EXPERIMENT - 5

AIM: Regression Analysis
a) Perform Logistic regression to find out relation between variables
b) Apply regression model technique to predict the data on the above dataset.

ABOUT DATASET:

Link to our dataset:https://www.kaggle.com/datasets/divyansh22/flight-delay-prediction

This is the first part of flight delay prediction i.e. for the month of January.

This data is collected from the Bureau of Transportation Statistics, Govt. of the USA. This data is open-sourced under U.S. Govt. Works.

This dataset contains all the flights in the month of January 2019 and January 2020. There are more than 400,000 flights in the month of January itself throughout the United States.

The features were manually chosen to do a primary time series analysis.

There are several other features available on their website.

THEORY:

1. Regression

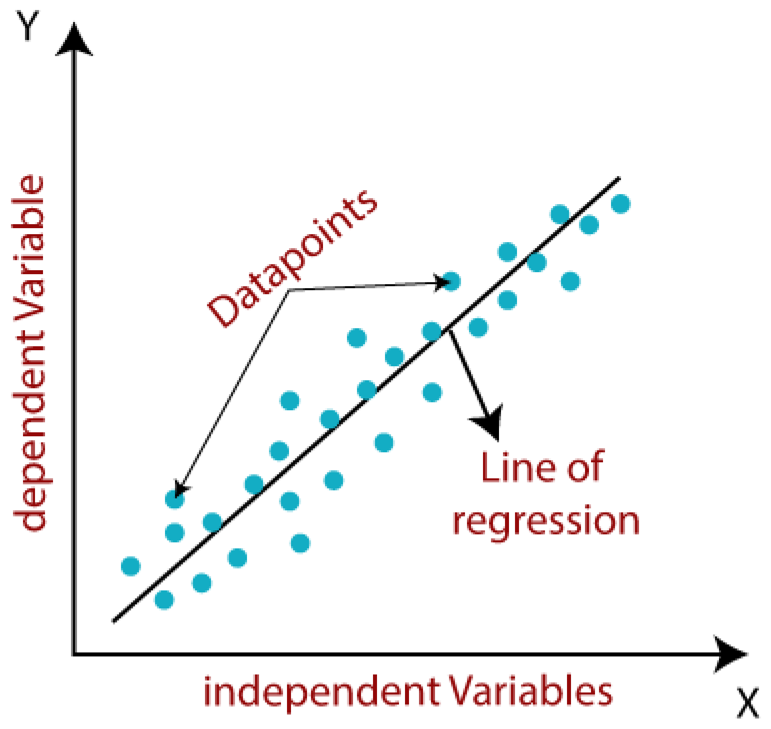
Regression analysis is an integral part of any forecasting or predictive model, so is a common method found in machine learning powered predictive analytics. Alongside classification, regression is a common use for supervised machine learning models. This approach to training models required labeled input and output training data. Machine learning regression models need to understand the relationship between features and outcome variables, so accurately labeled training data is vital.

Regression is a method for understanding the relationship between independent variables or features and a dependent variable or outcome. Outcomes can then be predicted once the relationship between independent and dependent variables has been estimated. Regression is a field of study in statistics which forms a key part of forecast models in machine learning. It's used as an approach to predict continuous outcomes in predictive modelling, so has utility in forecasting and predicting outcomes from data. Machine learning regression generally involves plotting a line of best fit through the data points. The distance between each point and the line is minimized to achieve the best fit line.

2. Linear Regression

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.



Types of Linear Regression

Linear regression can be further divided into two types of the algorithm:

- Simple Linear Regression:
  If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.
- Multiple Linear regression:
  If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

A linear line showing the relationship between the dependent and independent variables is called a regression line. A regression line can show two types of relationship:

- Positive Linear Relationship:
  If the dependent variable increases on the Y-axis and independent variable increases on X-axis, then such a relationship is termed as a Positive linear relationship.
- Negative Linear Relationship:
  If the dependent variable decreases on the Y-axis and independent variable increases on the X-axis, then such a relationship is called a negative linear relationship.

When working with linear regression, our main goal is to find the best fit line that means the error between predicted values and actual values should be minimized. The best fit line will have the least error.

The different values for weights or the coefficient of lines (a_0, a_1) gives a different line of regression, so we need to calculate the best values for a_0 and a_1 to find the best fit line, so to calculate this we use cost function.

3. Logistic Regression

- Logistic regression 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 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.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
- Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.
- Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:

Type of Logistic Regression:

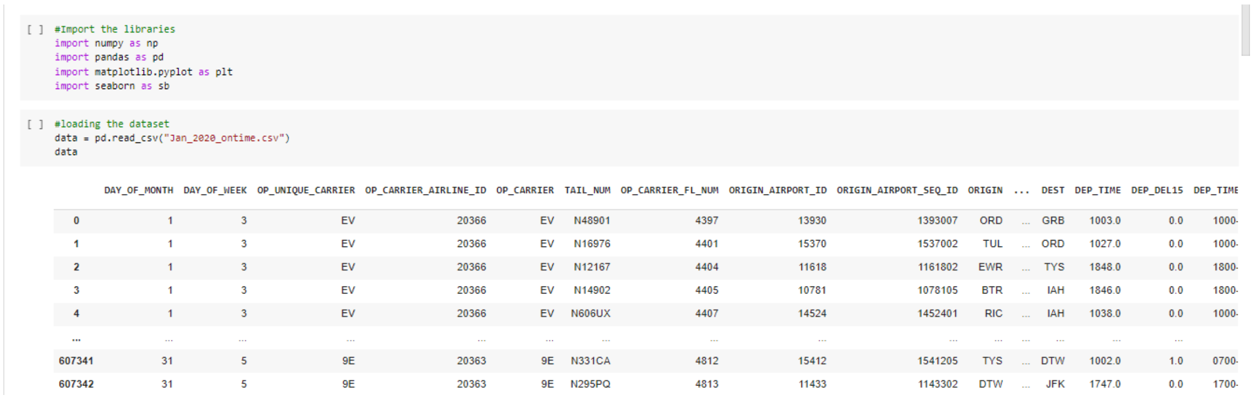
On the basis of the categories, Logistic Regression can be classified into three types:

- **Binomial**: In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.
- **Multinomial**: In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"
- **Ordinal**: In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

IMPLEMENTATION:

- **Logistic regression**

➢ Loading Data

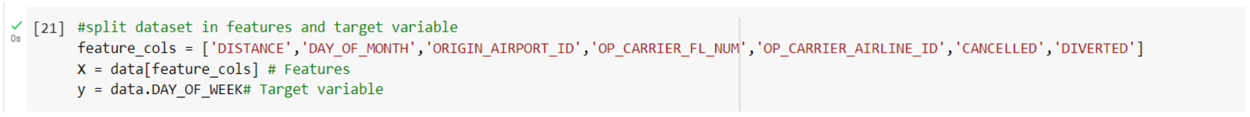


```
[ ] #Import the libraries
    import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sb
```

```
[ ] #loading the dataset
    data = pd.read_csv("Jan_2020_ontime.csv")
    data
```

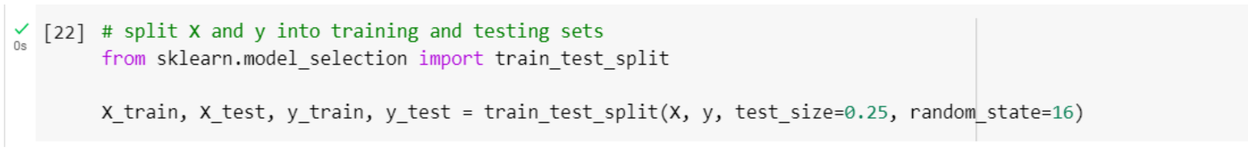
```
        DAY_OF_MONTH DAY_OF_WEEK OP_UNIQUE_CARRIER OP_CARRIER_AIRLINE_ID OP_CARRIER TAIL_NUM OP_CARRIER_FL_NUM ORIGIN_AIRPORT_ID ORIGIN_AIRPORT_SEQ_ID ORIGIN ... DEST DEP_TIME DEP_DEL15 DEP_TIME
      0            1           3                EV                 20366         EV   N48901              4397             13930               1393007    ORD ...  GRB   1003.0       0.0    1000-
      1            1           3                EV                 20366         EV   N16976              4401             15370               1537002    TUL ...  ORD   1027.0       0.0    1000-
      2            1           3                EV                 20366         EV   N12167              4404             11618               1161802    EWR ...  TYS   1848.0       0.0    1800-
      3            1           3                EV                 20366         EV   N14902              4405             10781               1078105    BTR ...  IAH   1846.0       0.0    1800-
      4            1           3                EV                 20366         EV   N606UX              4407             14524               1452401    RIC ...  IAH   1038.0       0.0    1000-
    ...          ...         ...               ...                   ...        ...      ...               ...               ...                   ...    ... ...  ...      ...       ...
 607341           31           5                9E                 20363         9E   N331CA              4812             15412               1541205    TYS ...  DTW   1002.0       1.0    0700-
 607342           31           5                9E                 20363         9E   N295PQ              4813             11433               1143302    DTW ...  JFK   1747.0       0.0    1700-
```

➢ Selecting Feature



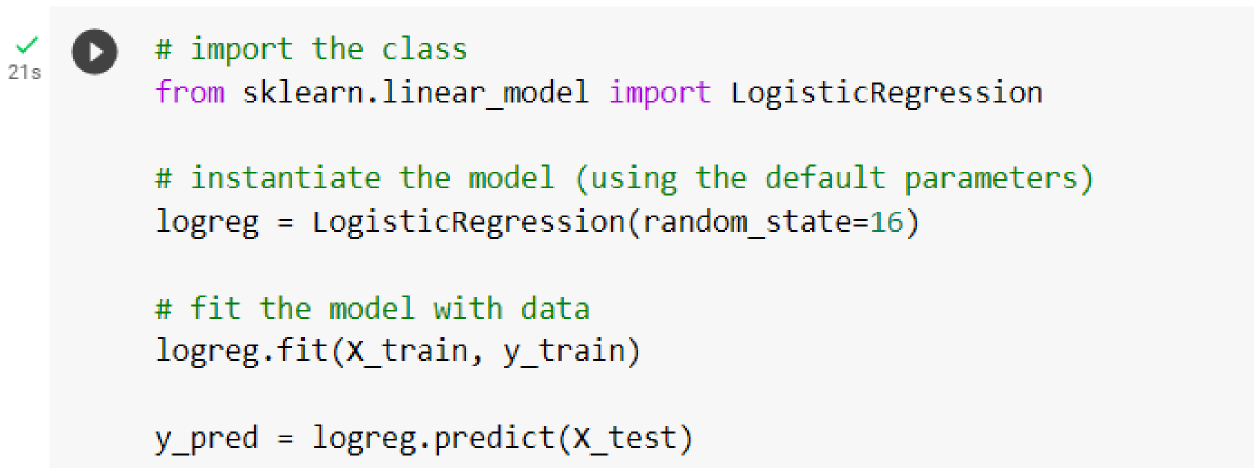
```
[21] #split dataset in features and target variable
     feature_cols = ['DISTANCE','DAY_OF_MONTH','ORIGIN_AIRPORT_ID','OP_CARRIER_FL_NUM','OP_CARRIER_AIRLINE_ID','CANCELLED','DIVERTED']
     X = data[feature_cols] # Features
     y = data.DAY_OF_WEEK# Target variable
```

➢ Splitting Data



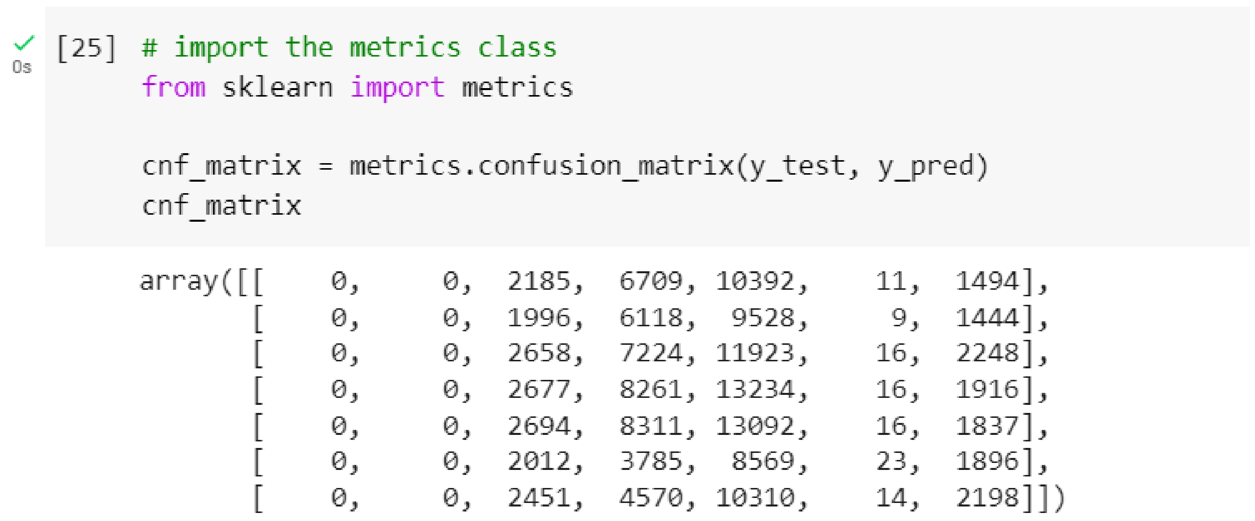
```
[22] # split X and y into training and testing sets
     from sklearn.model_selection import train_test_split

     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=16)
```

➢ Model Development and Prediction



```
# import the class
from sklearn.linear_model import LogisticRegression

# instantiate the model (using the default parameters)
logreg = LogisticRegression(random_state=16)

# fit the model with data
logreg.fit(X_train, y_train)

y_pred = logreg.predict(X_test)
```

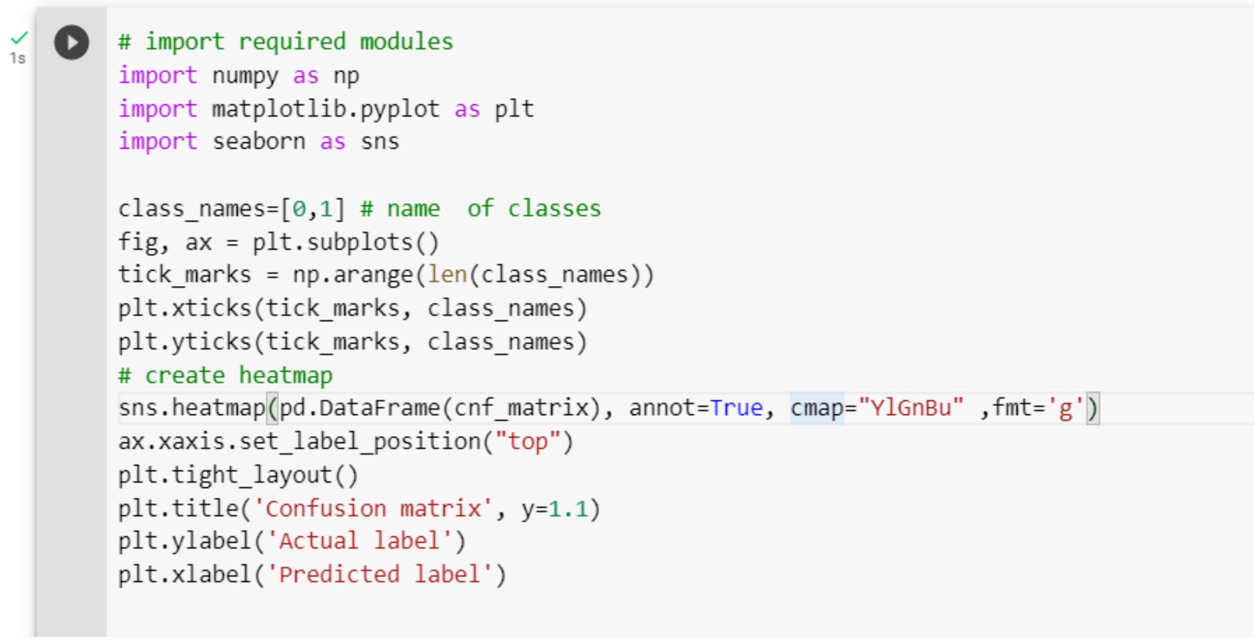
➢ Model Evaluation using Confusion Matrix



```
[25] # import the metrics class
     from sklearn import metrics

     cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
     cnf_matrix
```

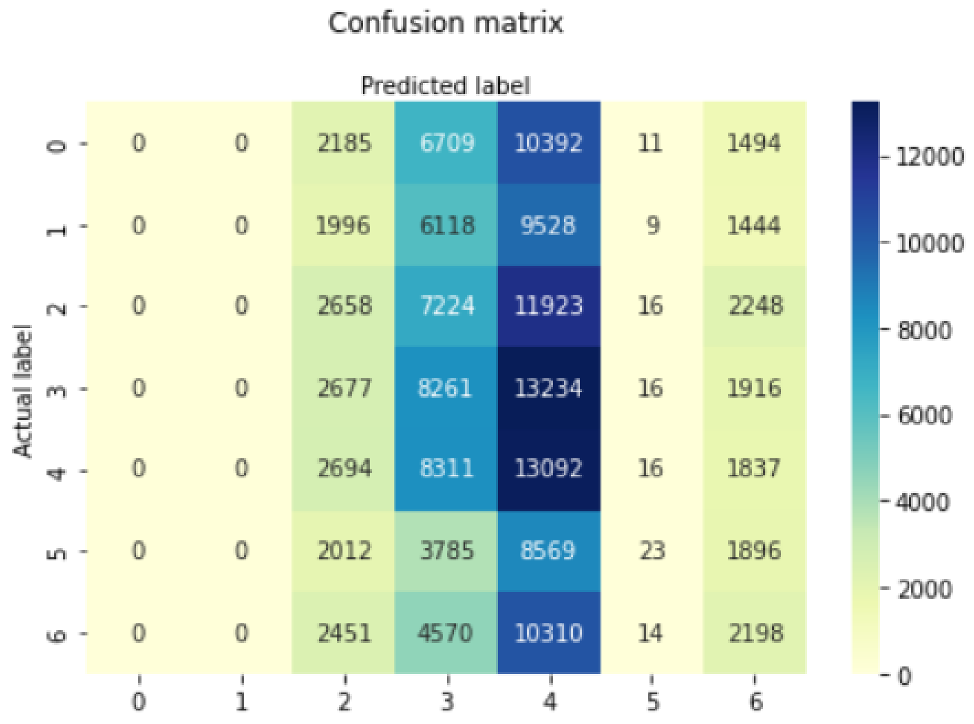
```
array([[    0,     0,  2185,  6709, 10392,    11,  1494],
       [    0,     0,  1996,  6118,  9528,     9,  1444],
       [    0,     0,  2658,  7224, 11923,    16,  2248],
       [    0,     0,  2677,  8261, 13234,    16,  1916],
       [    0,     0,  2694,  8311, 13092,    16,  1837],
       [    0,     0,  2012,  3785,  8569,    23,  1896],
       [    0,     0,  2451,  4570, 10310,    14,  2198]])
```

➢ Visualizing Confusion Matrix using Heatmap



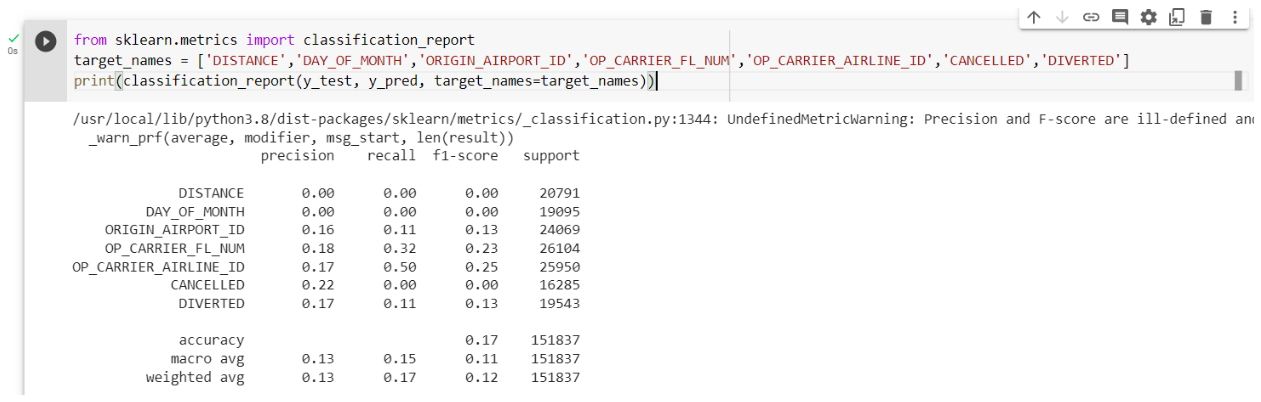
```
# import required modules
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

class_names=[0,1] # name  of classes
fig, ax = plt.subplots()
tick_marks = np.arange(len(class_names))
plt.xticks(tick_marks, class_names)
plt.yticks(tick_marks, class_names)
# create heatmap
sns.heatmap(pd.DataFrame(cnf_matrix), annot=True, cmap="YlGnBu" ,fmt='g')
ax.xaxis.set_label_position("top")
plt.tight_layout()
plt.title('Confusion matrix', y=1.1)
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
```

```
Text(0.5, 257.44, 'Predicted label')
```



	0	1	2	3	4	5	6
0	0	0	2185	6709	10392	11	1494
1	0	0	1996	6118	9528	9	1444
2	0	0	2658	7224	11923	16	2248
3	0	0	2677	8261	13234	16	1916
4	0	0	2694	8311	13092	16	1837
5	0	0	2012	3785	8569	23	1896
6	0	0	2451	4570	10310	14	2198

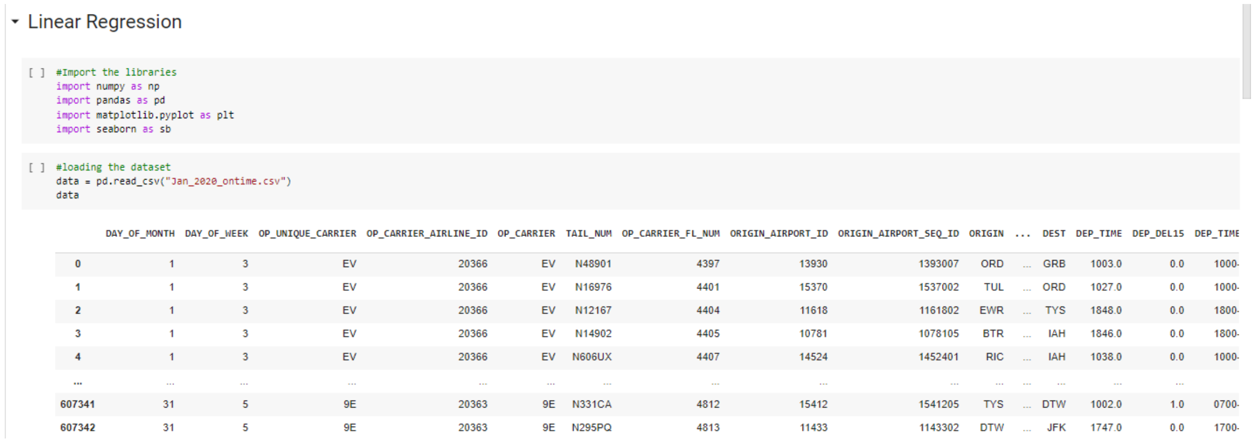
➢ Confusion Matrix Evaluation Metrics



```
from sklearn.metrics import classification_report
target_names = ['DISTANCE','DAY_OF_MONTH','ORIGIN_AIRPORT_ID','OP_CARRIER_FL_NUM','OP_CARRIER_AIRLINE_ID','CANCELLED','DIVERTED']
print(classification_report(y_test, y_pred, target_names=target_names))
```

```
/usr/local/lib/python3.8/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined an
  _warn_prf(average, modifier, msg_start, len(result))
                       precision    recall  f1-score   support

             DISTANCE       0.00      0.00      0.00     20791
         DAY_OF_MONTH       0.00      0.00      0.00     19095
    ORIGIN_AIRPORT_ID       0.16      0.11      0.13     24069
    OP_CARRIER_FL_NUM       0.18      0.32      0.23     26104
OP_CARRIER_AIRLINE_ID       0.17      0.50      0.25     25950
            CANCELLED       0.22      0.00      0.00     16285
             DIVERTED       0.17      0.11      0.13     19543

             accuracy                           0.17    151837
            macro avg       0.13      0.15      0.11    151837
         weighted avg       0.13      0.17      0.12    151837
```

- **Linear regression**



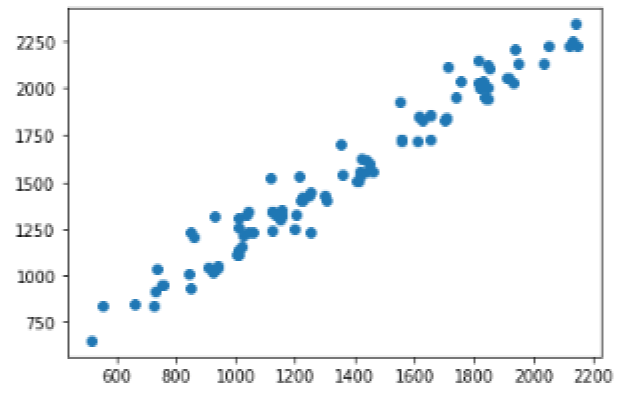
Linear Regression

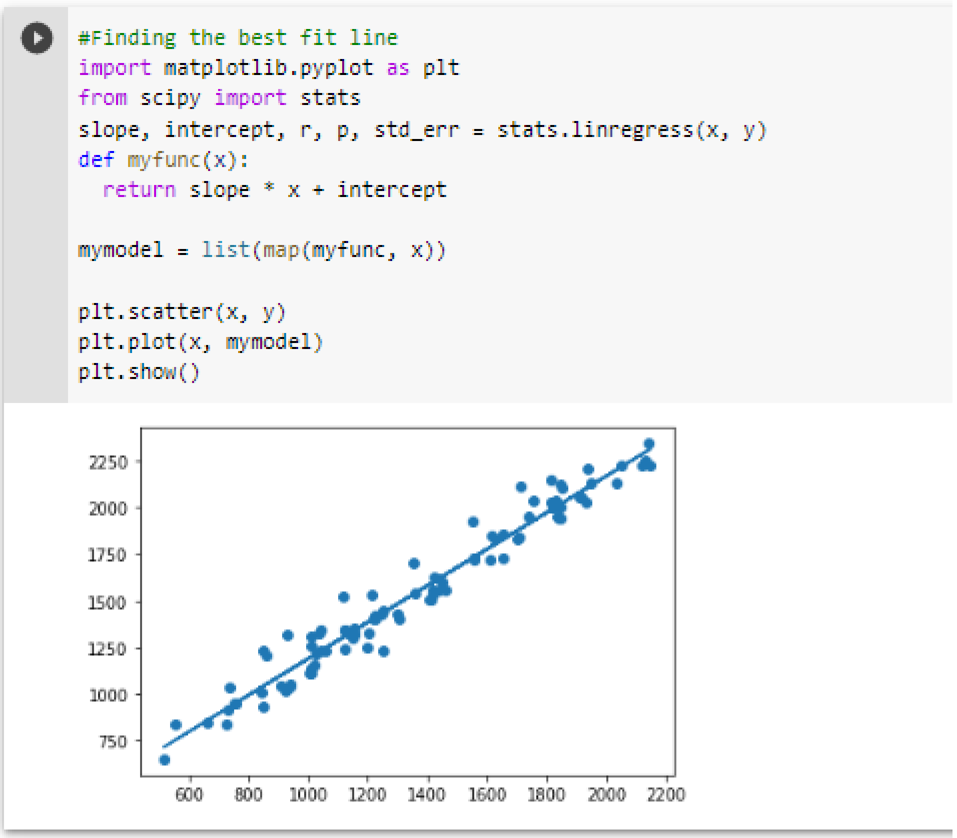
```
[ ] #Import the libraries
    import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sb
```

```
[ ] #loading the dataset
    data = pd.read_csv("Jan_2020_ontime.csv")
    data
```

	DAY_OF_MONTH	DAY_OF_WEEK	OP_UNIQUE_CARRIER	OP_CARRIER_AIRLINE_ID	OP_CARRIER	TAIL_NUM	OP_CARRIER_FL_NUM	ORIGIN_AIRPORT_ID	ORIGIN_AIRPORT_SEQ_ID	ORIGIN	...	DEST	DEP_TIME	DEP_DEL15	DEP_TIME
0	1	3	EV	20366	EV	N48901	4397	13930	1393007	ORD	...	GRB	1003.0	0.0	1000-
1	1	3	EV	20366	EV	N16976	4401	15370	1537002	TUL	...	ORD	1027.0	0.0	1000-
2	1	3	EV	20366	EV	N12167	4404	11618	1161802	EWR	...	TYS	1848.0	0.0	1800-
3	1	3	EV	20366	EV	N14902	4405	10781	1078105	BTR	...	IAH	1846.0	0.0	1800-
4	1	3	EV	20366	EV	N606UX	4407	14524	1452401	RIC	...	IAH	1038.0	0.0	1000-
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
607341	31	5	9E	20363	9E	N331CA	4812	15412	1541205	TYS	...	DTW	1002.0	1.0	0700-
607342	31	5	9E	20363	9E	N295PQ	4813	11433	1143302	DTW	...	JFK	1747.0	0.0	1700-

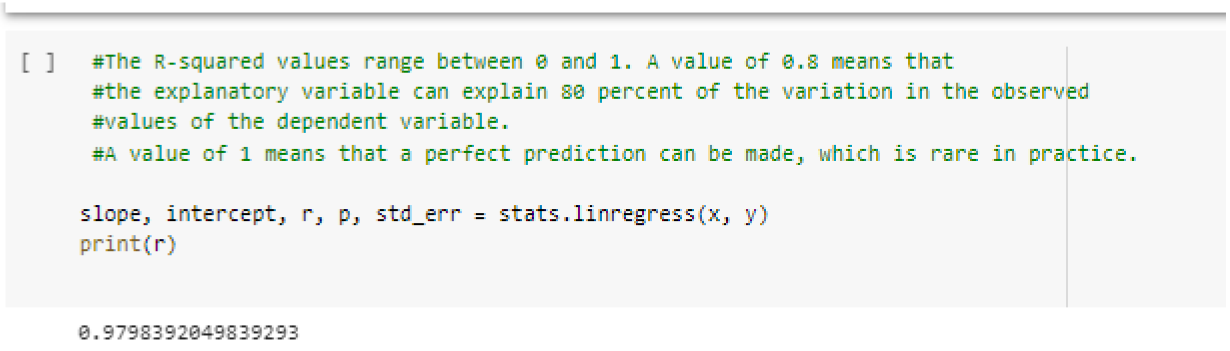
```
x = data['DEP_TIME'].head(100)
y = data['ARR_TIME'].head(100)
plt.scatter(x, y)
plt.show()
```





```
#Finding the best fit line
import matplotlib.pyplot as plt
from scipy import stats
slope, intercept, r, p, std_err = stats.linregress(x, y)
def myfunc(x):
  return slope * x + intercept

mymodel = list(map(myfunc, x))

plt.scatter(x, y)
plt.plot(x, mymodel)
plt.show()
```

➢ Value of R is 0.97 which means 97%. Thus our prediction result would be 97% accurate.

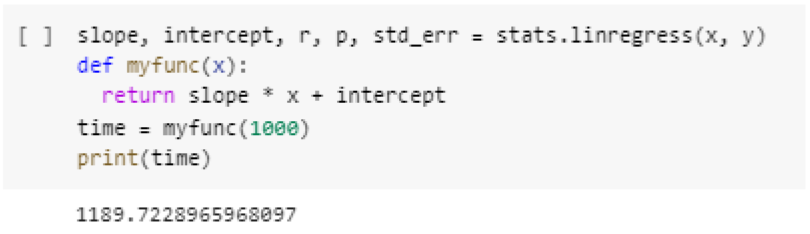


```
[ ]  #The R-squared values range between 0 and 1. A value of 0.8 means that
     #the explanatory variable can explain 80 percent of the variation in the observed
     #values of the dependent variable.
     #A value of 1 means that a perfect prediction can be made, which is rare in practice.

    slope, intercept, r, p, std_err = stats.linregress(x, y)
    print(r)
```

```
0.9798392049839293
```

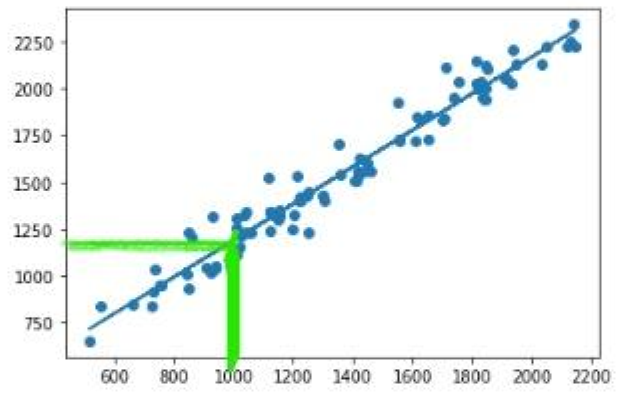
- **Predicting Values using linear regression**

- The value of given input is arrival time ,that is , 1000.
- We are predicting the value of Departure time for the flight which arrived at arrival time =1000.



```
[ ] slope, intercept, r, p, std_err = stats.linregress(x, y)
    def myfunc(x):
      return slope * x + intercept
    time = myfunc(1000)
    print(time)

    1189.7228965968097
```

- Thus the predicted departure time is 1189.



<u>CONCLUSION:</u> Thus we performed logistic and linear regression and studied the relation between two variables. We used linear regression to predict values accordingly. We used the flight dataset in this experiment to predict the departure time of the flight which just arrived. Thus we successfully implemented the logistic regression and linear regression along with the prediction.