

Exp:-4 :- To understand linear regression and its implementation in python programming language.

Program :-

```
import numpy as np
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt

X = np.array([1, 2, 3, 4, 5]).reshape(-1, 1)
y = np.array([2, 4, 5, 4, 5])

model = LinearRegression()
model.fit(X, y)

X_test = np.array([6, 7, 8]).reshape(-1, 1)
predictions = model.predict(X_test)

plt.scatter(X, y, label='data')
plt.plot(X_test, predictions, color='red', label='line of best fit')
plt.xlabel('X')
plt.ylabel('y')
plt.legend()
plt.show()
```

exp:-3

To study and implement range, variance, standard deviation and interquartile range in python programming language.

→

```
import statistics as st
```

```
data = [12, 15, 18, 22, 30, 35, 40, 45, 50, 60]
```

```
① data_range = max(data) - min(data)
print(data_range)
```



2)

```
data-mean = statistics.mean(data)
```

```
print(data-mean)
```

```
data-variance = statistics.variance(data)
```

```
print(variance)
```

3)

```
data-std dev = statistics.stdev(data)
```

```
print(data-std dev)
```

4)

```
import numpy as np  
data.sort()
```

```
lower-half = data[:len(data)//2]
```

```
median-lower-half = np.median(lower-half)
```

```
upper-half = data[len(data)//2:]
```

```
median-upper-half = np.median(upper-half)
```

```
iqr = median-upper-half - median-lower-half
```

```
print(iqr)
```

---



exp-5

Asm:- To understand multilinear regression & its implementation in the python programming language.

Program:-

```
import matplotlib as plt
import numpy as np
from sklearn import datasets, linear_model, metrics
from sklearn.model_selection import train_test_split
boston = datasets.load_boston, boston
boston = datasets.load_boston(return_X_y=False)

X = boston.data
y = boston.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)

reg = linear_model.LinearRegression()

reg.fit(X_train, y_train)

print('X - y - test')
```

```
plt.style.use('fivethingsight')
```

```
plt.scatter(reg.predict(X_train), y_train, color='green',
```

```
s=10, label='Train data')
```

```
plt.scatter
```

```
1
```

```
1
```

```
1
```



exp-6

Aim:- To understand Logistic regression and its implementation in python programming language.

Software used :- Jupyter Notebook.

Program:-

```
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix,
classification_report, accuracy_score
```

~~data = pd.read\_csv('your dataset.csv')~~

```
np.random.seed(0)
```

```
X = np.random.randn(100, 2)
```

```
Y = (X[:, 0] + X[:, 1] > 0).astype(int)
```

```
X_train, X_test, Y_train, Y_test = train_test_split
```

```
(X, Y, test_size=0.2, random_state=42)
```

```
model = LogisticRegression()
```

```
model.fit(X_train, Y_train)
```

```
Y_pred = model.predict(X_test)
```

```
accuracy = accuracy_score(Y_test, Y_pred)
```

```
report = classification_report(Y_test, Y_pred)
```

```
conf_matrix = confusion_matrix(Y_test, Y_pred)
```



```
y_pred = nb_classifier.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
accuracy
```

Exp-7

Aim :- To understand Decision Tree and its implementation in Python programming language.

Program: ①

```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
```

②

```
iris = datasets.load_iris()
X = iris.data
y = iris.target
```

③

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size = 0.3, random_state = 42)
```

④

```
dt_classifier = DecisionTreeClassifier()
```

⑤

```
dt_classifier.fit(X_train, y_train)
```

⑥

```
y_pred = dt_classifier.predict(X_test)
```

⑦

```
accuracy = accuracy_score(y_test, y_pred)
```

⑧

```
conf_matrix = confusion_matrix(y_test, y_pred)
```

⑨

```
accuracy
```

⑩

```
conf_matrix
```

## EXP-8

Aim:- To understand Naive Bayes classification algorithm and its implementation in python programming language.

### Program

1:- From sklearn.datasets import load\_iris

iris = load\_iris()

X = iris.data

y = iris.target

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.4, random\_state=1)

from sklearn.naive\_bayes import GaussianNB

gnb = GaussianNB()

gnb.fit(X\_train, y\_train)

y\_pred = gnb.predict(X\_test)

## EXP-9

### EXP-8

Program :- ① from sklearn import datasets

from sklearn.model\_selection import train\_test\_split

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import accuracy\_score

② iris = datasets.load\_iris()

X = iris.data

y = iris.target

} load the data

③ X\_train, X\_test, y\_train, y\_test = train\_test\_split

(X, y, test\_size=0.3, Random\_state=42)

④ nb\_classifier = GaussianNB()

⑤ nb\_classifier.fit(X\_train, y\_train)