**iPhone Purchase Status Prediction**

**1. Introduction**

This document outlines the end-to-end process for predicting iPhone purchase status using a Decision Tree Classifier. The project involves data loading, exploration, visualization, model training, and evaluation.

**2. Solution Architecture**

**2.1 Data Collection**

* **Data Source**: iphone\_purchase\_records.csv
* **Description**: Contains records of customer demographics and their iPhone purchase status.

**2.2 Data Preparation**

* **Libraries Used**:
  + pandas: For data manipulation
  + numpy: For numerical operations
  + seaborn, matplotlib: For data visualization
  + sklearn: For machine learning and model evaluation

**2.3 Data Processing**

* **Cleaning**:
  + Removal of duplicate rows
  + Handling missing values (if any)
* **Feature Engineering**:
  + Encoding categorical variables
* **Splitting Data**:
  + Training set and testing set

**2.4 Model Building**

* **Model Used**: Decision Tree Classifier
  + **Parameters**: max\_depth=3, criterion='gini'

**2.5 Model Evaluation**

* **Metrics**:
  + Accuracy
  + Classification Report
  + Confusion Matrix

**3. Methodology**

**3.1 Importing Libraries**

python

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

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

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import confusion\_matrix, accuracy\_score, classification\_report

import time

**3.2 Loading the Dataset**

python

data = pd.read\_csv("iphone\_purchase\_records.csv")

print(data.head())

print(data.tail())

**3.3 Preliminary Data Exploration**

python

print(data.shape)

print(data.info())

print(data.isnull().sum())

print(data.duplicated().sum())

data = data.drop\_duplicates()

print(data.duplicated().sum())

**3.4 Outlier Detection and Visualization**

python

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)

sns.boxplot(data['Age'])

plt.title("Distribution of Age")

plt.ylabel('Age')

plt.subplot(1, 2, 2)

sns.boxplot(data['Salary'])

plt.title("Distribution of Salary")

plt.ylabel('Salary')

plt.tight\_layout()

plt.savefig("D:/path\_to\_save/box\_plots.jpg")

plt.show()

**3.5 Exploratory Data Analysis (EDA)**

python

print(data.describe())

plt.figure(figsize=(8, 6))

sns.displot(data=data, x='Salary', hue='Gender', kind='kde', fill=True)

plt.title("Distribution of Salary across Gender")

plt.xlabel('Salary')

plt.savefig("D:/path\_to\_save/kde\_salary.png")

plt.show()

plt.figure(figsize=(8, 6))

sns.displot(data=data, x='Age', hue='Gender', kind='kde', fill=True)

plt.title("Distribution of Age across Gender")

plt.xlabel('Age')

plt.savefig("D:/path\_to\_save/kde\_age.png")

plt.show()

plt.figure(figsize=(8, 6))

sns.barplot(data=data, x='Gender', y='Purchase Iphone', errwidth=0)

plt.title("iPhone Purchasing Across Gender")

plt.xlabel('Gender')

plt.ylabel('Purchasing Frequency')

plt.savefig("D:/path\_to\_save/bar\_plot.png")

plt.show()

**3.6 Preparing Data for Modeling**

python

data['Gender'] = data['Gender'].map({'Female': 1, 'Male': 0})

X = data.drop(columns=['Purchase Iphone'])

y = data['Purchase Iphone']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.20, random\_state=1234)

**3.7 Model Training**

python

dt = DecisionTreeClassifier(max\_depth=3, criterion='gini')

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

**3.8 Model Evaluation**

python

dt\_pred = dt.predict(X\_test)

accuracy = accuracy\_score(y\_test, dt\_pred) \* 100

print(f'Accuracy of the model: {accuracy:.2f}%')

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*Classification Report\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print(classification\_report(y\_test, dt\_pred))

conf\_mat = confusion\_matrix(y\_test, dt\_pred)

plt.figure(figsize=(8, 6))

sns.heatmap(conf\_mat, annot=True, fmt='d', cmap='viridis', cbar=False)

plt.title("Confusion Matrix")

plt.xlabel("Predicted Label")

plt.ylabel("Actual Label")

plt.savefig("D:/path\_to\_save/confusion\_matrix.png")

plt.show()

**5. Conclusion**

The Decision Tree Classifier model has been successfully trained and evaluated for predicting iPhone purchase status. The model achieved an accuracy of 88.16%, and the results were further evaluated using a classification report and confusion matrix.