JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Airline Flight Delay Prediction System

Krushna Mudigonda [1], Sonal Jagtap[2], Ravina Asole[3], Kalyani Sisodiya [4], Kartik Mirkhale [5]
Department of E&TC, SKNCOE, SPPU, Pune
[1]krushnamudigonda99@gmail.com, [2]skjagtap.skncoe@sinhgad.edu,
[3]ravina.asole.skncoe@sinhgad.edu, [4]kalyamisisodiya829@gmail.com, [5]kartikmirkhale2000@gmail.com

Abstract — **It is essential for both passengers and airlines to anticipate flight delays because they not only result in significant financial losses but also damage the reputation that has been developed over many years and cost customers valuable time. By utilising the data that is currently available, this dataset seeks to estimate the arrival delay of a scheduled individual flight at the destination airport. The prediction model proposed in this paper uses supervised machine learning methods to forecast airline arrival delays. We use a total of five methods to forecast the precise flight delay. Each algorithm's performance is examined. The model is provided flight data as well as meteorological data. With this information, each algorithm makes a specific prediction.**

Keywords— **Flight delay prediction, Airline Transport, Machine learning, Classification Algorithms, Data Analytics**

I. Introduction

Because air travel is so important to the industry's economy, airlines and airports must raise the caliber of their offerings. A major problem for airports and airline companies nowadays is flight delays. Also, passengers are concerned when a flight is delayed, and the agency and the airport experience additional costs as a result. Flight delays cost the U.S. government between 31 and 40 billion dollars in 2007. 76% of the flights landed on time in 2017.

The proportion of on-time flights dropped by 8.5% from 2016 to 2017. The following can be listed as some of the causes of flight delays: safety, weather, part shortages, technical difficulties with aircraft equipment, and delays in the flight crew. Unavoidable flight delays have significant financial repercussions for travelers, airlines, and airports. Moreover, delays can damage the environment by causing higher fuel costs and harmful gas emissions.

II. Literature Survey

N. K. Singh and K. R. K. Rao. provides a comprehensive overview of the existing research in airline flight delay prediction using ML techniques. The authors review various ML models, feature selection techniques, and evaluation metrics used in the literature. They also discuss the challenges and future directions of this research area. A. H. Alazzawi et al. focuses on the application of ML techniques in predicting flight delays. The authors review the various ML models, data sources, and features used in the literature. They also discuss the limitations and challenges of the current research and suggest future research directions. *M. Ghasemi et al.* provides a systematic analysis of the existing research in airline flight delay prediction using ML techniques. The authors review the various ML models, data sources, and features used in the literature. They also discuss the strengths and limitations of the current research and provide suggestions for future research. *A. Kumar et al.* focuses on the application of ML techniques in predicting flight delays in the airline industry. The authors review the various ML models, data sources, and features used in the literature. They also discuss the challenges and future directions of this research area.

III. Methedology

This project could be relevant in field applications of machine learning such as surveys, monitoring and studies in maintaining a good ecosystem for the Airline field. It is also applicable in Train , bus & other transportation system for Delay prediction. The aim of this research is to predict flight delays, which are the highest economy-producing field for many countries, and among many transportation modes, this one is the fastest and most comfortable, so identifying and reducing flight delays can dramatically reduce flight delays and save huge amounts of money by using machine-learning algorithms.

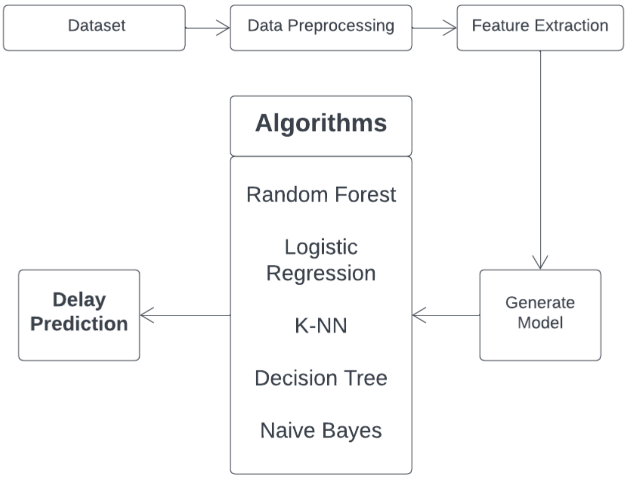


Fig.1 Data for prediction process

IV. Methodology Description

All the data manipulation or dropping together happens before it is used, so the dataset is cleaned up in the pre-processing Process. All the representation of the dataset is in the form of graphics. So, it is efficient to communicate in this process. Parameter Selection for getting proper configuration model information from the given data. The parameters are mean (mu) and standard deviation (sigma). All the processed data is usable for building the project model. With the help of algorithms, our model can learn, test, and predict the features of the data. The prediction process depends on the model. So, when the output of an algorithm comes from a dataset and is applied to new data to forecast the likelihood of a particular outcome, The user sends input to the modelling dataset for prediction. A total of five algorithms are going to be used for the prediction purpose. Random forest, logistic regression, decision tree, naive bayes, and K-nearest neighbors are going to be implemented. All algorithms perform the operation and give their particular output. It predicts the accuracy of a flight delay with its proper classification report and final status.The user sends input to the modelling dataset for prediction. inputs like flight numbers, timings, etc. If the flight is delayed, then it shows a "Delay prediction." And if the flight is on time, then it shows "Flight on Time" to the application users.

We will upload our dataset into application. The quality of the data should be checked before applying our algorithms. Transforming raw data into numerical features that can be processed while preserving the information in the original data set. After extractingfeatures from the dataset we will generate our algorithms on that dataset. We will plot the accuracies comparison graph between all the algorithms.

V. Experimentation

In Simulation,
The performance of the classifier can be calculated from the confusion matrix. After being compared to the actual result, the classifier results can generate four values namely (TP), (TN), (FP), (FN). Four measures were used to measure the performance of selected algorithms: accuracy, precision, recall, and F1 score. These measures are all positively related to the quality of algorithms. Consequently, for a specific algorithm, the higher values of these measures are, the better their performances are. The value of the four measures can be obtained by calculations using these parameters:

Accuracy = TP+TN / TP+TN+FP+FN Where,
Precision= TP / TP+FP
Recall= TP / TP+FN
F1−Score = 2 × (Recall × Precision) / (Recall + Precision)

True Positive (TP)
True Negative (TN)
False Positive (FP)
False Negative (FN)

All algorithms perform the operation and give their particular output. It predicts the accuracy of a flight delay with its proper classification report and final status. The user sends input to the modelling dataset for prediction. inputs like flight numbers, timings, etc. If the flight is delayed, then it shows a "Delay prediction." And if the flight is on time, then it shows "Flight on Time" to the application users.

A. *Flowchart*

We will upload our dataset into application. The quality of the data should be checked before applying our algorithms. Transforming raw data into numerical features that can be processed while preserving the information in the original data set. After extractingfeatures from the dataset we will generate our algorithms on that dataset. We will plot the accuracies comparison graph between all the algorithms.

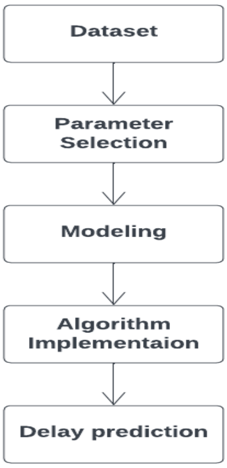


Fig.2 Flow of the Prediction Process

B. *Accuracy Graph*

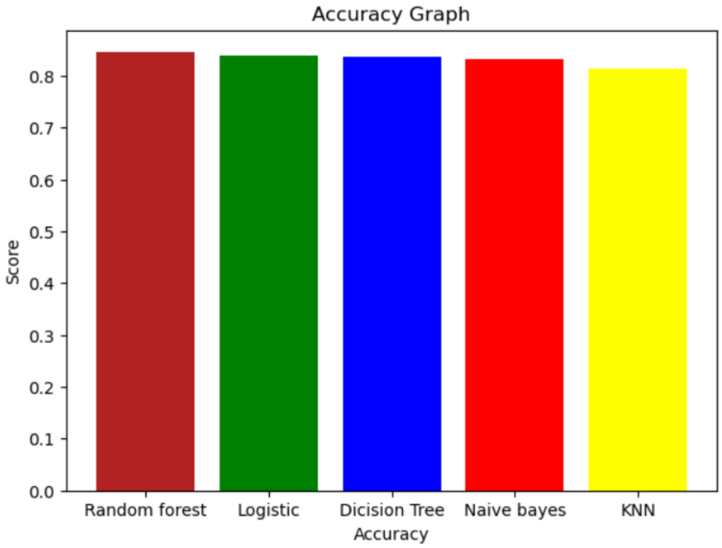


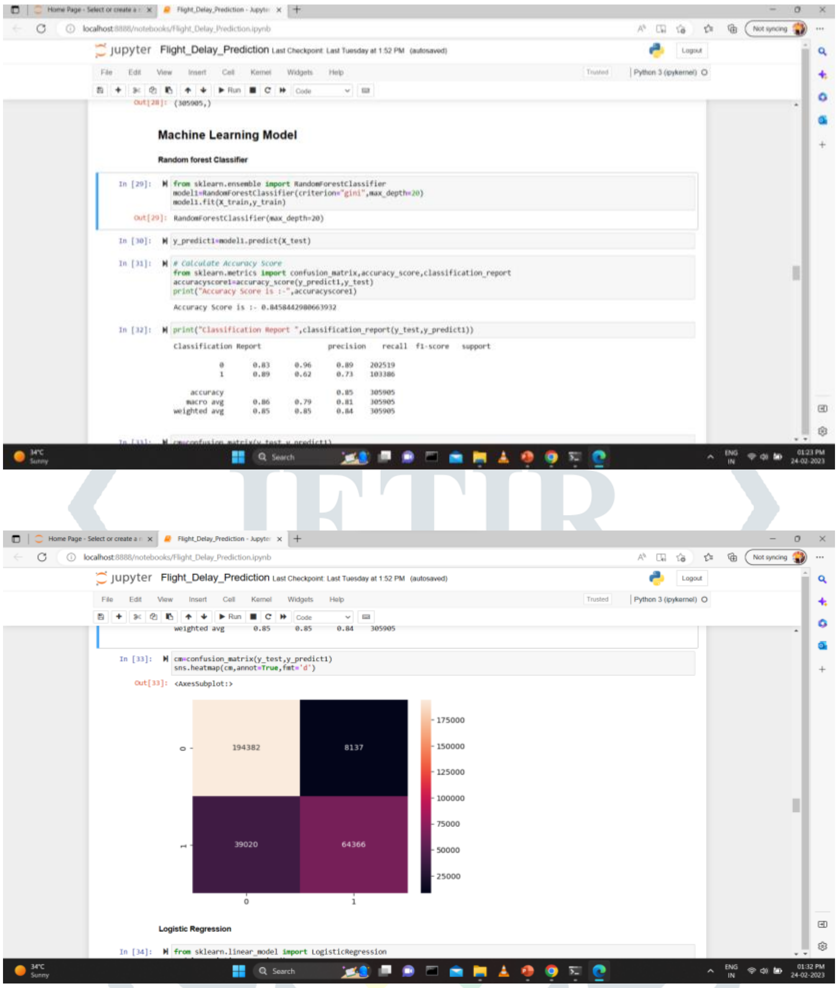
Fig.3 Accuracy Graph

Accuracy is calculated using comparison of all the algorithms based on graph. The final prediction of every algorithm is shown in fig 3. It determines that which algorithm gives the highest accurate prediction.

V. Experimental Results

A. Random Forest :

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.



Home Page - Select or create a
Flight_Delay_Prediction - Jupyter
localhost:8888/notebooks/Flight_Delay_Prediction.ipynb
jupyter Flight_Delay_Prediction Last Checkpoint: Last Tuesday at 1:52 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Search

```
Out[28]: (305905,)
```

Machine Learning Model

Random forest Classifier

```
In [29]:  from sklearn.ensemble import RandomForestClassifier
          model1=RandomForestClassifier(criterion="gini",max_depth=20)
          model1.fit(X_train,y_train)

Out[29]:  RandomForestClassifier(max_depth=20)

In [30]:  y_predict1=model1.predict(X_test)

In [31]:  # Calculate Accuracy Score
          from sklearn.metrics import confusion_matrix,accuracy_score,classification_report
          accuracyscore1=accuracy_score(y_predict1,y_test)
          print("Accuracy Score is :-",accuracyscore1)

          Accuracy Score is :- 0.8458442980663932

In [32]:  print("Classification Report ",classification_report(y_test,y_predict1))

          Classification Report               precision    recall  f1-score   support

                     0       0.83      0.96      0.89    202519
                     1       0.89      0.62      0.73    103386

              accuracy                           0.85    305905
             macro avg       0.86      0.79      0.81    305905
          weighted avg       0.85      0.85      0.84    305905

In [33]:  cm=confusion_matrix(y_test,y_predict1)
          sns.heatmap(cm,annot=True,fmt='d')

Out[33]:  <AxesSubplot:>
```

Logistic Regression

```
In [34]:  from sklearn.linear_model import LogisticRegression
```

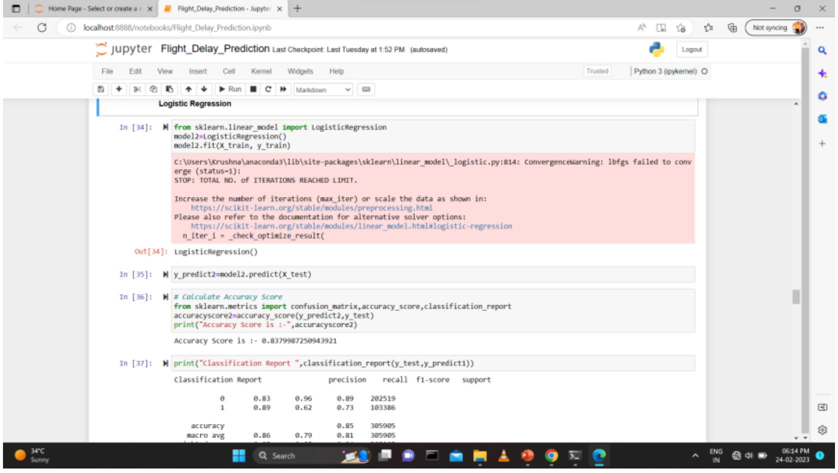
Fig.4 Random Forest output

Expected outcome – 1) Random forest –
Random Forest output – 0.8458 = 84.58% Accuracy with classification Report

In this Fig.3 we observed that we got 84.58% accuracy by the Random Forest algorithm. And it shows the classification report of the output.

B. Logistic regression :

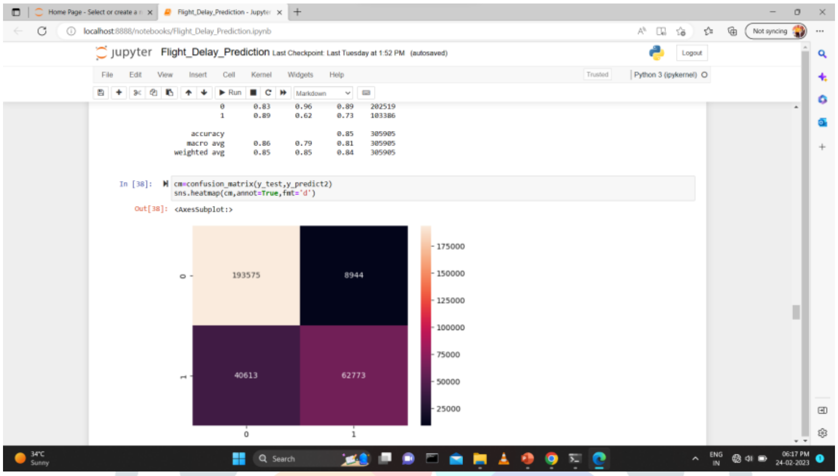
It is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.



Flight_Delay_Prediction - Jupyter
localhost:8888/notebooks/Flight_Delay_Prediction.ipynb
jupyter Flight_Delay_Prediction Last Checkpoint: Last Tuesday at 1:52 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Markdown

Logistic Regression

```
In [34]:  from sklearn.linear_model import LogisticRegression
          model2=LogisticRegression()
          model2.fit(X_train, y_train)

          C:\Users\Krushna\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfgs failed to conv
          erge (status=1):
          STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

          Increase the number of iterations (max_iter) or scale the data as shown in:
              https://scikit-learn.org/stable/modules/preprocessing.html
          Please also refer to the documentation for alternative solver options:
              https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
            n_iter_i = _check_optimize_result(

Out[34]:  LogisticRegression()

In [35]:  y_predict2=model2.predict(X_test)

In [36]:  # Calculate Accuracy Score
          from sklearn.metrics import confusion_matrix,accuracy_score,classification_report
          accuracyscore2=accuracy_score(y_predict2,y_test)
          print("Accuracy Score is :-",accuracyscore2)

          Accuracy Score is :- 0.8379987250943921

In [37]:  print("Classification Report ",classification_report(y_test,y_predict1))

          Classification Report               precision    recall  f1-score   support

                     0       0.83      0.96      0.89    202519
                     1       0.89      0.62      0.73    103386

              accuracy                           0.85    305905
             macro avg       0.86      0.79      0.81    305905
```



Flight_Delay_Prediction - Jupyter
localhost:8888/notebooks/Flight_Delay_Prediction.ipynb
jupyter Flight_Delay_Prediction Last Checkpoint: Last Tuesday at 1:52 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Markdown
193575
8944
40613
62773
0
1
0
1
175000
150000
125000
100000
75000
50000
25000

```
                     0       0.83      0.96      0.89    202519
                     1       0.89      0.62      0.73    103386

              accuracy                           0.85    305905
             macro avg       0.86      0.79      0.81    305905
          weighted avg       0.85      0.85      0.84    305905

In [38]:  cm=confusion_matrix(y_test,y_predict2)
          sns.heatmap(cm,annot=True,fmt='d')

Out[38]:  <AxesSubplot:>
```

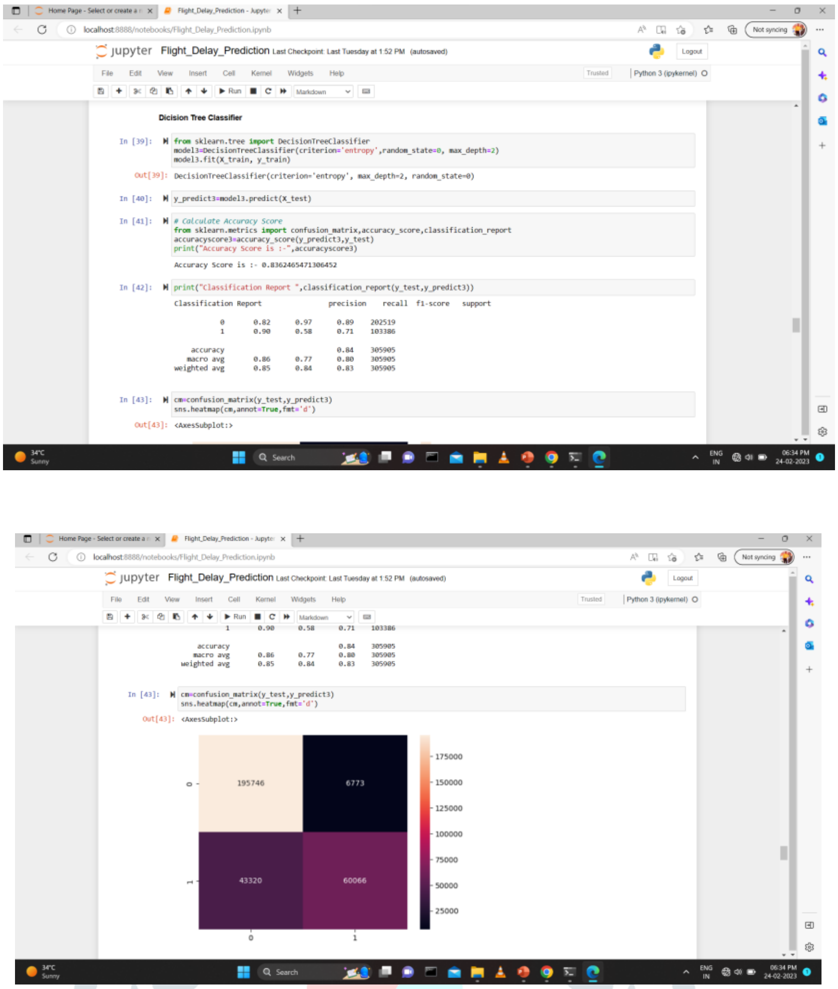
Fig.5 Logistic Regression output

Expected outcome – 2) Logistic Regression
Logistic Regression output – 0.8379 = 83.79% Accuracy with Classification Report

In this Fig.5 we observed that we got 83.79% accuracy by the Logistic Regression algorithm. And it shows the classification report of the output.

C. Decision tree :

is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.



Dicision Tree Classifier

```
In [39]:  from sklearn.tree import DecisionTreeClassifier
          model3=DecisionTreeClassifier(criterion='entropy',random_state=0, max_depth=2)
          model3.fit(X_train, y_train)

Out[39]:  DecisionTreeClassifier(criterion='entropy', max_depth=2, random_state=0)

In [40]:  y_predict3=model3.predict(X_test)

In [41]:  # Calculate Accuracy Score
          from sklearn.metrics import confusion_matrix,accuracy_score,classification_report
          accuracyscore3=accuracy_score(y_predict3,y_test)
          print("Accuracy Score is :-",accuracyscore3)

          Accuracy Score is :- 0.8362465471306452

In [42]:  print("Classification Report ",classification_report(y_test,y_predict3))

          Classification Report               precision    recall  f1-score   support

                     0       0.82      0.97      0.89    202519
                     1       0.90      0.58      0.71    103386

              accuracy                           0.84    305905
             macro avg       0.86      0.77      0.80    305905
          weighted avg       0.85      0.84      0.83    305905


In [43]:  cm=confusion_matrix(y_test,y_predict3)
          sns.heatmap(cm,annot=True,fmt='d')

Out[43]:  <AxesSubplot:>
```

```
                     1       0.90      0.58      0.71    103386

              accuracy                           0.84    305905
             macro avg       0.86      0.77      0.80    305905
          weighted avg       0.85      0.84      0.83    305905


In [43]:  cm=confusion_matrix(y_test,y_predict3)
          sns.heatmap(cm,annot=True,fmt='d')

Out[43]:  <AxesSubplot:>
```

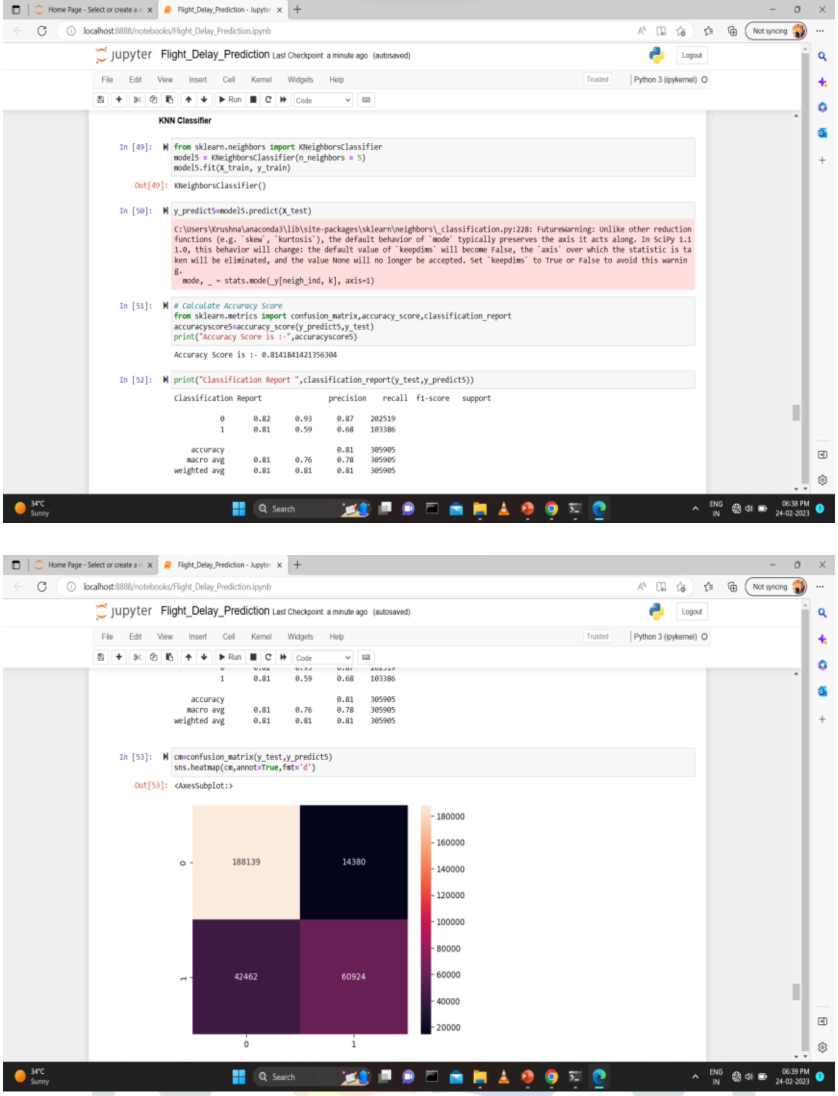
Flight_Delay_Prediction - Jupyter
localhost:8888/notebooks/Flight_Delay_Prediction.ipynb
jupyter Flight_Delay_Prediction Last Checkpoint: Last Tuesday at 1:52 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Markdown
195746
6773
43320
60066
0
1
0
1
175000
150000
125000
100000
75000
50000
25000

Fig.6 Decision Tree output

Expected outcome – 2) Decision Tree
Decision Tree output – 0.8362 = 83.62% Accuracy with Classification Report
In this Fig.6 we observed that we got 83.62% accuracy by the Decision Tree algorithm. And it shows the classification report of the output.

D. K- Nearest neighbour :

K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.



KNN Classifier

```
In [49]:  from sklearn.neighbors import KNeighborsClassifier
          model5 = KNeighborsClassifier(n_neighbors = 5)
          model5.fit(X_train, y_train)

Out[49]:  KNeighborsClassifier()

In [50]:  y_predict5=model5.predict(X_test)

          C:\Users\Krushna\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction
          functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In SciPy 1.1
          1.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statistic is ta
          ken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this warnin
          g.
            mode, _ = stats.mode(_y[neigh_ind, k], axis=1)

In [51]:  # Calculate Accuracy Score
          from sklearn.metrics import confusion_matrix,accuracy_score,classification_report
          accuracyscore5=accuracy_score(y_predict5,y_test)
          print("Accuracy Score is :-",accuracyscore5)

          Accuracy Score is :- 0.8141841421356304

In [52]:  print("Classification Report ",classification_report(y_test,y_predict5))

          Classification Report               precision    recall  f1-score   support

                     0       0.82      0.93      0.87    202519
                     1       0.81      0.59      0.68    103386

              accuracy                           0.81    305905
             macro avg       0.81      0.76      0.78    305905
          weighted avg       0.81      0.81      0.81    305905
```

```
                     1       0.81      0.59      0.68    103386

              accuracy                           0.81    305905
             macro avg       0.81      0.76      0.78    305905
          weighted avg       0.81      0.81      0.81    305905

In [53]:  cm=confusion_matrix(y_test,y_predict5)
          sns.heatmap(cm,annot=True,fmt='d')

Out[53]:  <AxesSubplot:>
```

jupyter Flight_Delay_Prediction Last Checkpoint: a minute ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
localhost:8888/notebooks/Flight_Delay_Prediction.ipynb
188139
14380
42462
60924
0
1
0
1
180000
160000
140000
120000
100000
80000
60000
40000
20000

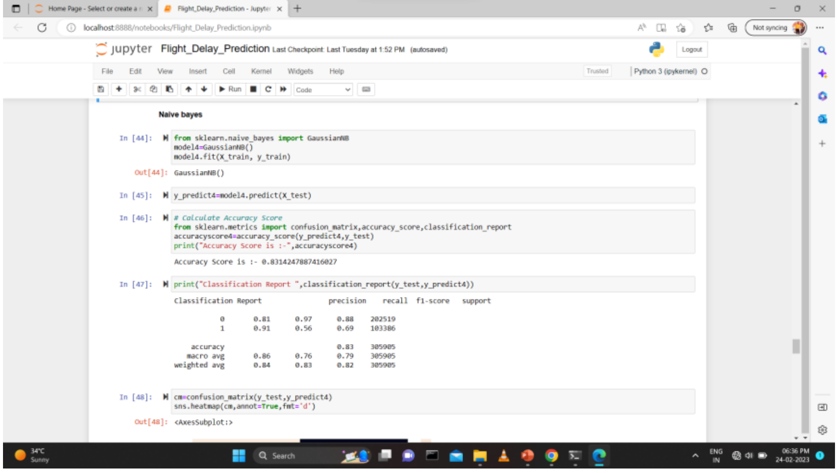
Fig.7 K-Nearest Neighbor output

Expected outcome – 5) K-Nearest Neighbor
K-Nearest Neighbor output – 0.8141 = 81.41% Accuracy with Classification Report

In this Fig.7 we observed that we got 81.41% accuracy by the K-NN algorithm. And it shows the classification report of the output.

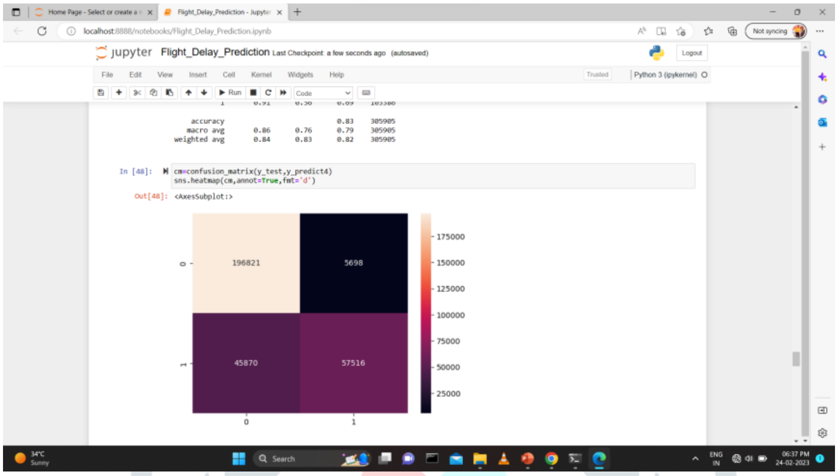
E. *Naive bayes :*

This algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.



Naive bayes

```
In [44]:  from sklearn.naive_bayes import GaussianNB
          model4=GaussianNB()
          model4.fit(X_train, y_train)

Out[44]:  GaussianNB()

In [45]:  y_predict4=model4.predict(X_test)

In [46]:  # Calculate Accuracy Score
          from sklearn.metrics import confusion_matrix,accuracy_score,classification_report
          accuracyscore4=accuracy_score(y_predict4,y_test)
          print("Accuracy Score is :-",accuracyscore4)

          Accuracy Score is :- 0.8314247887416027

In [47]:  print("Classification Report ",classification_report(y_test,y_predict4))

          Classification Report                precision    recall  f1-score   support

                     0       0.81      0.97      0.88    202519
                     1       0.91      0.56      0.69    103386

              accuracy                           0.83    305905
             macro avg       0.86      0.76      0.79    305905
          weighted avg       0.84      0.83      0.82    305905


In [48]:  cm=confusion_matrix(y_test,y_predict4)
          sns.heatmap(cm,annot=True,fmt='d')

Out[48]:  <AxesSubplot:>
```



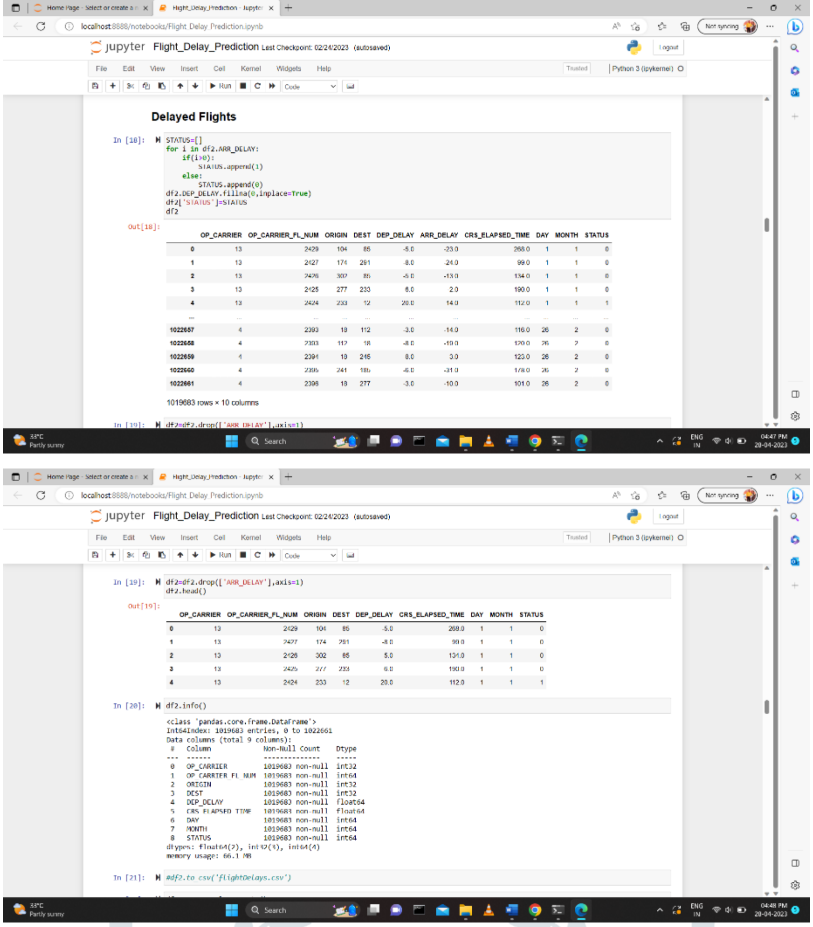
```
              accuracy                           0.83    305905
             macro avg       0.86      0.76      0.79    305905
          weighted avg       0.84      0.83      0.82    305905


In [48]:  cm=confusion_matrix(y_test,y_predict4)
          sns.heatmap(cm,annot=True,fmt='d')

Out[48]:  <AxesSubplot:>
```

Fig.8 Naïve Bayes output

Expected outcome – 6) Naive bayes
Naïve Bayes output – 0.8314 = 83.14% Accuracy with Classification Report

In this Fig.8 we observed that we got 83.14% accuracy by the naïve bayes algorithm. And it shows the classification report of the output.

It can be seen that during the prediction, the one with the best performance among the Five algorithms is the Random Forest model. For example, the accuracy value for the Random Forest is 0.8458. This value is significantly higher than that of Logistic Regression, with the second-greatest value of 0.8379 accuracy. Similar patterns of noticeable differences for performance scores of Random Forest can also be seen in the other three measures. Besides the Random Forest, the two tree-based ensemble classifiers Logistic Regression and Decision Tree, are also better performed than others. The values of measures of these two algorithms are relatively similar. The difference between their performance scores and the other Two algorithms is also significant. All algorithms perform the operation and give their particular output. It predicts the accuracy of a flight delay with its proper classification report and final status. The user sends input to the modelling dataset for prediction. inputs like flight numbers, timings, etc. If the flight is delayed, then it shows a "Delay prediction." And if the flight is on time, then it shows "Flight on Time" to the application users.



Flight_Delay_Prediction Last Checkpoint: 02/24/2023 (autosaved)

Delayed Flights

In [18]:

```
STATUS=[]
for i in df2.ARR_DELAY:
    if(i>0):
        STATUS.append(1)
    else:
        STATUS.append(0)
df2.DEP_DELAY.fillna(0,inplace=True)
df2['STATUS']=STATUS
df2
```

Out[18]:

	OP_CARRIER	OP_CARRIER_FL_NUM	ORIGIN	DEST	DEP_DELAY	ARR_DELAY	CRS_ELAPSED_TIME	DAY	MONTH	STATUS
0	13	2429	104	85	-5.0	-23.0	268.0	1	1	0
1	13	2427	174	291	-8.0	-24.0	99.0	1	1	0
2	13	2426	302	85	-5.0	-13.0	134.0	1	1	0
3	13	2425	277	233	6.0	2.0	190.0	1	1	0
4	13	2424	233	12	20.0	14.0	112.0	1	1	1
...	...	...	...	...	...	...	...	...	...	...
1022657	4	2393	18	112	-3.0	-14.0	116.0	26	2	0
1022658	4	2393	112	18	-8.0	-19.0	120.0	26	2	0
1022659	4	2394	18	245	0.0	3.0	123.0	26	2	0
1022660	4	2395	241	185	-6.0	-31.0	178.0	26	2	0
1022661	4	2396	18	277	-3.0	-10.0	101.0	26	2	0

1019683 rows × 10 columns

In [19]:

```
df2=df2.drop(['ARR_DELAY'],axis=1)
df2.head()
```

Out[19]:

	OP_CARRIER	OP_CARRIER_FL_NUM	ORIGIN	DEST	DEP_DELAY	CRS_ELAPSED_TIME	DAY	MONTH	STATUS
0	13	2429	104	85	-5.0	268.0	1	1	0
1	13	2427	174	291	-8.0	99.0	1	1	0
2	13	2426	302	85	5.0	134.0	1	1	0
3	13	2425	277	233	6.0	190.0	1	1	0
4	13	2424	233	12	20.0	112.0	1	1	1

In [20]:

```
df2.info()
```

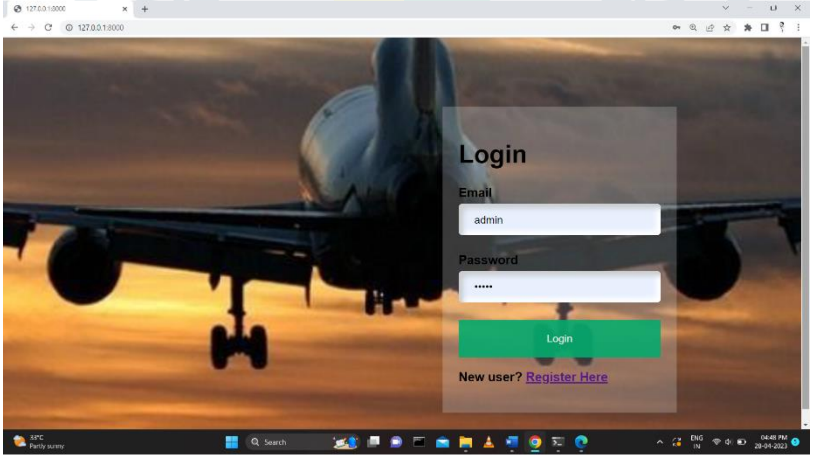
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1019683 entries, 0 to 1022661
Data columns (total 9 columns):
 #   Column             Non-Null Count    Dtype
---  ------             --------------    -----
 0   OP_CARRIER         1019683 non-null  int32
 1   OP_CARRIER_FL_NUM  1019683 non-null  int64
 2   ORIGIN             1019683 non-null  int32
 3   DEST               1019683 non-null  int32
 4   DEP_DELAY          1019683 non-null  float64
 5   CRS_ELAPSED_TIME   1019683 non-null  float64
 6   DAY                1019683 non-null  int64
 7   MONTH              1019683 non-null  int64
 8   STATUS             1019683 non-null  int64
dtypes: float64(2), int32(3), int64(4)
memory usage: 66.1 MB
```

In [21]:

```
#df2.to_csv('flightDelays.csv')
```

Fig.9 Delayed flight data

In this fig.9 It shows the delayed flight data, so from that data only arrival delay data should be finalized for the prediction process. All data should be put inside the web page for prediction process.



127.0.0.1:8000

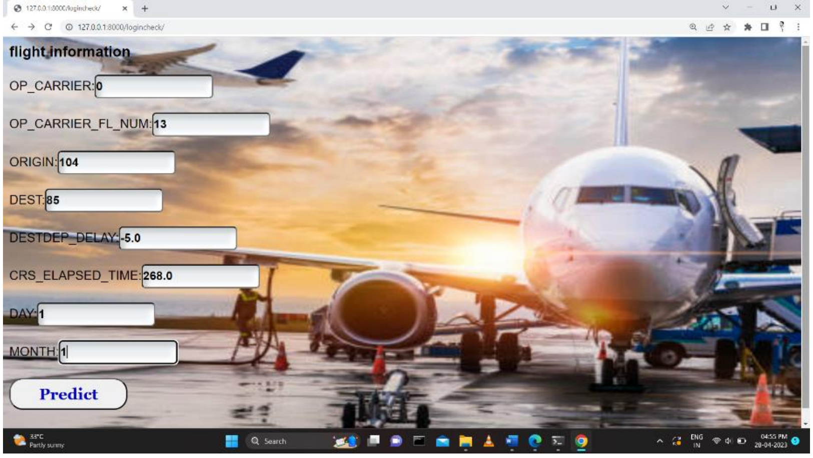
Login

Email: admin

Password: •••••

Fig.10 Login page

In this login page, user should put their login id and password so they can log into their page.



flight information

OP_CARRIER: 0

OP_CARRIER_FL_NUM: 13

ORIGIN: 104

DEST: 85

DESTDEP_DELAY: -5.0

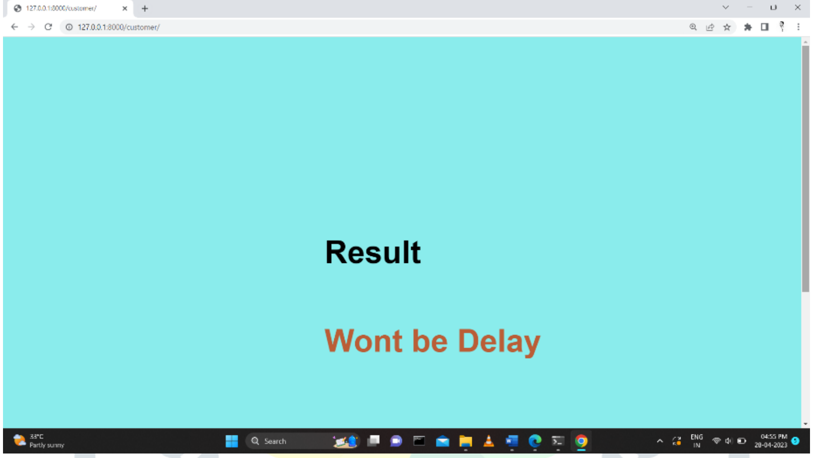
CRS_ELAPSED_TIME: 268.0

DAY: 1

MONTH: 1

Fig.11 Flight input Information

After login it shows the input methods for prediction process, so user need to put all the data from arrival delay for prediction process.



Result

Wont be Delay

Fig.12 Final Prediction

After entering all input user need to press predict button. Then it shows the final prediction of the flight if it is delayed or not.

VIII. Conclusion And Future Scope

Machine learning algorithms have been implemented to predict individual flight delays. The experimental results show that the random forest-based method can obtain good performance for the prediction task. Every algorithm performs the operations and predicts the accurate values. After getting the output result, it should be classified and shown on the accuracy graph.

Which algorithm predicts the highest and best accuracy that is going to be used as the final output for delay prediction? Users can see the flight status on the web page. Based on the analysis of their results, it is evident that the integration of multidimensional heterogeneous data, combined with the application of different techniques for feature selection and regression, can provide promising tools for inference in the domain. Machine learning algorithms that are used predict delays with high accuracy, which gives a proper result. This project can be used in other services like train, metro, and bus transportation systems.

REFERENCES

[1] JJ Rebollo and H. Balakrishnan, "Characterization and prediction of air traffic delays", *Transportation Res Part C Emerg Technol.*, vol. 44, pp. 231-41, Aug 2014.

[2] AJ Reynolds-Feighan and KJ Button, "An assessment of the capacity and congestion levels at European airports", *J Air Transp Manag.*, vol. 5, no. 3, pp. 113-34, Nov - 1999.

[3] S AhmadBeygi et al., "Analysis of the potential for delay propagation in passenger airline networks", *J Air Transp Manag.*, vol. 14, no. 5, pp. 221-36, June 2008.

[4] Y Tu, MO Ball and WS. Jank, "Estimating flight departure delay distributions-a statistical approach with long-term trend and short-term pattern", *J Am Stat Assoc.*, vol. 103, no. 481, pp. 112-25, Aug 2008.

[5] S Oza et al., "Flight delay prediction system using weighted multiple linear regression", *Int J Eng Comp Sci.*, vol. 4, no. 05, pp. 11765, Feb 2015

[6] C-Y Hsiao and M Hansen, "Air transportation network flows: equilibrium model", *Transp Res Rec.*, vol. 1915, no. 1, pp. 12-9, Feb 2005.

[7] R Britto, M Dresner and A Voltes, "The impact of flight delays on passenger demand and societal welfare", *Transp Res Part E Logist Transp Rev.*, vol. 48, no. 2, pp. 460-9, Nov 2012.

[8] S AhmadBeygi et al., "Analysis of the potential for delay propagation in passenger airline networks", *J Air Transp Manag.*, vol. 14, no. 5, pp. 221-36, Oct 2008.

[9] Bin Yu et al., "Flight delay prediction for commercial air transport: A deep learning approach", *Transportation Research Part E: Logistics and Transportation Review*, vol. 125, pp. 203-221, Jan 2019.

[10] Ding Yi, "Predicting flight delay based on multiple linear regression", *IOP Conference Series: Earth and Environmental Science*, vol. 81, no. 1, Jan 2017.

[11] Bojia Ye et al., "A Methodology for Predicting Aggregate Flight Departure Delays in Airports Based on Supervised Learning", *Sustainability*, vol. 12.7, pp. 2749, Aug 2020.

[12] Sina Khanmohammadi, Salih Tutun and Yunus Kucuk, "A new multilevel input layer artificial neural network for predicting flight delays at JFK airport", *Procedia Computer Science*, vol. 95, pp. 237-244, June 2016.

[13] Jennifer S. Raj and J. Vijitha Ananthi, "Recurrent neural networks and nonlinear prediction in support vector machines", *Journal of Soft Computing Paradigm (JSCP)*, vol. 1.01, pp. 33-40, June 2019.

[14] Vignesh Muthukumar and N. Bhalaji, "MOOCVERSITY-Deep Learning Based Dropout Prediction in MOOCs over Weeks", *Journal of Soft Computing Paradigm (JSCP)*, vol. 2.03, pp. 140-152, Apr 2020.