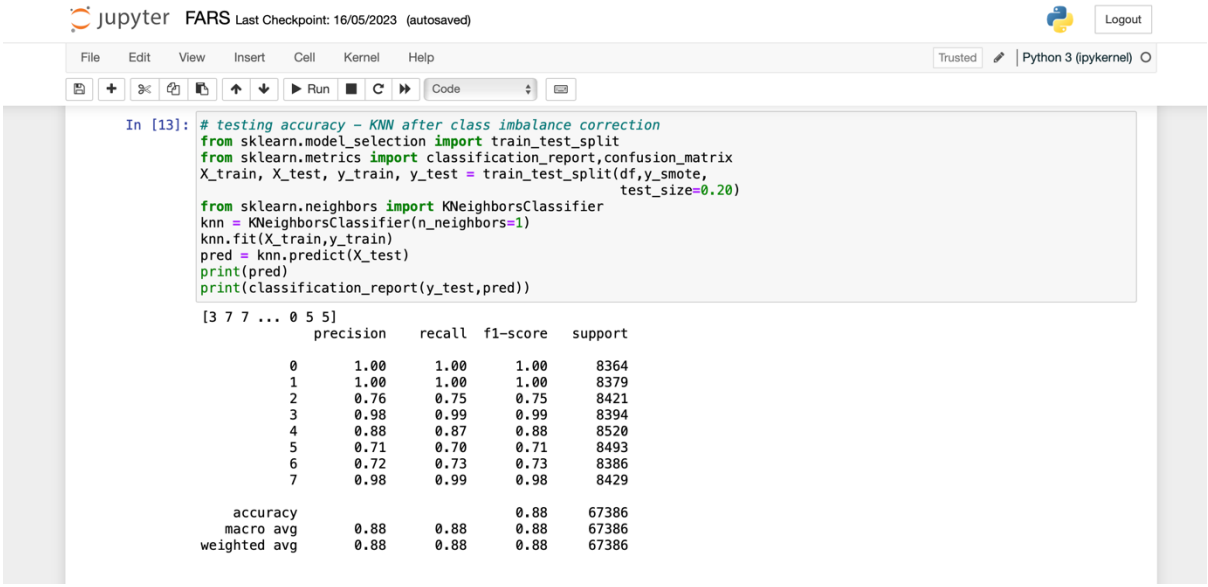


Figure 14

K is initially set to 1, and KNN is applied without the use of SMOTE directly to the data. It was clear that the SMOTE technique produced high accuracy because KNN without SMOTE performed worse than KNN utilising SMOTE data.

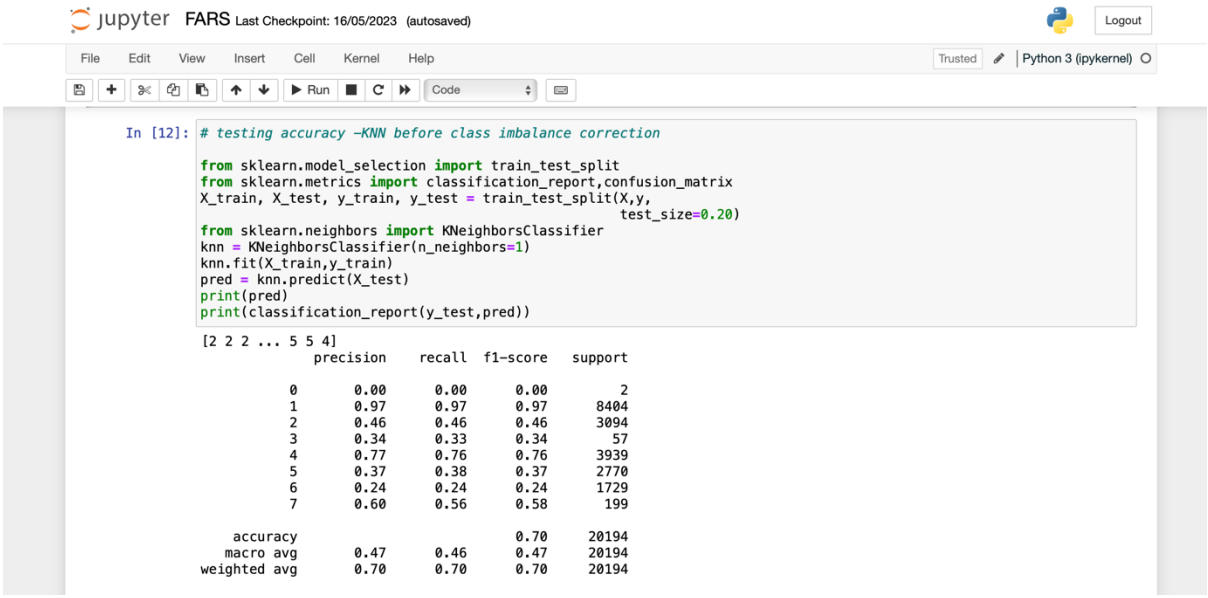


Figure 15

Hyper parameter tuning

The best value for KNN is discovered by hyper parameter tuning. When K is 1, the error rate is very low. In compared to K = 1, the mistake rate is significantly higher for other K values.

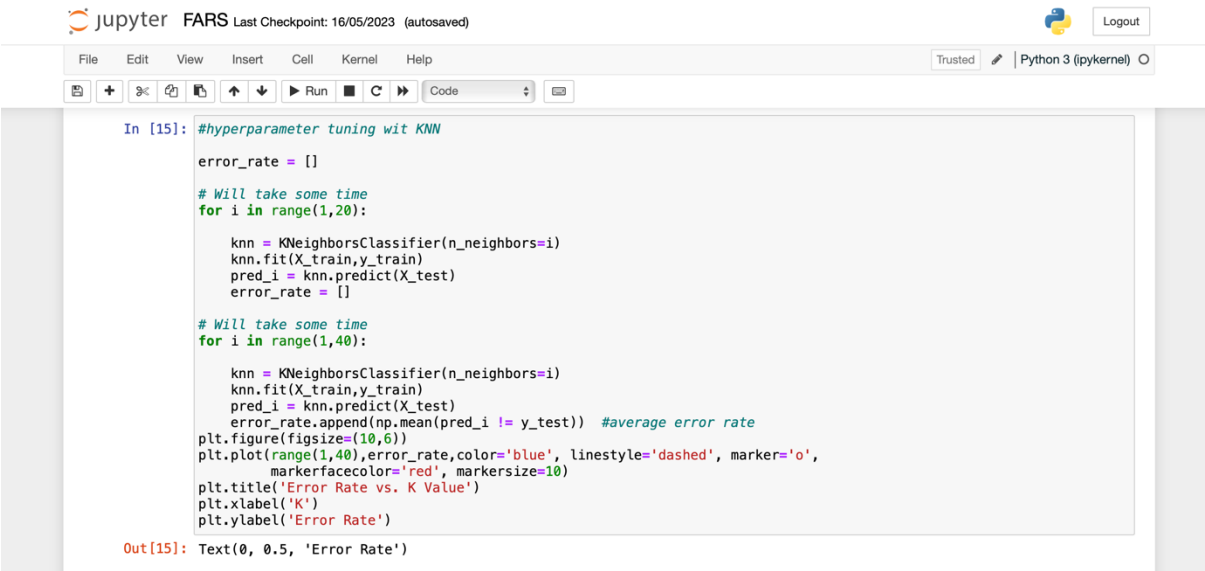


Figure 16

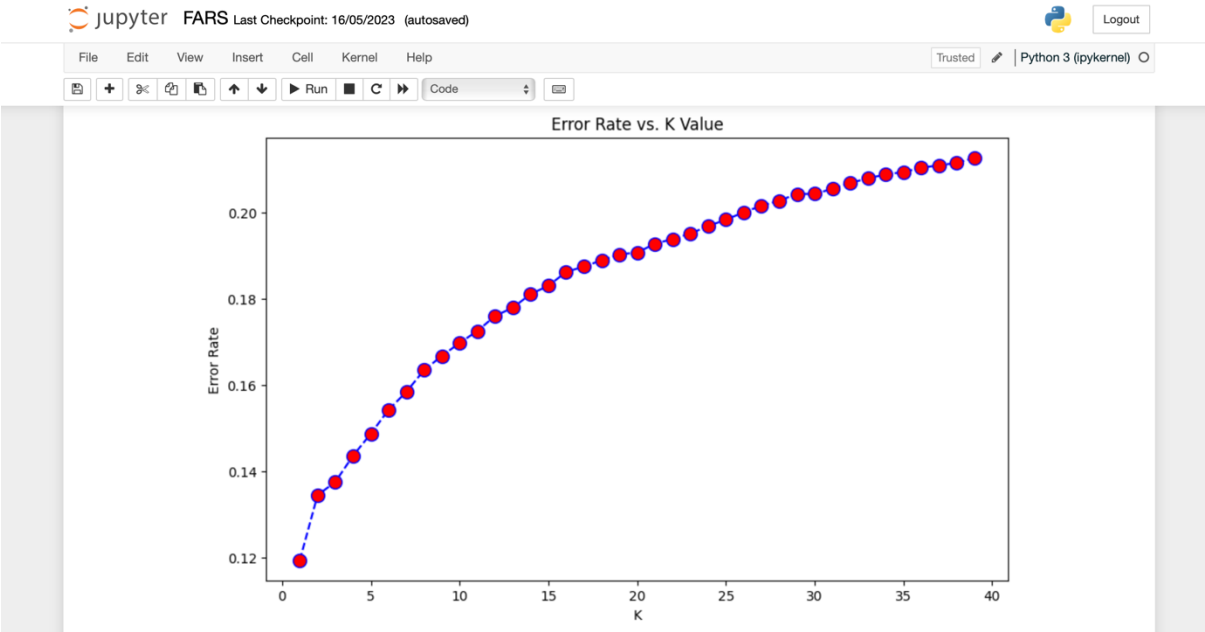


Figure 17

When K=1 accuracy is as below, 88,

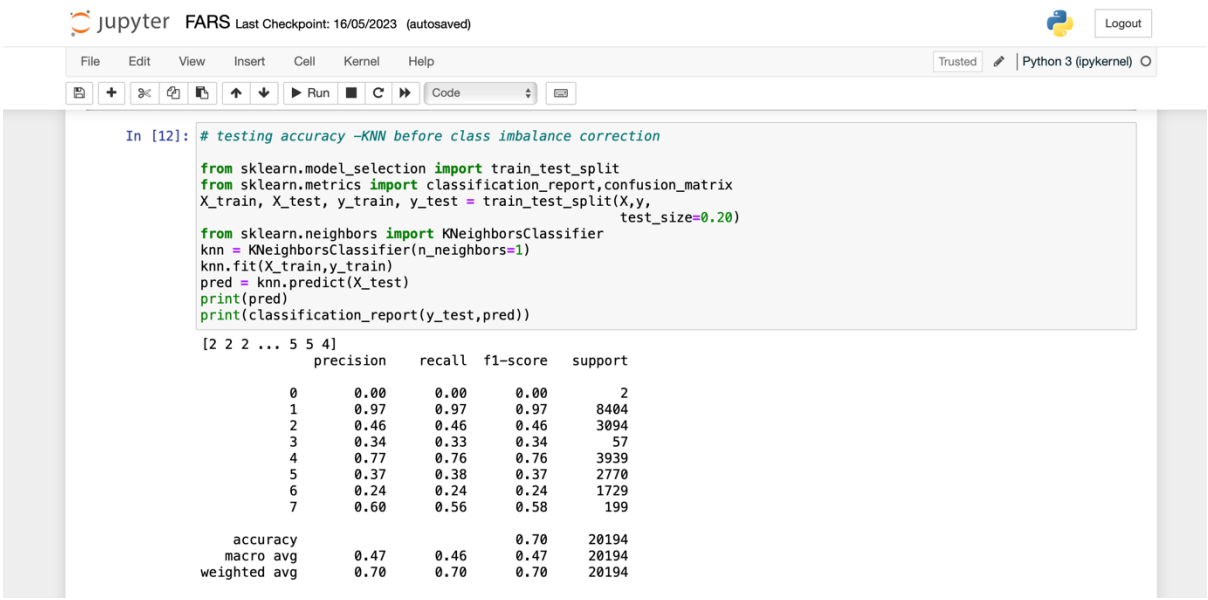


Figure 18

7. Decision tree classifier

A decision tree is just a list of choices made to accomplish a specific goal. Data samples are categorised in a decision tree from root to leaf node based on the decision rule inferred from the training data. Using a decision tree technique, one may forecast class labels by training the data and learning the fundamental decision-making principles from previous training data. The KNN model's accuracy score has already increased significantly because of splitting the oversampled SMOTE data. Using SMOTE data allows the model to be trained with a sizable sample size from each of the eight classes while maintaining balance. The accuracy rating was good, and the 88 obtained by using sample data. The decision tree classifier's results are as follows.

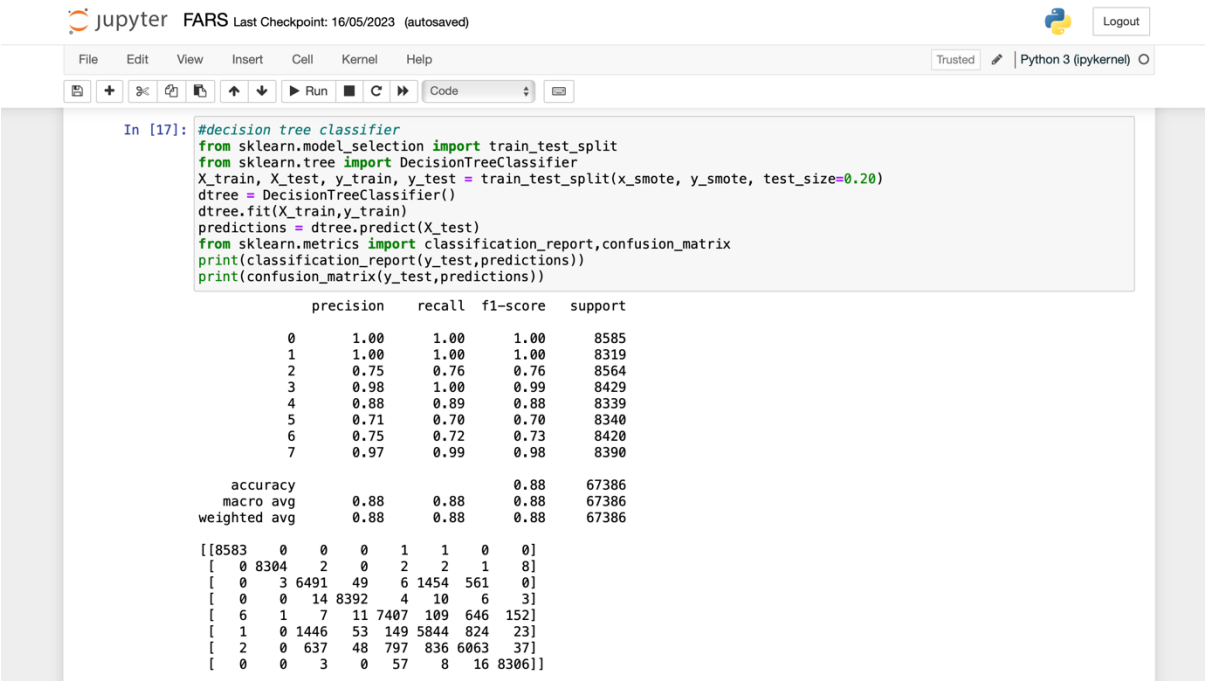


Figure 19

8. Random forest classifier

A random forest classifier is an illustration of an ensemble learning method in which a prediction is made using the results of various decision tree algorithms. An 8:2 split of the SMOTE data between the train and test datasets is fed into the model. By default, the N estimator is set to 100, meaning that 100 decision trees will be utilised for each prediction. Predictions are verified using test data once the model has been trained using simulated data. The model scored highly for accuracy, 90. As can be seen here, the confusion matrix contains a significant amount of real positive value.

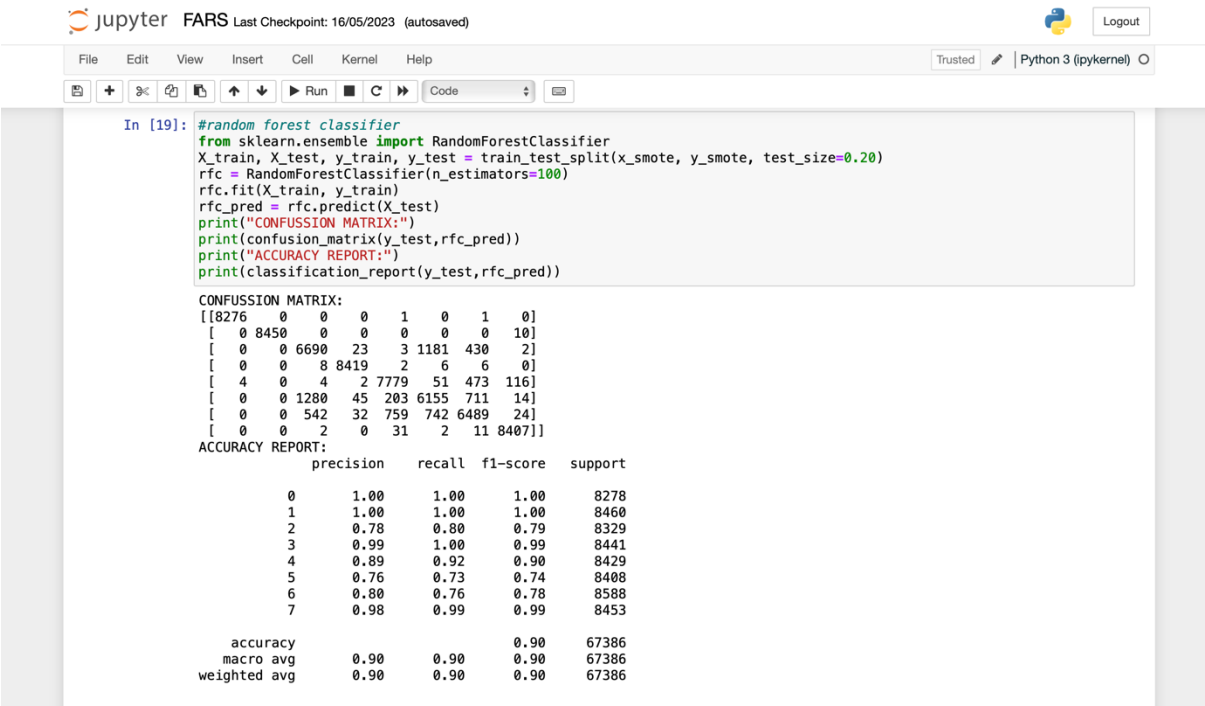


Figure 20

9. Cross Validation

Cross validation is a technique for confirming that a model is a legitimate machine learning model and has a high prediction score. As testing is entirely conducted on a new data set that the model is completely unaware of during training, the idea of dividing the data into test and train is a cross validation strategy. All of the models I've used use the splitting technique, but I've also used the K-fold cross validation method, in which we alternate splitting the training data.

In my eight iterations, the model will randomly accept train and test data, and the train and test data will change each time. The results of cross validation using the K-fold technique are displayed below. The model will be trained and predicted on multiple train and test data sets concurrently.

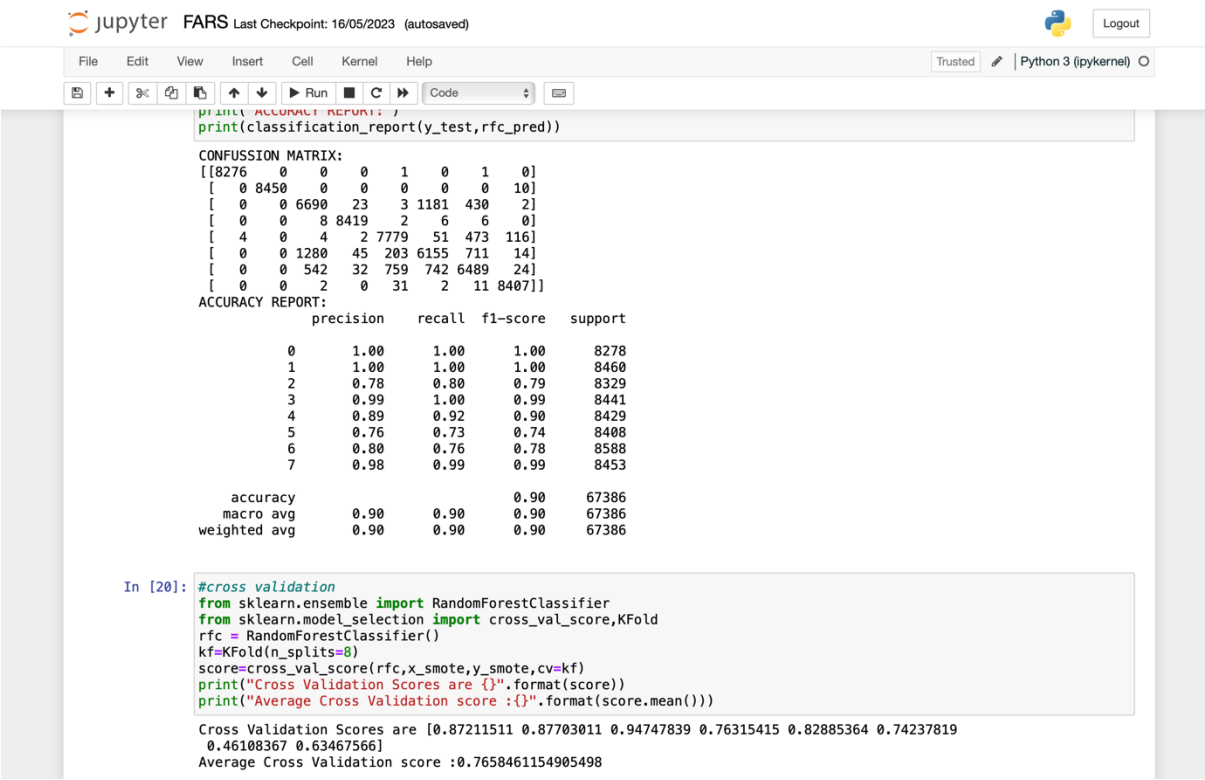


Figure 21

The model has a high accuracy rating, demonstrating its ability to accurately anticipate any new dataset.

10. Conclusion

All the models produced very accurate results, however the best classifier for the FARS dataset was, on balance, Random Forest. The accuracy results for several classifiers are shown below.

	Logistic Regression	KNN without SMOTE	KNN with SMOTE	Decision tree	Random forest
Accuracy	66	70	88	88	90

I used all the models to forecast the 8 various class labels, and SMOTE assisted me in achieving an exceptional accuracy score. Even though Random Forest has a very high accuracy score, I choose KNN because, for a particular sample of test data, Random Forest accuracy during cross validation is just 45.

TWITTER SENTIMENT ANALYSIS

1. Introduction

The data set include both user experiences and tweets concerning the airline sector. The work entails preparing Twitter text input and converting category target labels to integers. As shallow machine learning techniques, SVM, Nave Bayes, and the Random Forest classifier were used. The convolution neural network and basic neural network, both of which use deep learning, were also used.

2. Exploratory data analysis

The airline sentiment is the only dependent variable in the dataset, with the tweets and tweet id serving as the only independent factors. The dependent target column displays labels for the positive, negative, and neutral classes. The tweet id field was removed because it included only float values and was irrelevant to my investigation.

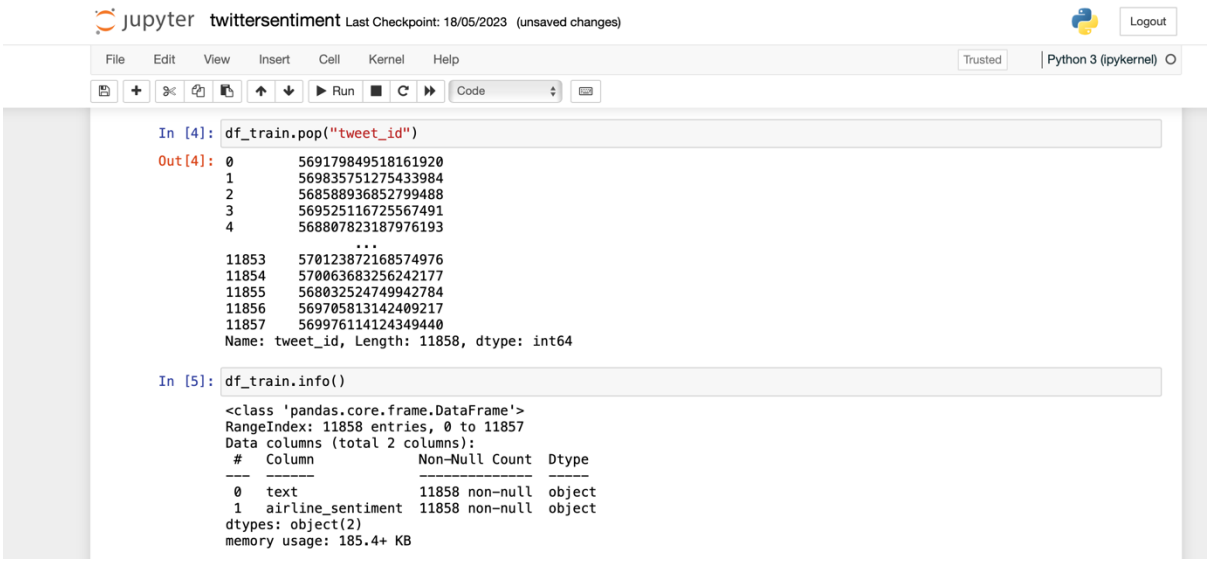


Figure 22

The dataset contains 11858 tweet samples with the corresponding sentiment. Additionally, the collection contains no null values. In train data, there are 7434 unfavourable tweets, compared to only 2510 neutral and 1914 positive tweets. The predictor variables are not evenly distributed among the model train samples.

jupyter twittersentiment Last Checkpoint: 18/05/2023 (unsaved changes) Logout

File Edit View Insert Cell Kernel Help Trusted Python 3 (ipykernel)

Run Code

```
In [9]: print(df_train.groupby('airline_sentiment').count())
        plot_target = df_train.groupby('airline_sentiment').count().plot(kind='bar', title='Distribution of data',legend=Fal
        plot_target.set_xticklabels(['Negative','Positive','Neutral'], rotation=0)
```

```
                     text
airline_sentiment
negative             7434
neutral              2510
positive             1914
```

```
Out[9]: [Text(0, 0, 'Negative'), Text(1, 0, 'Positive'), Text(2, 0, 'Neutral')]
```

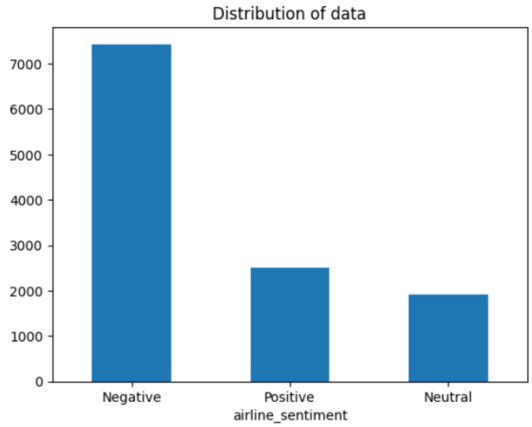


Figure 23

3. Data cleaning

The below-listed loading of the test and training data into the pandas format.

jupyter twittersentiment Last Checkpoint: 18/05/2023 (unsaved changes) Logout

File Edit View Insert Cell Kernel Help Trusted Python 3 (ipykernel)

Run Code

```
In [2]: # Importing libraries and functions
        import pandas as pd
        import numpy as np
        import re
        import nltk
        from nltk.corpus import stopwords
        from numpy import array
        #from wordcloud import WordCloud
        import matplotlib.pyplot as plt
        # nltk
        from sklearn.feature_extraction.text import TfidfVectorizer
        from scipy import stats
```

```
In [3]: df_train = pd.read_csv("Tweets_train (2).csv")
        df_test = pd.read_csv("Tweets_test (1) (1).csv", encoding="latin-1")
        df_dev = pd.read_csv("Tweets_dev (1).csv")
```

Figure 24

The data is subjected to various data cleaning processes because natural language will impair the accuracy score if supplied straight to machine learning models. Stop words are eliminated as the first step in the data cleaning process. For this, the "ENGLISH STOP WORDS" sklearn package is utilised. The term is eliminated if it appears in the library database since it is a stop word. The regex functionality is then used to remove the punctuation. URLs—which are just text that starts with "http"—can also be deleted using regex functionality. After that, the data is enhanced by the lemmatization and stemming algorithms from nltk, which are used to clean up the noisy words. The

procedure of cleansing the data yielded the following findings. Additionally, the categorical predictor variable is transformed into components as part of the cleaning process so that the machine learning model can understand it.

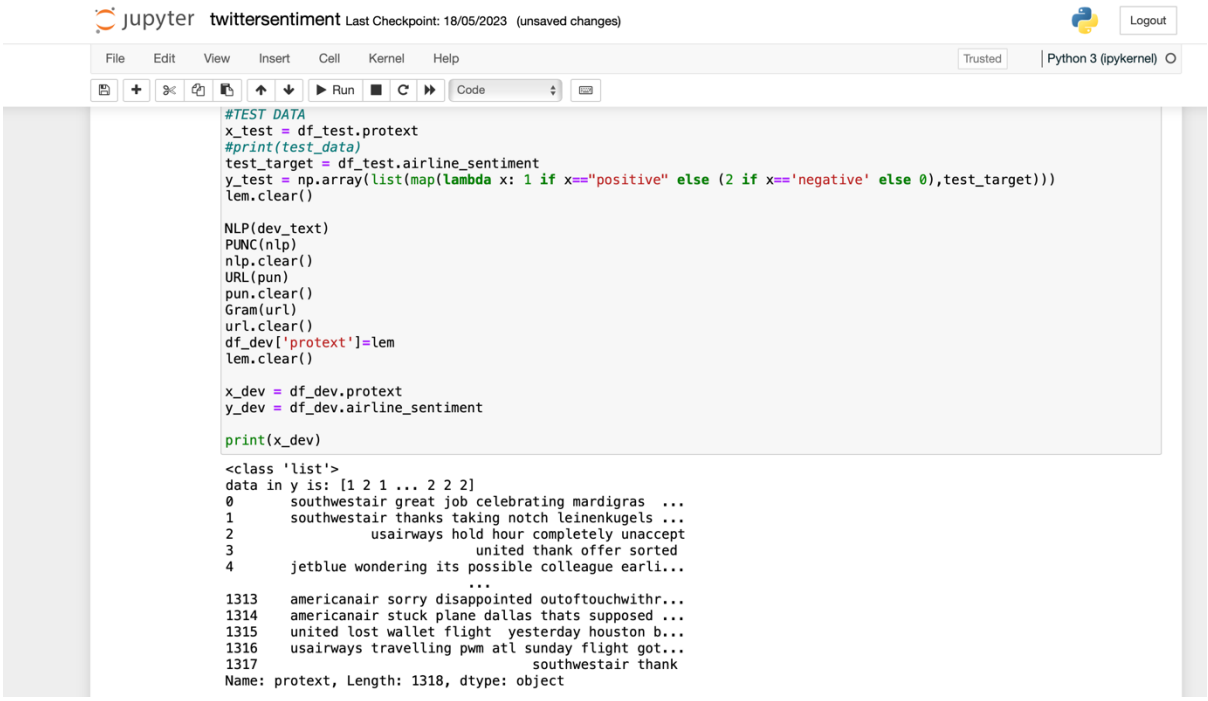


Figure 25

Data purification frequently requires converting text data to numerical data because machine learning algorithms can only comprehend numbers. For this, the GLOVE model is employed. The text data is tokenized using the Keras Tokenizer function before the GLOVE model is applied. Thus, there are 12409 terms in the lexicon.

Figure 26

The divided words are then padded to eliminate the balancing challenge. Because of padding, each word has a vector of 100 characters. It will only include 100 '0's, which represents an empty character, if there is no character. For this, the keras library pad Sequences is employed. A list of sequences (lists of integers) with length num samples is converted into a 2D Numpy array with shape (num samples, num timesteps) by this function. The GLOVE file is then imported, which contains a vector for each word. The

dataset word is replaced with its relevant vector data if the tokenized words are available in the GLOVE file.

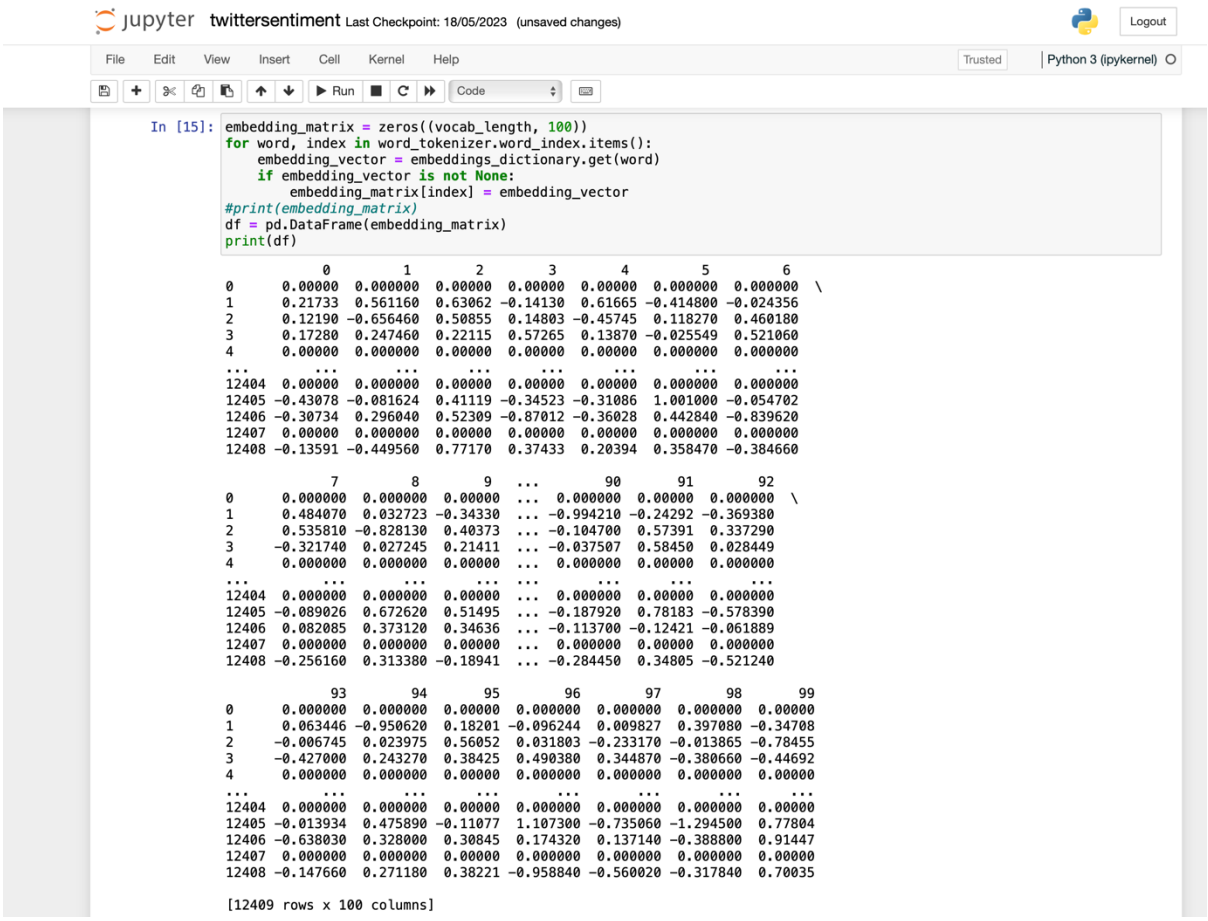


Figure 27

4. Deep Learning – Simple Neural Network

A fundamental neural network is the deep learning strategy applied in this instance. Deep learning is a technique that learns categories in a step-by-step manner using its hidden layer architecture, starting with low-level categories like letters, then progressing to slightly higher-level categories like words, and finally to higher level categories like sentences. A comprehensive representation of the image is formed by the network's neurons and nodes, each of which represents a different part of the entire network. I used deep learning because it provides highly accurate outcomes when given natural language input.

The model utilised here is a sequential model, which implies that the output of one layer is immediately fed into the next layer in all deep learning approaches. Next is the embedding layer. The Embedding layer contains the specification of the input data. Each word can be as long as 100 characters, which was established after padding. Both the entire data that is received immediately after Glove encoding and the entire vector matrix that results from the GLOVE encoding model are supplied as input. There are

three thick layers, and RELU is the activation function used on each layer. The output is altered, and categorisation is aided by the weights that are randomly created for each dense layer. The accuracy I was able to attain for the data below is 63 using 6 epochs. Entropy classification is the loss metric used, and the report includes the data at each epoch is as follows.

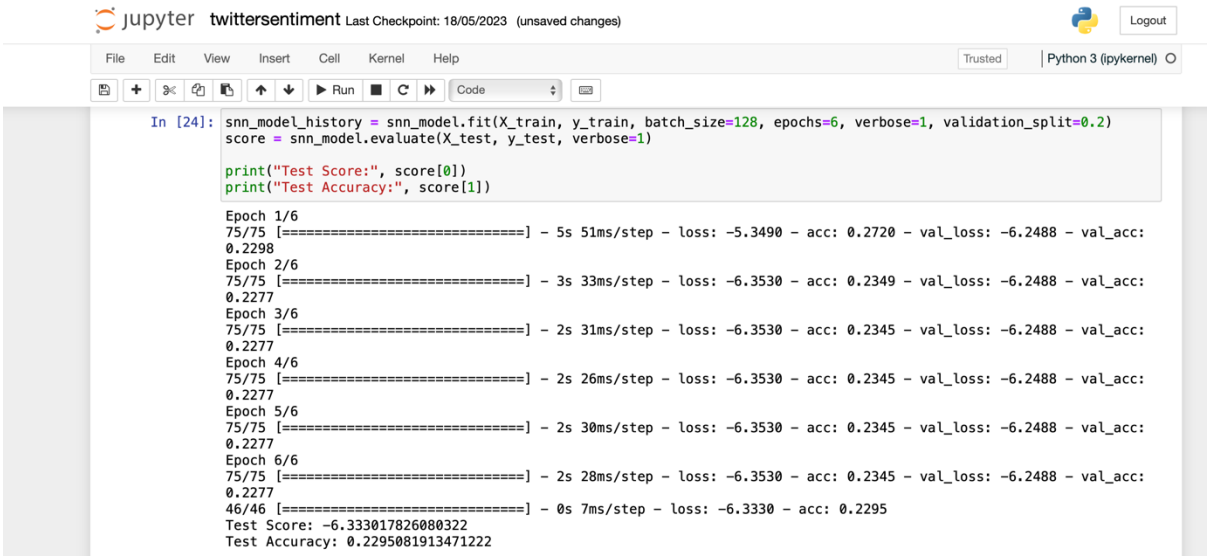


Figure 28

5. Shallow learning – Naïve bayes

A classification technique called Naive Bayes was created using the Bayes theorem. The Bayes theorem, developed by Thomas Bayes, determines the likelihood that an event will occur based on previously known conditions. Its foundation is the following equation:

$P(A|B) = P(A) * P(B|A)/P(B)$

The train and test parts of the model are vectorized using the TFID vectorizer. The vectorized test data is the prediction data. The accuracy of the F1 score is 68, which is less. Because Nave Bayes is a straightforward method based on probability, it has a poor accuracy.

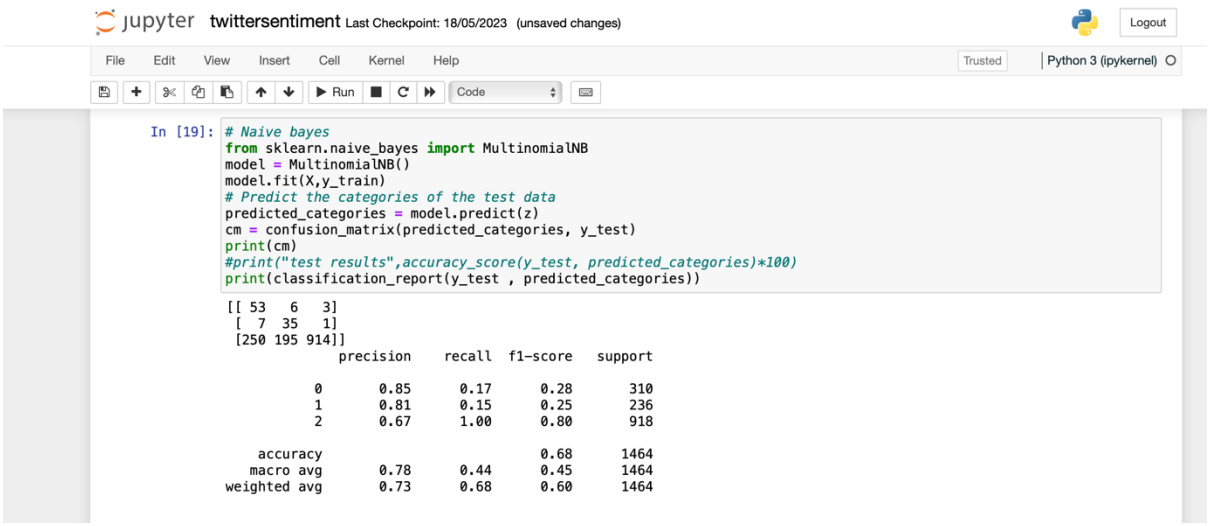


Figure 29

6. Shallow Learning – SVM

A supervised learning algorithm known as SVM does well with little data. It more accurately assigns each sample to a particular label. The initial training data, which was vectorized early using TFID Vectorizer as follows, is used to fit the model.

On a graph, the data is shown, and the hyper plane divides each class into groups. When there are more than two classes in SVM, this method, known as the one-to-test strategy, is utilised.

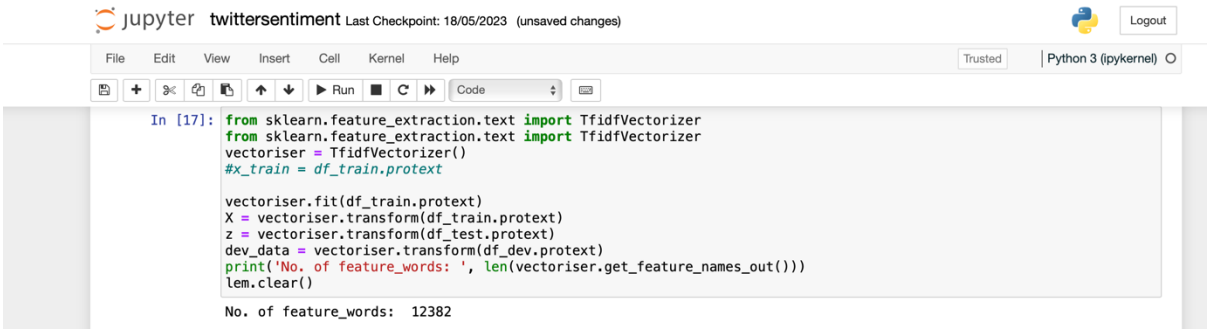


Figure 30

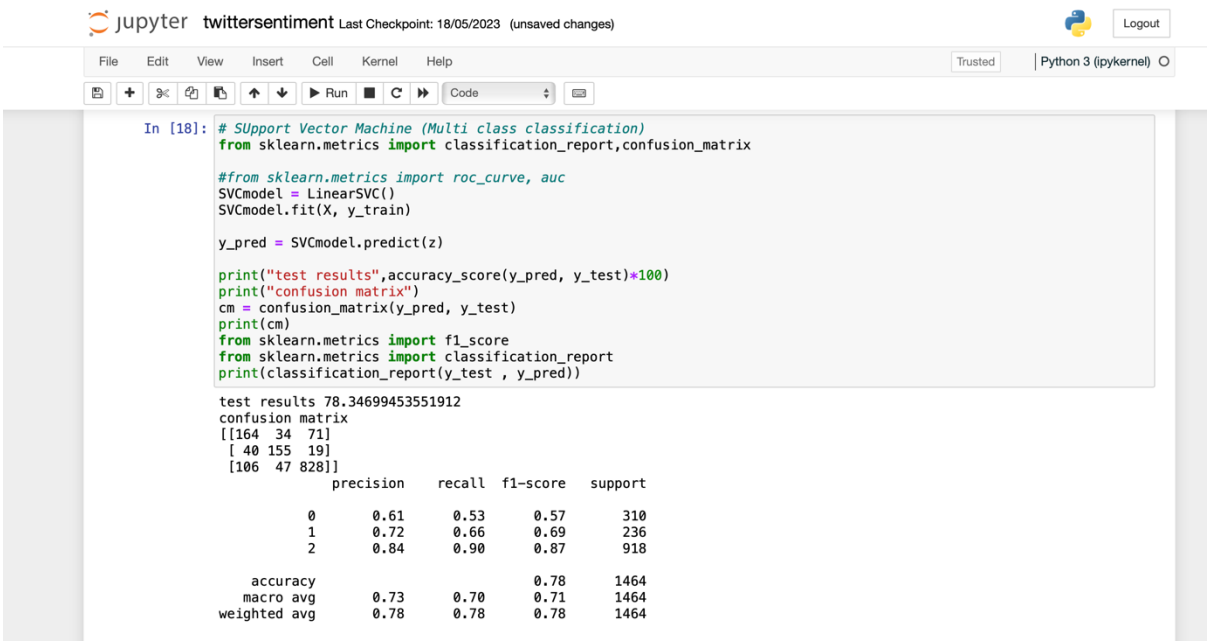


Figure 31

Here, applying the deep learning technique to the vectorized data used for the GLOVE encoding method only produced 62 accuracy. After vectorizing the data, the SVM algorithms produced an accuracy of 78 that is commendable. SVM generated an extremely high accuracy score due to the 1k samples it had, even though deep learning methods are meant to produce outcomes with high accuracy.

7. Shallow classifier – Random Forest

In an ensemble learning method called random forest, the output of numerous decision trees is considered when labelling a particular sample.

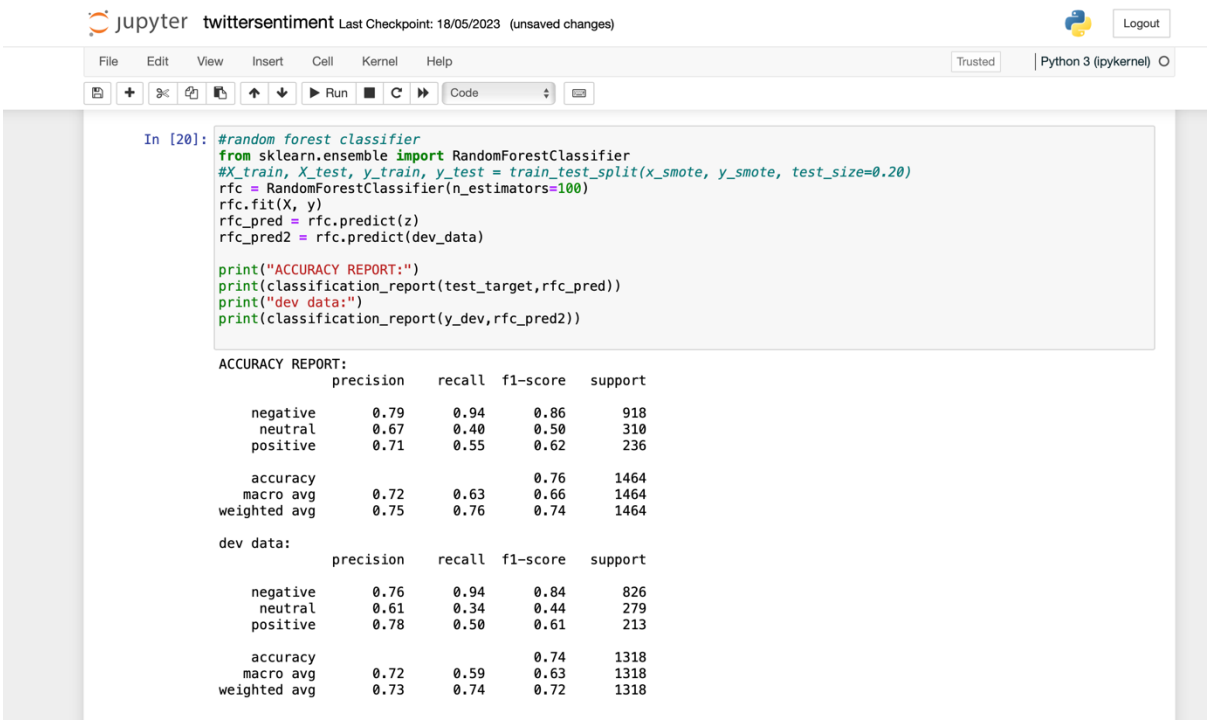


Figure 32

In the n_estimator option, which has the value 100, it is stated that 100 decision trees have been run in parallel, with the results being combined to forecast the class labels. As a part of the fit technique, the model begins to learn from the train data. With the help of TFID vectorizer, the train data is transformed into vectors that are then sent to the random forest classifier. The results of the test data are predicted by the machine learning model after it has learned from the train data. In comparison to most models, the F1 score accuracy is 77, which is greater.

As shown below, the random forest classifier's findings for development data.

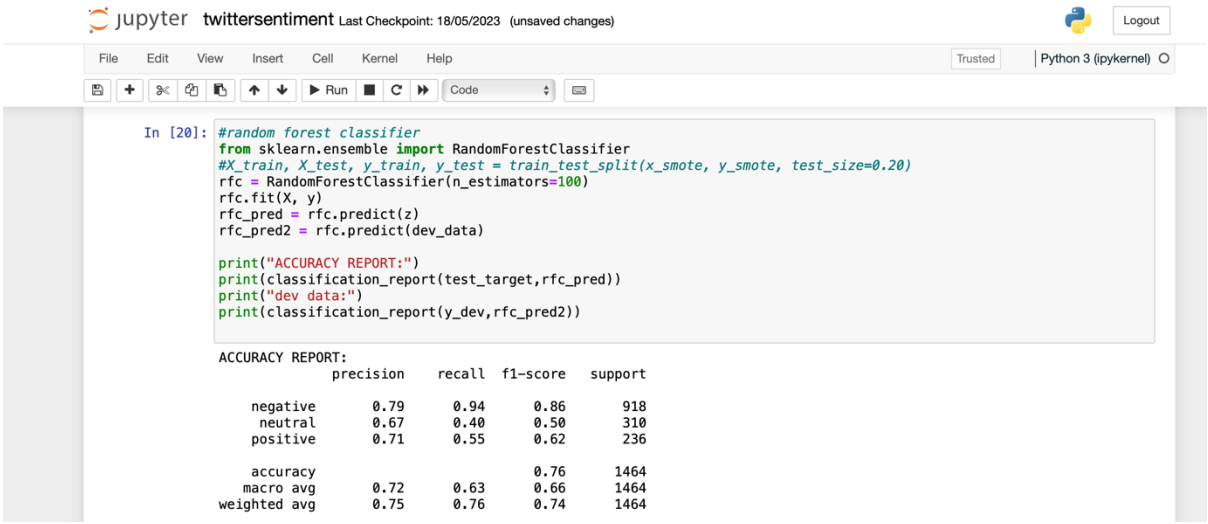


Figure 33

8. Conclusion

Since SVM with TFID produced a very high accuracy score compared to other models, I prefer it.

	SNN with GLOVE encoding	SVM with TFID vectorizer	Naïve bayes with TFID vectorizer	Random forest with TFID vectorizer
Accuracy	63	78	68	77

All machine learning models' accuracy is determined by the quantity of true positive values they produce from classes with negative labels and poor scores for other classes. As a result, the model is unable to accurately forecast neutral and positive values, while having learned how to do so with negative values. This results from an imbalance in the data samples. All machine learning models' accuracy is declining because negative sample sizes are disproportionately bigger than those of the other two classes.

References

[1] – Understanding FARS dataset.
https://health.gov/healthypeople/objectives-and-data/data-sources-and-methods/data-sources/fatality-analysis-reporting-system-fars
[2] - https://towardsdatascience.com/an-introduction-to-preprocessing-data-for-machine-learning-8325427f07ab
[3] - https://scikit-learn.org/stable/
[4] - https://medium.com/analytics-vidhya/understanding-embedding-layer-in-keras-bbe3ff1327ce#:~:text=Embedding%20layer%20enables%20us%20to,way%20along%20with%20reduced%20dimensions.
[5] - https://towardsdatascience.com/stemming-corpus-with-nltk-7a6a6d02d3e5 stemming
[6] - https://towardsdatascience.com/tf-idf-simplified-aba19d5f5530
regex numpy pandas
[7] - https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4
[8] - https://www.baeldung.com/cs/svm-multiclass-classification
[9] - https://www.tensorflow.org/
[11] - https://www.analyticsvidhya.com/blog/2021/06/exploratory-data-analysis-using-data-visualization-techniques/#:~:text=A%20histogram%20is%20a%20value,at%20the%20center(mean).