S. B. JAIN INSTITUTE OF TECHNOLOGY, MANAGEMENT & RESEARCH, NAGPUR.

Practical No. 1 (b)

Aim: Demonstrate the Problem related to Logistics Regression in Data Analytics.

Name of Student : Shrutika Pradeep Bagdi

Roll No. : CS22130

Semester/Year : 6th /3rd

Academic Session : 2024 - 2025

Date of Performance :

Date of Submission :

AIM: Demonstrate the Problem related to Logistics Regression in Data Analytics.

OBJECTIVE/EXPECTED LEARNING OUTCOME:

The objectives and expected learning outcome of this practical are:

- Understand the use of odds, odds ratios and transformations in logistic regression.
- Logistic regression is a statistical analysis method to predict a binary outcome
- To measure the relationship between a categorical dependent variable and one or more independent variables (usually continuous) by plotting the dependent variables' probability scores.
- Able to calculate both simple and multiple regression models. You will learn how to assess the model's "fit", test model assumptions, and transform predictor and response variables to improve outcomes.

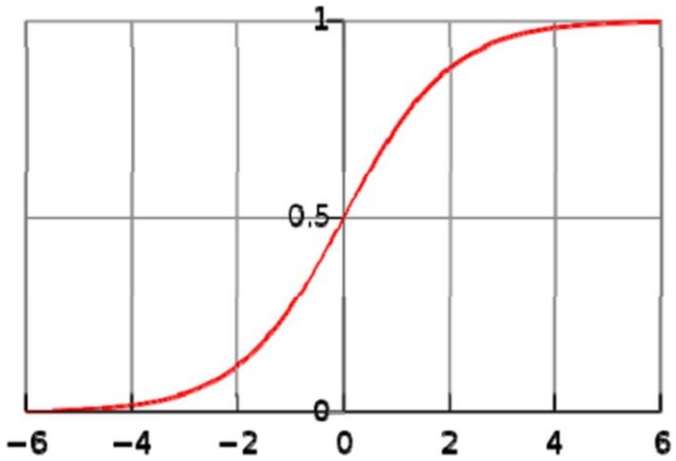
THEORY:

Logistic regression

- Name is somewhat misleading. Really a technique for classification, not regression. technique for classification, not regression.
- "Regression" comes from fact that we fit a linear model to the feature space
- Involves a more probabilistic view of classification

$$p = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}} \qquad \text{logistic function}$$

Standard logistic function



Using a logistic regression model

Can interpret prediction from a logistic regression model as:

- A probability of class membership –
- A class assignment by applying threshold to A class assignment, by applying threshold to probability.
- Threshold represents decision boundary in feature space

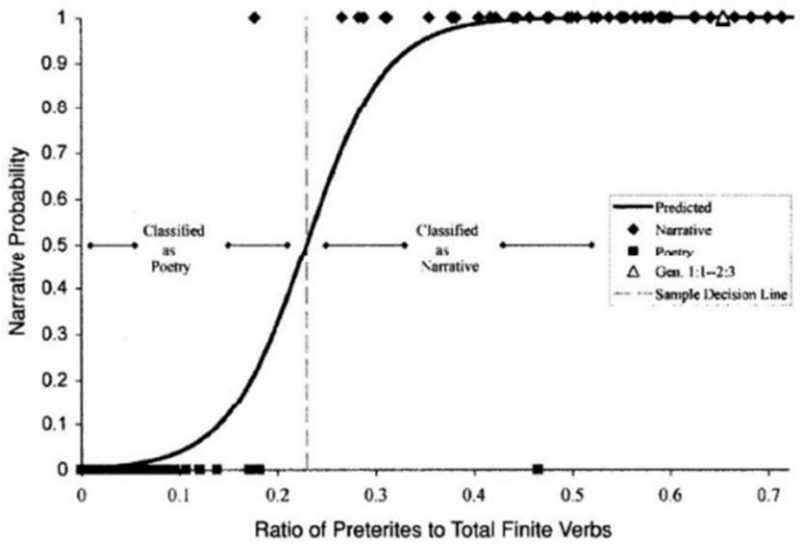
Training a logistic regression model

Need to optimize β so the model gives the best possible reproduction of training set labels possible reproduction of training set labels

– Usually done by numerical approximation of maximum likelihood

– On really large datasets, may use stochastic gradient descent

Logistic regression in one dimension

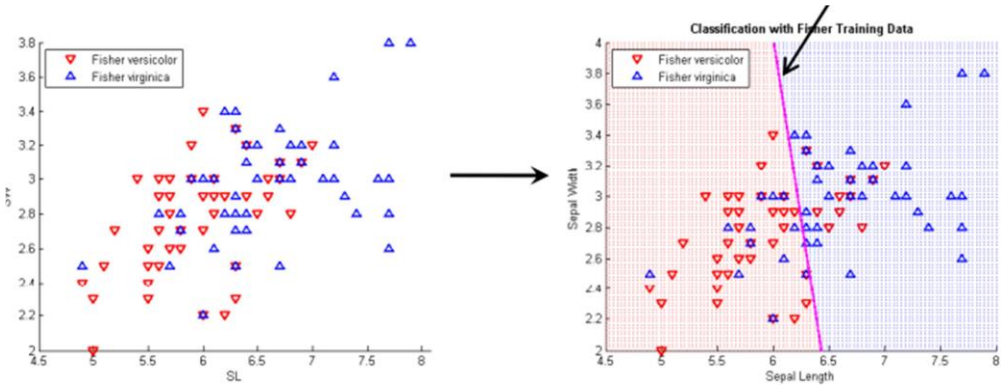


Logistic regression in two dimensions

Subset of Fisher iris dataset

– Two classes –

First two columns (SL, SW)



STEPS:

1) Split the data set into training and test sets.

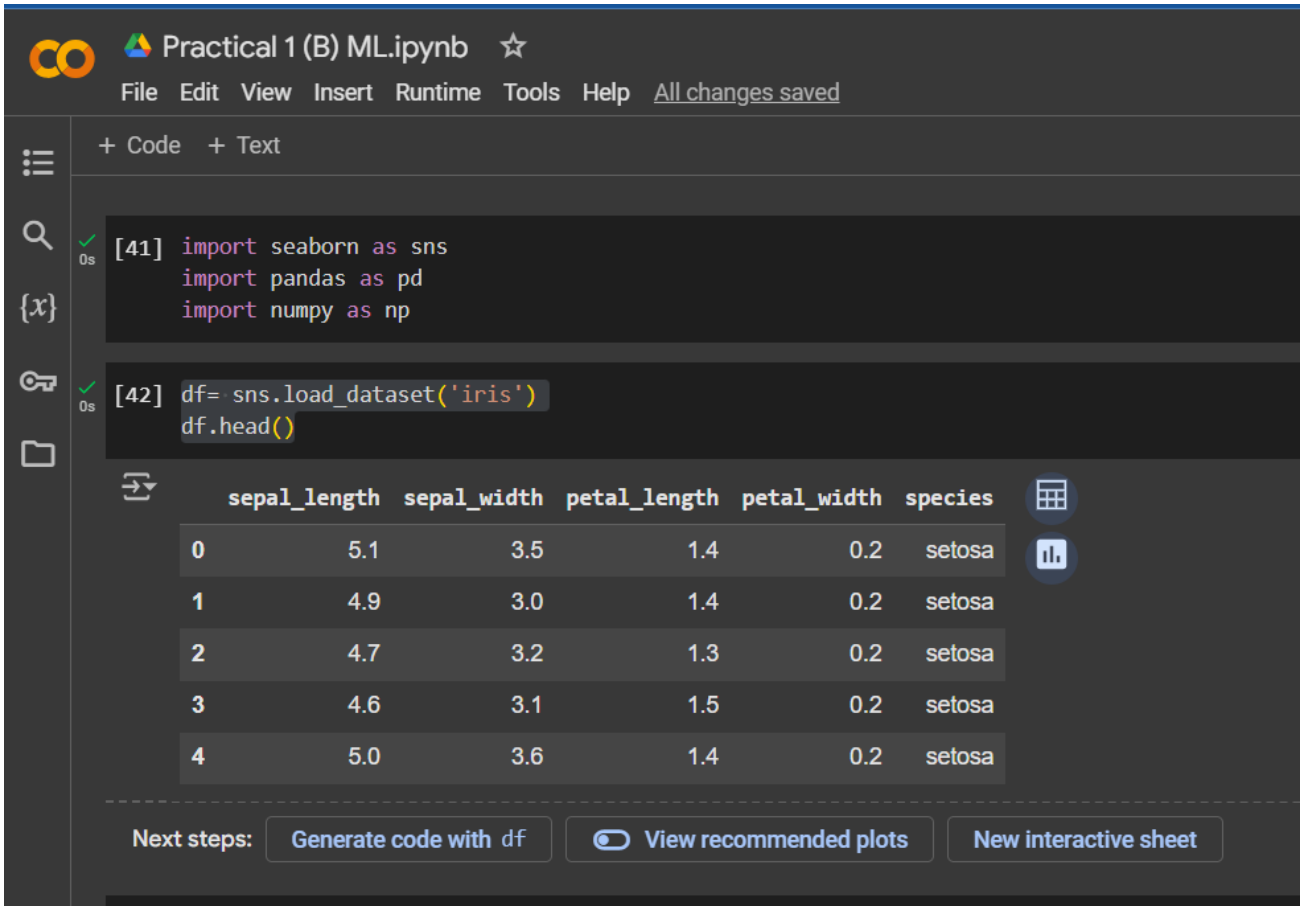
2) Fit logistic regression model on training set.

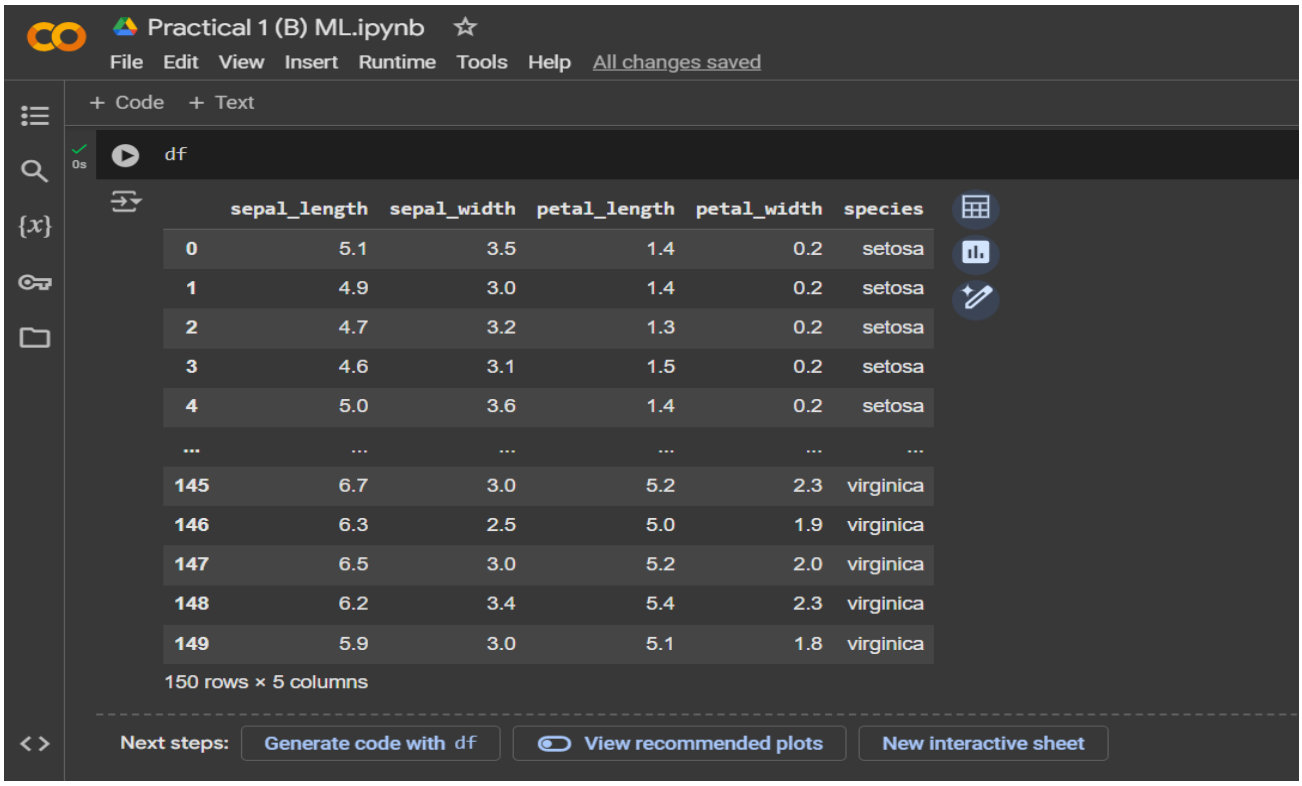
3) Prepare Confusion matrix.

4) Compute model accuracy, precision, recall.

PROGRAM CODE:

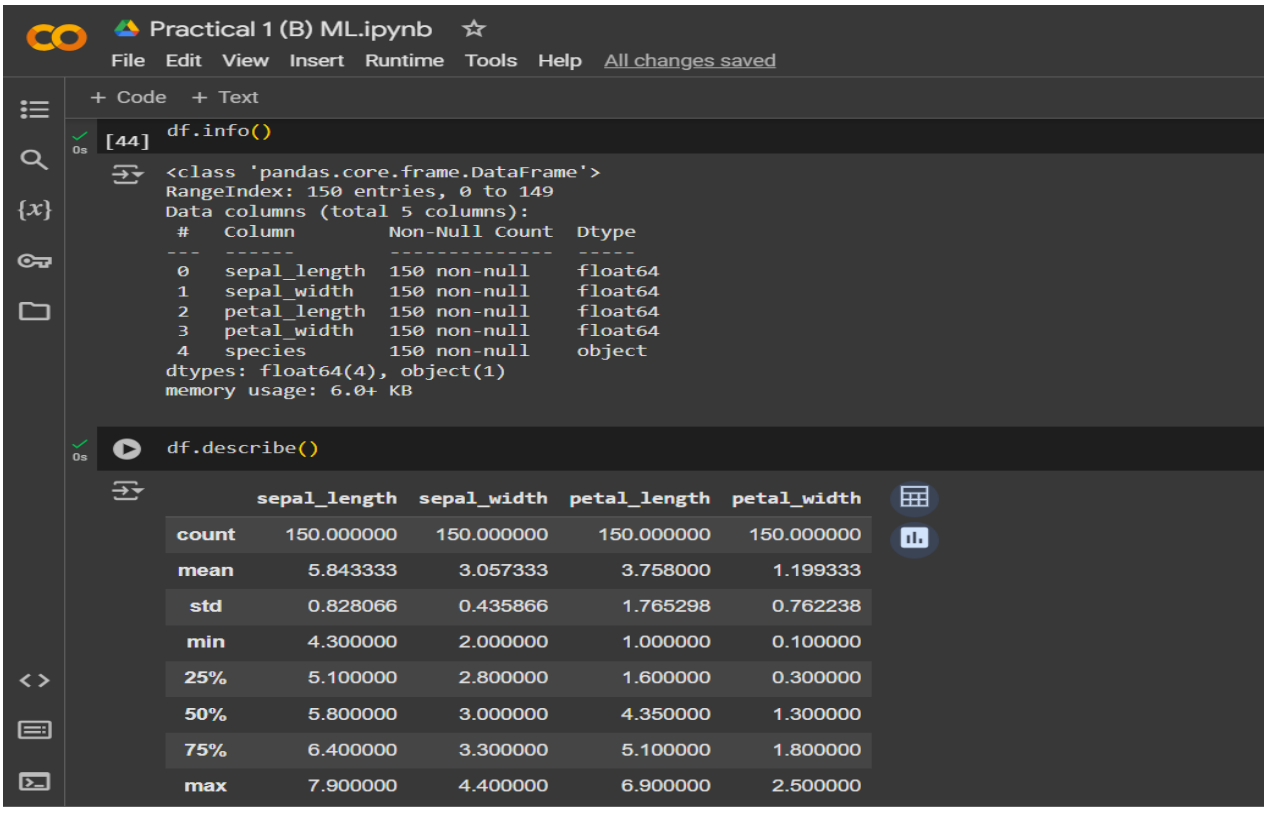
OUTPUT (SCREENSHOT):





	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns



	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

Practical 1 (B) ML.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[46] df.shape
```

```
(150, 5)
```

```
[47] df.tail()
```

	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

```
[48] #it shows the categories
     df['species'].unique()
```

```
array(['setosa', 'versicolor', 'virginica'], dtype=object)
```

Practical 1 (B) ML.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[49] #to find null
     df.isnull().sum()
```

	0
sepal_length	0
sepal_width	0
petal_length	0
petal_width	0
species	0

dtype: int64

```
[50] #to remove the species setosa from df and store in df
     df=df[df['species']!='setosa']
```

```
[52] df.shape
```

```
(100, 5)
```

Practical 1 (B) ML.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[53] df['species']=df['species'].map({'versicolor':0, 'virginica':1})
```

```
<ipython-input-53-cb6bebc20fa7>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['species']=df['species'].map({'versicolor':0, 'virginica':1})
```

```
[54] df.head()
```

	sepal_length	sepal_width	petal_length	petal_width	species
50	7.0	3.2	4.7	1.4	0
51	6.4	3.2	4.5	1.5	0
52	6.9	3.1	4.9	1.5	0
53	5.5	2.3	4.0	1.3	0
54	6.5	2.8	4.6	1.5	0

Next steps: Generate code with df | View recommended plots | New interactive sheet

Practical 1 (B) ML.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[55] df.tail()
```

	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	1
146	6.3	2.5	5.0	1.9	1
147	6.5	3.0	5.2	2.0	1
148	6.2	3.4	5.4	2.3	1
149	5.9	3.0	5.1	1.8	1

```
X=df.iloc[:,:-1]
print(X)
```

```
     sepal_length  sepal_width  petal_length  petal_width
50            7.0          3.2           4.7          1.4
51            6.4          3.2           4.5          1.5
52            6.9          3.1           4.9          1.5
53            5.5          2.3           4.0          1.3
54            6.5          2.8           4.6          1.5
..            ...          ...           ...          ...
145           6.7          3.0           5.2          2.3
146           6.3          2.5           5.0          1.9
147           6.5          3.0           5.2          2.0
148           6.2          3.4           5.4          2.3
149           5.9          3.0           5.1          1.8

[100 rows x 4 columns]
```

Practical 1 (B) ML.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[57] y=df.iloc[:,-1]
     print(y)
```

```
50     0
51     0
52     0
53     0
54     0
      ..
145    1
146    1
147    1
148    1
149    1
Name: species, Length: 100, dtype: int64
```

```
[58] from sklearn.model_selection import train_test_split
     X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.7, test_size = 0.25, random_state = 42)
```

```
X_train
```

	sepal_length	sepal_width	petal_length	petal_width
65	6.7	3.1	4.4	1.4
90	5.5	2.6	4.4	1.2
146	6.3	2.5	5.0	1.9
59	5.2	2.7	3.9	1.4
122	7.7	2.8	6.7	2.0

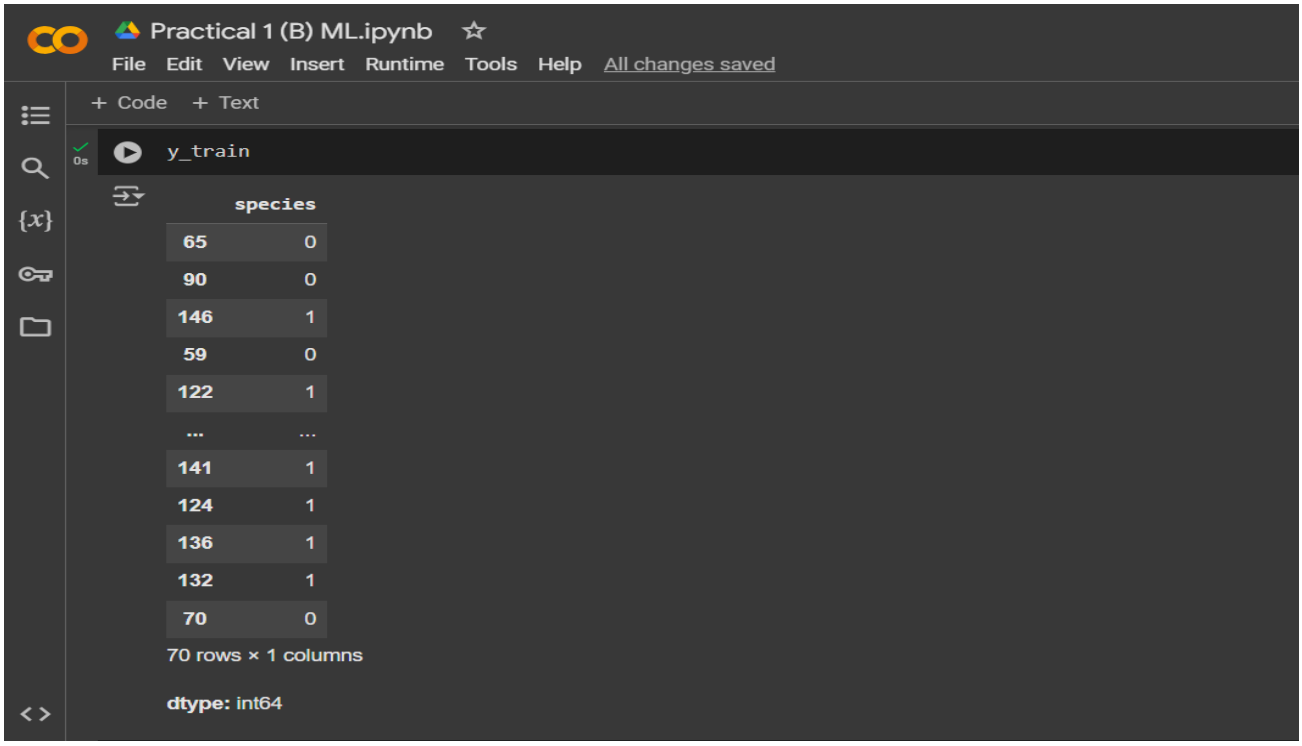
Practical 1 (B) ML.ipynb

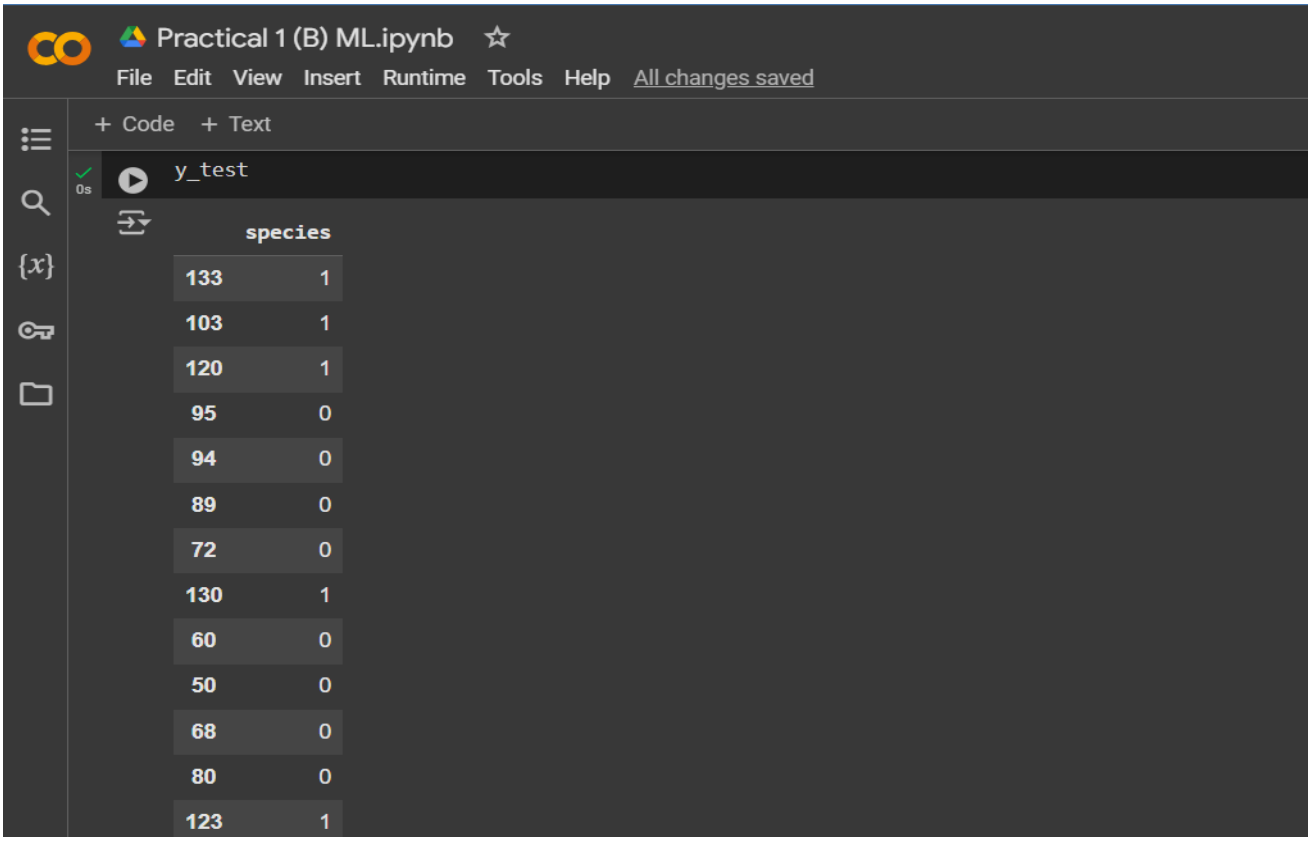
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
X_test
```

	sepal_length	sepal_width	petal_length	petal_width
133	6.3	2.8	5.1	1.5
103	6.3	2.9	5.6	1.8
120	6.9	3.2	5.7	2.3
95	5.7	3.0	4.2	1.2
94	5.6	2.7	4.2	1.3
89	5.5	2.5	4.0	1.3
72	6.3	2.5	4.9	1.5
130	7.4	2.8	6.1	1.9
60	5.0	2.0	3.5	1.0
50	7.0	3.2	4.7	1.4
68	6.2	2.2	4.5	1.5
80	5.5	2.4	3.8	1.1


Practical 1 (B) ML.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
y_train
species
65 0
90 0
146 1
59 0
122 1
... ...
141 1
124 1
136 1
132 1
70 0
70 rows × 1 columns
dtype: int64


Practical 1 (B) ML.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
y_test
species
133 1
103 1
120 1
95 0
94 0
89 0
72 0
130 1
60 0
50 0
68 0
80 0
123 1

Practical 1 (B) ML.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[63] from sklearn.linear_model import LogisticRegression
     classifier=LogisticRegression()
```

```
[64] classifier.fit(X_train, y_train)
```

```
LogisticRegression()
```

```
[65] #prediction
     y_pred = classifier.predict(X_test)
```

```
[ ] y_pred
```

```
array([1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0,
       0, 0, 1])
```

```
[66] #Accuracy Score
     from sklearn.metrics import accuracy_score, classification_report
```

```
score = accuracy_score(y_pred, y_test)
print(score)
```

```
0.88
```

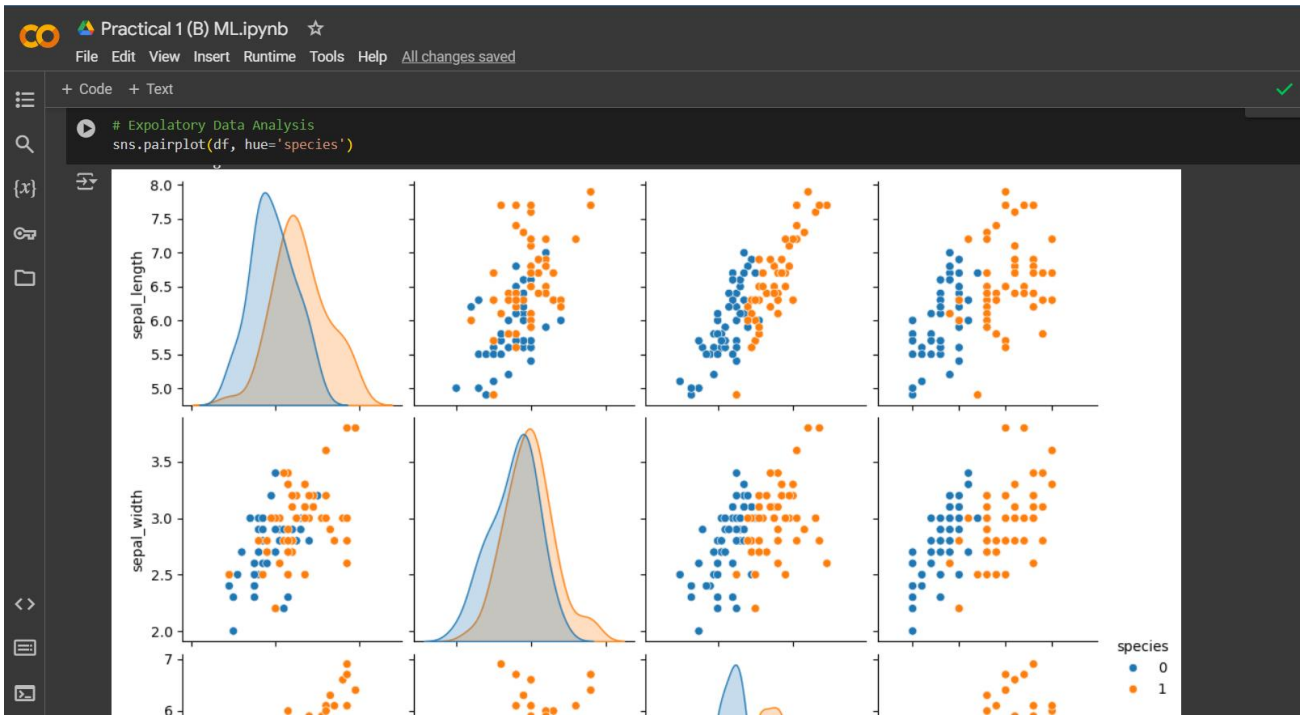
Practical 1 (B) ML.ipynb

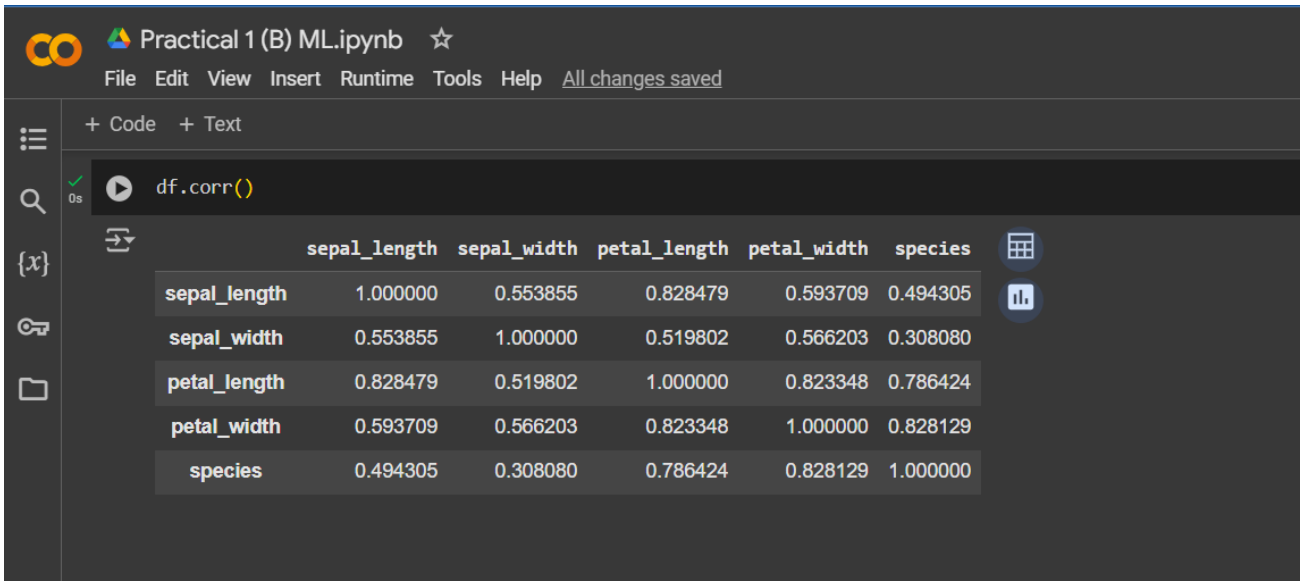
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
# For precision, Recall, F1 Score & Accuracy
print(classification_report(y_pred, y_test))
```

	precision	recall	f1-score	support
0	0.93	0.87	0.90	15
1	0.82	0.90	0.86	10
accuracy			0.88	25
macro avg	0.87	0.88	0.88	25
weighted avg	0.88	0.88	0.88	25





	sepal_length	sepal_width	petal_length	petal_width	species
sepal_length	1.000000	0.553855	0.828479	0.593709	0.494305
sepal_width	0.553855	1.000000	0.519802	0.566203	0.308080
petal_length	0.828479	0.519802	1.000000	0.823348	0.786424
petal_width	0.593709	0.566203	0.823348	1.000000	0.828129
species	0.494305	0.308080	0.786424	0.828129	1.000000

CONCLUSION:

DISCUSSION AND VIVA VOCE:

- What is the basic principle of logistic regression?
- What is the practical application of logistic regression?
- What is logistic regression used for in machine learning?
- What are the analytical challenges during model development?
- What are the difference between linear regression and logistic?

REFERENCE:

- https://www.techtarget.com/searchitoperations/definition/virtual-machine-VM
- https://medium.com/analytics-vidhya/20-interview-questions-on-linear-regression-and-logistic-regression-ef4d341d2805
- www.cs.sfu.ca/~han/MachineLearnig.html