Experiment Number 5

Name ::	Rishabh Anand	UID ::	19BCS4525
Branch ::	CSE - IoT	Sec/Grp ::	1/A
Semester ::	6th	Date ::	28th Mar, 2022
Subject ::	ML Lab	CODE ::	CSD-386

1. Aim :

To implement Naive Bayes Technique using Python programming Language.

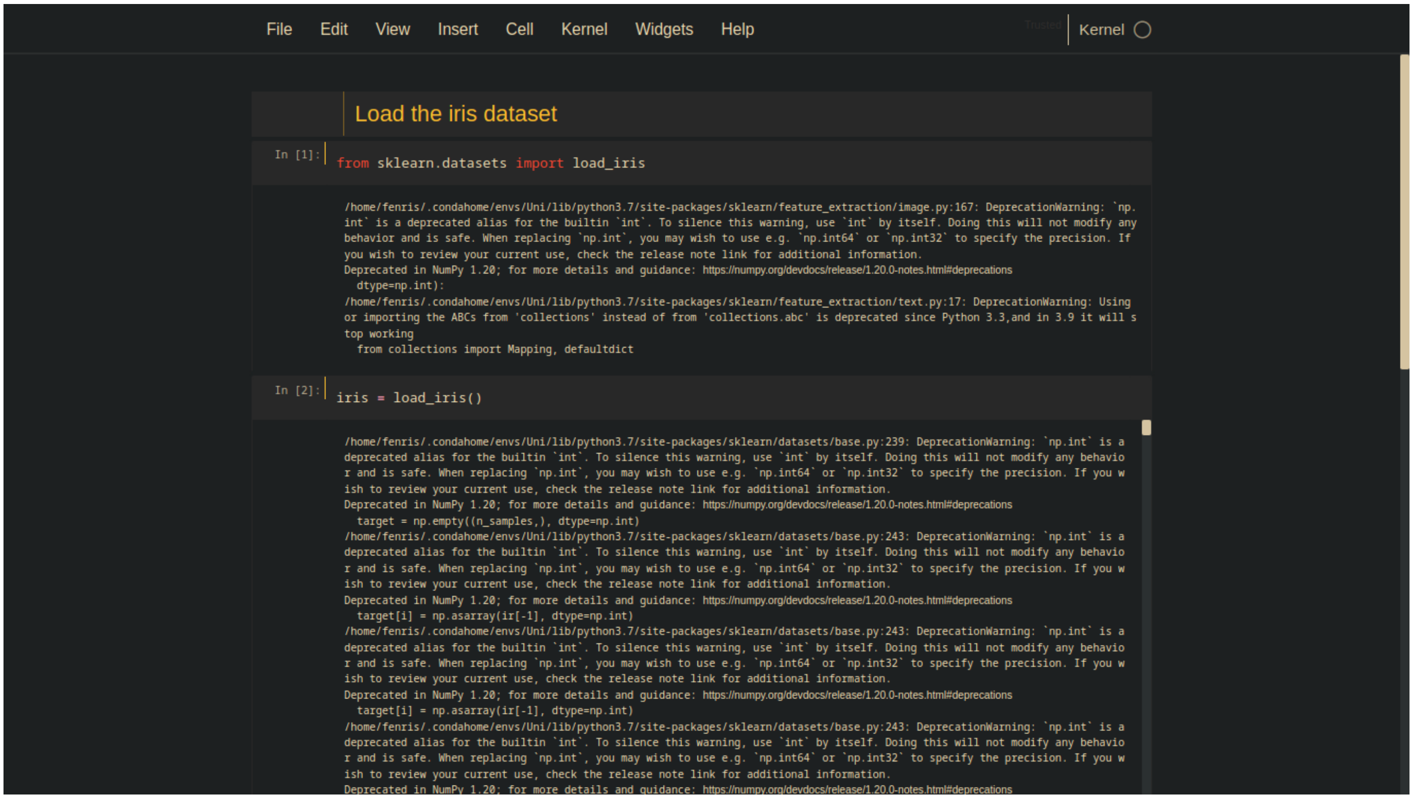
2. Task :

1. Naive Bayes on Iris Dataset.

3. Source Code :

```
# load the iris dataset
from sklearn.datasets import load_iris

iris = load_iris()

# store the feature matrix (X) and response vector (y)
X = iris.data
y = iris.target

# splitting 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=0.4,
   random_state=1)

# training the model on training set
from sklearn.naive_bayes import GaussianNB

gnb = GaussianNB()
gnb.fit(X_train, y_train)

# making predictions on the testing set
y_pred = gnb.predict(X_test)

# comparing actual response values (y_test)
# with predicted response values (y_pred)
from sklearn import metrics

print(
    ”Gaussian Naive Bayes model accuracy(in %):”,
    metrics.accuracy_score(y_test, y_pred) * 100,
)
```

4. Observations :



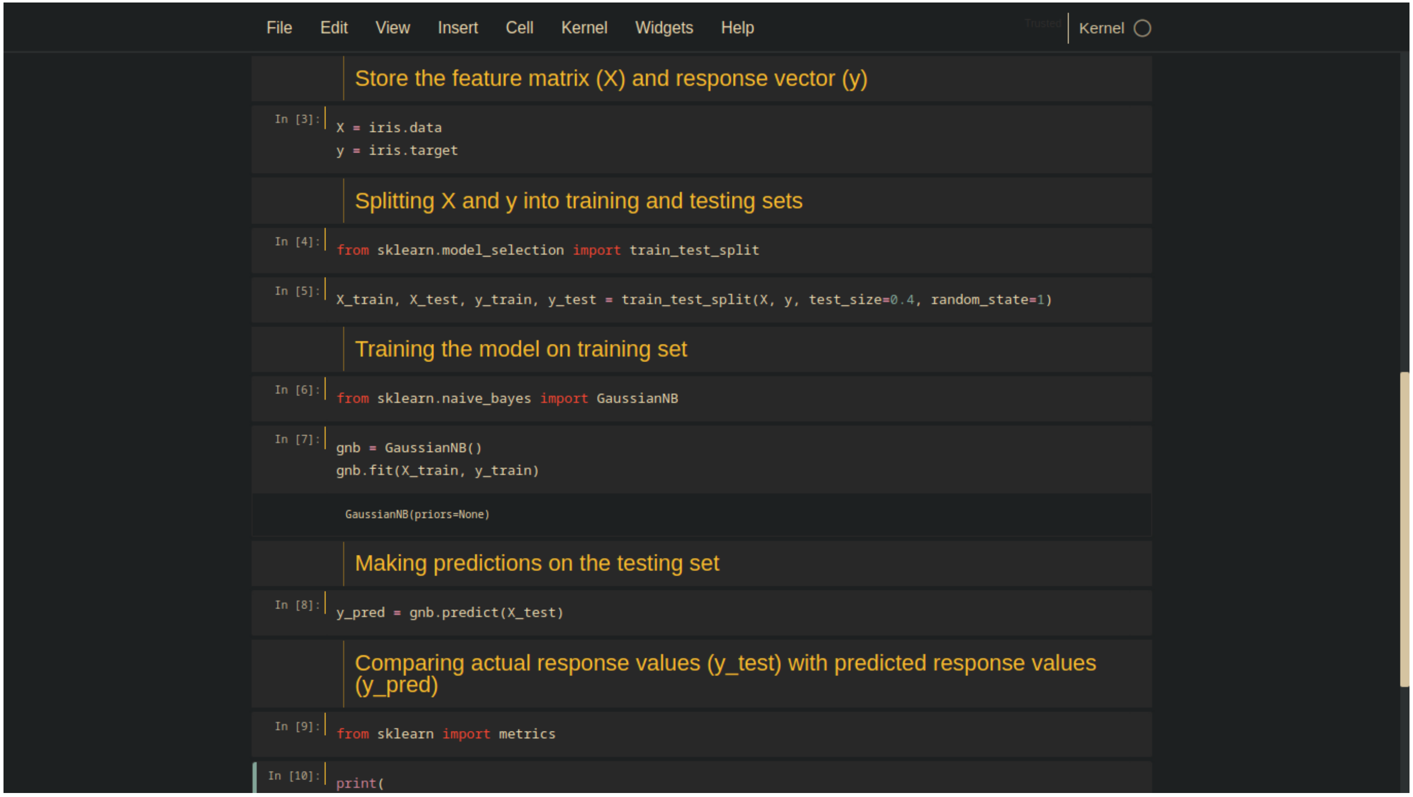
Load the iris dataset

```
In [1]: from sklearn.datasets import load_iris
```

```
/home/fenris/.condahome/envs/Uni/lib/python3.7/site-packages/sklearn/feature_extraction/image.py:167: DeprecationWarning: `np.
int` is a deprecated alias for the builtin `int`. To silence this warning, use `int` by itself. Doing this will not modify any
behavior and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` or `np.int32` to specify the precision. If
you wish to review your current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int):
/home/fenris/.condahome/envs/Uni/lib/python3.7/site-packages/sklearn/feature_extraction/text.py:17: DeprecationWarning: Using
or importing the ABCs from 'collections' instead of from 'collections.abc' is deprecated since Python 3.3,and in 3.9 it will s
top working
  from collections import Mapping, defaultdict
```

```
In [2]: iris = load_iris()
```

```
/home/fenris/.condahome/envs/Uni/lib/python3.7/site-packages/sklearn/datasets/base.py:239: DeprecationWarning: `np.int` is a
deprecated alias for the builtin `int`. To silence this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` or `np.int32` to specify the precision. If you w
ish to review your current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
  target = np.empty((n_samples,), dtype=np.int)
/home/fenris/.condahome/envs/Uni/lib/python3.7/site-packages/sklearn/datasets/base.py:243: DeprecationWarning: `np.int` is a
deprecated alias for the builtin `int`. To silence this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` or `np.int32` to specify the precision. If you w
ish to review your current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
  target[i] = np.asarray(ir[-1], dtype=np.int)
/home/fenris/.condahome/envs/Uni/lib/python3.7/site-packages/sklearn/datasets/base.py:243: DeprecationWarning: `np.int` is a
deprecated alias for the builtin `int`. To silence this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` or `np.int32` to specify the precision. If you w
ish to review your current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
  target[i] = np.asarray(ir[-1], dtype=np.int)
/home/fenris/.condahome/envs/Uni/lib/python3.7/site-packages/sklearn/datasets/base.py:243: DeprecationWarning: `np.int` is a
deprecated alias for the builtin `int`. To silence this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` or `np.int32` to specify the precision. If you w
ish to review your current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
```



Store the feature matrix (X) and response vector (y)

```
In [3]: X = iris.data
        y = iris.target
```

Splitting X and y into training and testing sets

```
In [4]: from sklearn.model_selection import train_test_split
```

```
In [5]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=1)
```

Training the model on training set

```
In [6]: from sklearn.naive_bayes import GaussianNB
```

```
In [7]: gnb = GaussianNB()
        gnb.fit(X_train, y_train)
```

```
GaussianNB(priors=None)
```

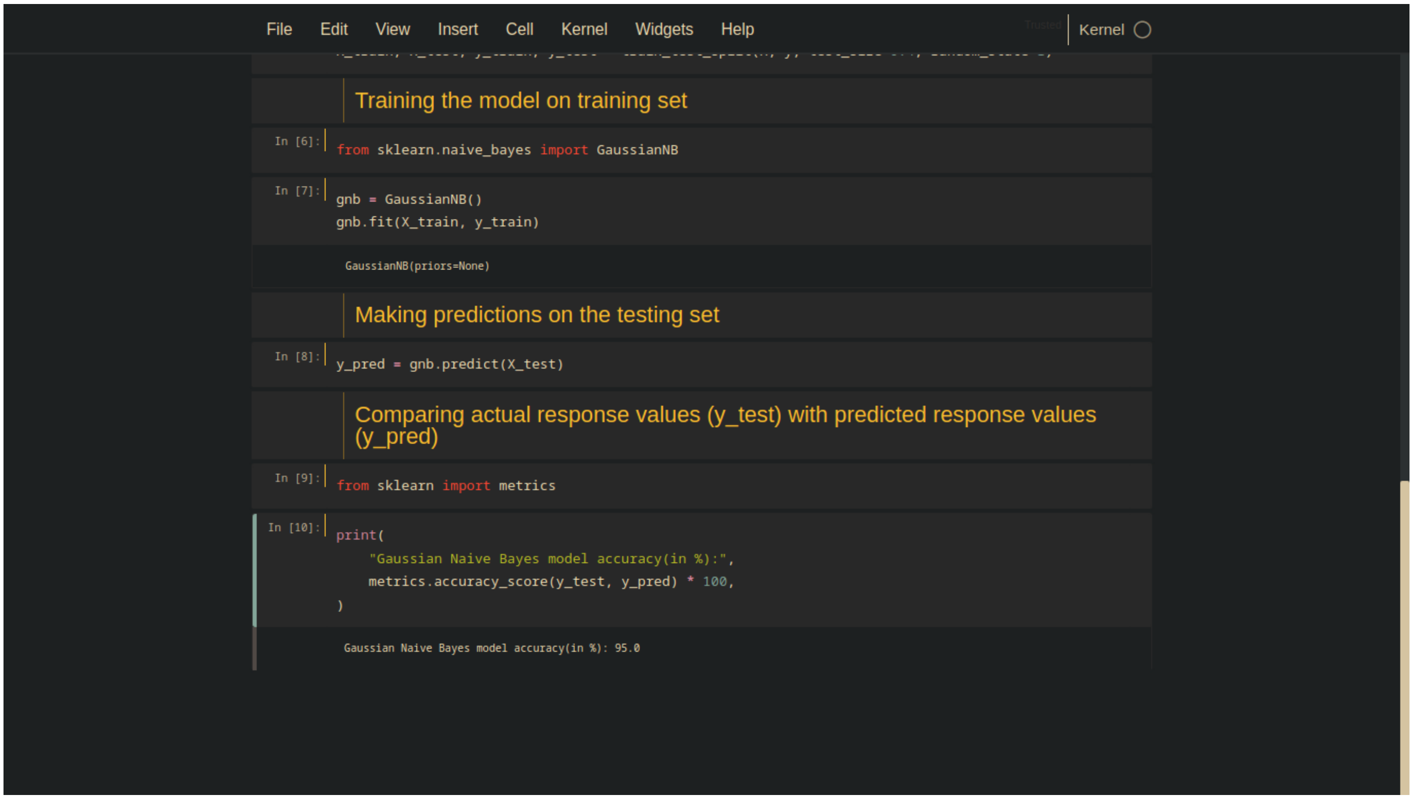
Making predictions on the testing set

```
In [8]: y_pred = gnb.predict(X_test)
```

Comparing actual response values (y_test) with predicted response values (y_pred)

```
In [9]: from sklearn import metrics
```

```
In [10]: print(
```



```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=1)
```

Training the model on training set

```
In [6]: from sklearn.naive_bayes import GaussianNB
```

```
In [7]: gnb = GaussianNB()
        gnb.fit(X_train, y_train)
```

```
GaussianNB(priors=None)
```

Making predictions on the testing set

```
In [8]: y_pred = gnb.predict(X_test)
```

Comparing actual response values (y_test) with predicted response values (y_pred)

```
In [9]: from sklearn import metrics
```

```
In [10]: print(
             "Gaussian Naive Bayes model accuracy(in %):",
             metrics.accuracy_score(y_test, y_pred) * 100,
         )
```

```
Gaussian Naive Bayes model accuracy(in %): 95.0
```

Learning Outcomes :

- Learnt to do Naïve Bayes implementation on dataset.
- Learnt to determine the likelihood of an event A happening given B happens
- Learnt to perform multi class prediction
- Learnt to predict the probability of different class based on various attributes.

S. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			