# Internship Weekly Report – Week 5

---

## ❖ Title Page

**Name:** Sandeep Ravaji Patel
**Domain:** Data Science
**Week Number:** Week 5

---

## ❖ Task Description

**Objective:**
To develop skills in building and evaluating supervised machine learning models, specifically focusing on classification and regression using Decision Tree and Logistic Regression algorithms with Scikit-Learn.

**Tasks Completed:**

**Supervised Learning Models:**

- Learned the key differences between regression and classification problems.
- Understood how Logistic Regression is used for binary classification.
- Implemented Decision Tree Classifier for rule-based decision making.

**Model Building:**

- Used datasets such as tennis.csv to train classification models.
- Applied **Logistic Regression** and **Decision Tree Classifier** using Scikit-Learn.
- Split data into training and test sets using train_test_split.
- Trained models and made predictions on unseen data.

**Evaluation Metrics:**

- Measured model performance using:
  - **Accuracy Score**
  - **Confusion Matrix**
  - **Classification Report**
- Interpreted confusion matrices and accuracy scores to assess model quality.

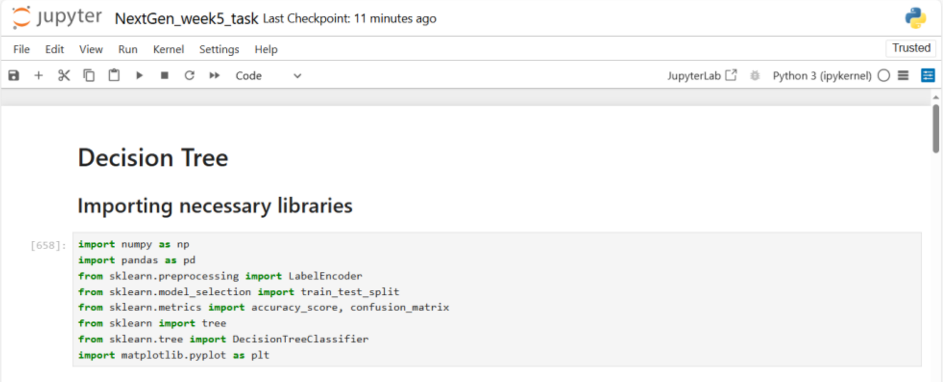
## Tools Used:

- Scikit-Learn
- Pandas
- NumPy
- Jupyter Notebook

---

## 🔷 Code Snippets / Design Screenshots

### Example 1: Importing all necessary libraries

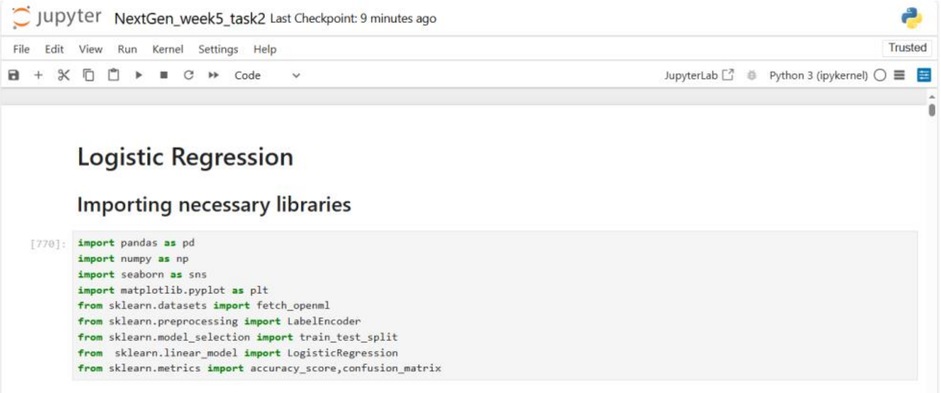


jupyter NextGen_week5_task Last Checkpoint: 11 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)

Decision Tree

Importing necessary libraries

```
[658]: import numpy as np
       import pandas as pd
       from sklearn.preprocessing import LabelEncoder
       from sklearn.model_selection import train_test_split
       from sklearn.metrics import accuracy_score, confusion_matrix
       from sklearn import tree
       from sklearn.tree import DecisionTreeClassifier
       import matplotlib.pyplot as plt
```



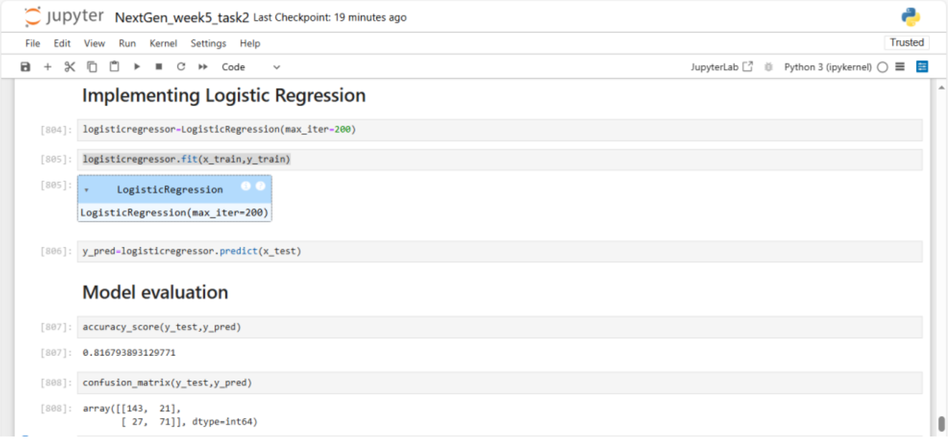
jupyter NextGen_week5_task2 Last Checkpoint: 9 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)

Logistic Regression

Importing necessary libraries

```
[770]: import pandas as pd
       import numpy as np
       import seaborn as sns
       import matplotlib.pyplot as plt
       from sklearn.datasets import fetch_openml
       from sklearn.preprocessing import LabelEncoder
       from sklearn.model_selection import train_test_split
       from  sklearn.linear_model import LogisticRegression
       from sklearn.metrics import accuracy_score,confusion_matrix
```

### Example 2: Logistic Regression with Accuracy Score and confusion matrix



jupyter NextGen_week5_task2 Last Checkpoint: 19 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)

Implementing Logistic Regression

```
[804]: logisticregressor=LogisticRegression(max_iter=200)

[805]: logisticregressor.fit(x_train,y_train)

[805]: ▾ LogisticRegression
       LogisticRegression(max_iter=200)

[806]: y_pred=logisticregressor.predict(x_test)
```

Model evaluation

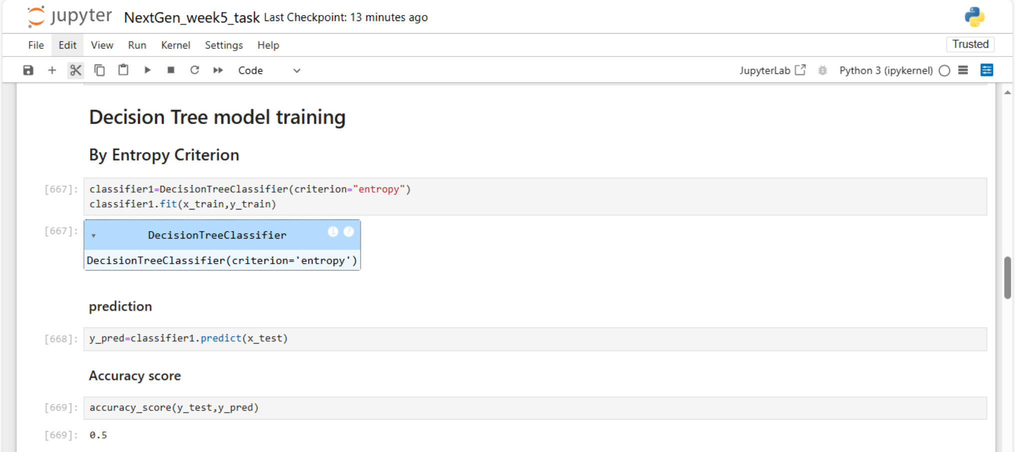
```
[807]: accuracy_score(y_test,y_pred)

[807]: 0.816793893129771

[808]: confusion_matrix(y_test,y_pred)

[808]: array([[143,  21],
              [ 27,  71]], dtype=int64)
```

## Example 3: Decision Tree Classifier with Accuracy Score and Confusion Matrix


jupyter
NextGen_week5_task Last Checkpoint: 13 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)

Decision Tree model training

By Entropy Criterion

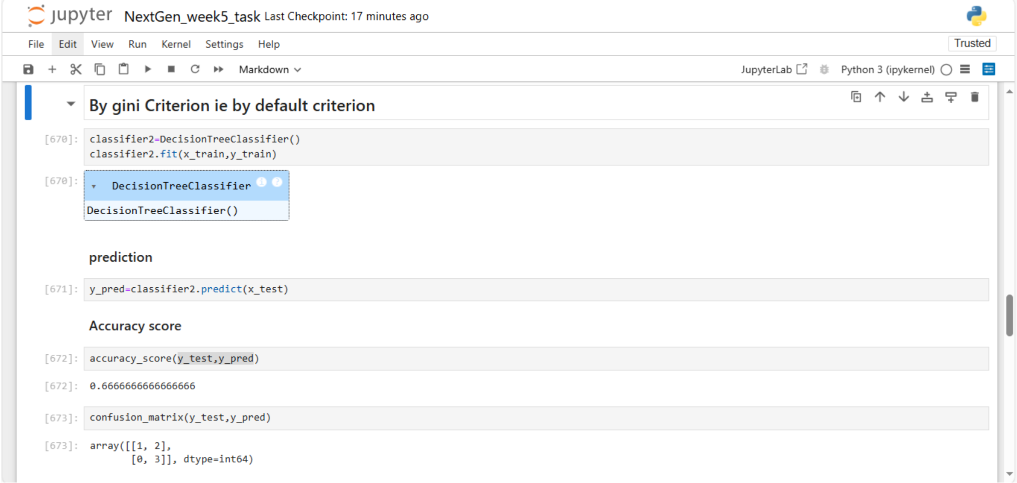
```
[667]: classifier1=DecisionTreeClassifier(criterion="entropy")
       classifier1.fit(x_train,y_train)
[667]: DecisionTreeClassifier
       DecisionTreeClassifier(criterion='entropy')
```

prediction

```
[668]: y_pred=classifier1.predict(x_test)
```

Accuracy score

```
[669]: accuracy_score(y_test,y_pred)
[669]: 0.5
```


jupyter
NextGen_week5_task Last Checkpoint: 17 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Markdown
JupyterLab
Python 3 (ipykernel)

By gini Criterion ie by default criterion

```
[670]: classifier2=DecisionTreeClassifier()
       classifier2.fit(x_train,y_train)
[670]: DecisionTreeClassifier
       DecisionTreeClassifier()
```

prediction

```
[671]: y_pred=classifier2.predict(x_test)
```

Accuracy score

```
[672]: accuracy_score(y_test,y_pred)
[672]: 0.6666666666666666
[673]: confusion_matrix(y_test,y_pred)
[673]: array([[1, 2],
              [0, 3]], dtype=int64)
```

---

## ❖ Challenges Faced

### 1. Dataset Structure Understanding:

- Took time to understand the structure and meaning of features in tennis.csv.

### 2. Classification Evaluation:

- Initially unclear on how to interpret confusion matrix outputs.
- *Resolution:* Reviewed metric documentation and practical examples for better clarity.

---

## ❖ Learning Outcome

- Developed a solid understanding of classification algorithms and their real-world application.
- Built and evaluated Logistic Regression and Decision Tree models using Scikit-Learn.
- Gained experience with performance metrics like accuracy and confusion matrix.

- Improved ability to analyze results and interpret machine learning outputs.

---

### ❖ Next Steps

For **Week 6**, the focus will be on:

- Learning **Unsupervised Learning** techniques.
- Implementing **K-Means Clustering** and performing **PCA** for dimensionality reduction.

---

### ❖ Resources

- **Regression Guide**
- **Classification Guide**
- Scikit-Learn Documentation
- **Dataset Used:** tennis.csv