21BCE7371
RADHA KRISHNA GARG

MACHINE LEARNING LAB - 5
Logistic Regression for multiclass classification

Exercise

Use sklearn.datasets iris flower dataset to train your model using logistic regression. You need to figure out the accuracy of your model and use that to predict different samples in your test dataset. In the iris dataset, there are 150 samples containing the following features,

1. Sepal Length
2. Sepal Width
3. Petal Length
4. Petal Width

Using the above 4 features you will classify a flower into one of the three categories,

1. Setosa
2. Versicolour
3. Virginica

CODE:

```
import pandas as pd
import seaborn as sn
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.datasets import load_iris
%matplotlib inline
```

```
iris = load_iris()
dir(iris)
```

```
['DESCR',
 'data',
 'data_module',
 'feature_names',
 'filename',
 'frame',
 'target',
 'target_names']
```

```
iris.data[0]
```

```
array([5.1, 3.5, 1.4, 0.2])
```

```
iris.feature_names
```

```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

```
iris.target[0]
```

```
iris.target[0]
```

```
0
```

```
iris.target_names
```

```
array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

```
model = LogisticRegression()
X_train, X_test, y_train, y_test =
train_test_split(iris.data,iris.target,test_size=0.2)
```

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

```
▾ LogisticRegression
LogisticRegression()
```

```
model.score(X_test,y_test)
```

```
model.score(X_test,y_test)
```

```
0.9333333333333333
```

```
target_index = 98
target_index_predicted = model.predict([iris.data[target_index]])
iris.target_names[target_index_predicted]
```

```
array(['versicolor'], dtype='<U10')
```

```
iris.target_names[iris.target[target_index]]
```

```
'versicolor'
```

```
y_predicted = model.predict(X_test)
cm = confusion_matrix(y_test, y_predicted)
cm
```

```
array([[ 7,  0,  0],
       [ 0, 11,  0],
       [ 0,  2, 10]], dtype=int64)
```

```
plt.figure(figsize=(5, 3))
sn.heatmap(cm, annot=True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

```
Text(33.22222222222222, 0.5, 'Truth')
```