Predicting Customer Purchases Using Random Forest Machine Learning

APPROACH:

- In today's digital age, understanding customer behavior is key to enhancing business strategies.
- This project aims to predict whether a customer will make a purchase based on two simple factors: age and estimated salary.
- By using a Random Forest Classifier, a powerful machine learning algorithm, we can accurately predict buying patterns and help businesses make smarter decisions.
- The project starts with a small dataset containing information about customers' ages, salaries, and whether they made a purchase or not.
- The data is divided into two sets—one for training the model and another for testing its predictions.
- The model learns from the training data and then makes predictions on the test data, allowing us to assess how well it performs.

CODE:

```
# Import necessary libraries
Import numpy as np
Import pandas as pd
From sklearn.model_selection import train_test_split
From sklearn.ensemble import RandomForestClassifier
From sklearn.metrics import accuracy_score
# Example dataset (Age, Estimated Salary, Purchased)
Data = {
   ‘Age’: [22, 25, 47, 52, 46, 56, 35, 23, 48, 42],
   ‘EstimatedSalary’: [15000, 18000, 55000, 57000, 45000, 60000, 35000, 12000, 52000, 49000],
```

```
    'Purchased': [0, 0, 1, 1, 0, 1, 0, 0, 1, 0]
}
# Convert the data into a pandas DataFrame
Df = pd.DataFrame(data)
# Features (X) and target (y)
X = df[['Age', 'EstimatedSalary']]  # Input features
Y = df['Purchased']  # Target variable
# Split the data into training and testing sets (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Initialize the Random Forest Classifier
Classifier = RandomForestClassifier(n_estimators=10, random_state=42)
# Train the classifier
Classifier.fit(X_train, y_train)
```

```
# Make predictions on the test set
Y_pred = classifier.predict(X_test)
# Calculate accuracy
Accuracy = accuracy_score(y_test, y_pred)
Print(f”Accuracy: {accuracy * 100:.2f}%”)
```

EXECUTION:

Kiren V nair 220901042.ipynl

Code | Notebook | Python (Pyodide)

[2]:
```python
# Import necessary libraries
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# Example dataset (Age, Estimated Salary, Purchased)
data = {
    'Age': [22, 25, 47, 52, 46, 56, 35, 23, 48, 42],
    'EstimatedSalary': [15000, 18000, 55000, 57000, 45000, 60000, 350
    'Purchased': [0, 0, 1, 1, 0, 1, 0, 0, 1, 0]
}

# Convert the data into a pandas DataFrame
df = pd.DataFrame(data)

# Features (X) and target (y)
X = df[['Age', 'EstimatedSalary']]  # Input features
y = df['Purchased']  # Target variable

# Split the data into training and testing sets (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0

# Initialize the Random Forest Classifier
classifier = RandomForestClassifier(n_estimators=10, random_state=42)

# Train the classifier
classifier.fit(X_train, y_train)

# Make predictions on the test set
y_pred = classifier.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy * 100:.2f}%")
```

```
Accuracy: 100.00%
```

[ ]:

(Pyodide) | Idle Mode: Edit Ln 1, Col 1 Kiren V nair 220901042.ipynb 3

EXPLANATION:

- Data Creation: We start with a dataset of 10 customers that includes their age, estimated salary, and whether or not they made a purchase
- Model Training: We use 80% of the data to train the model, allowing it to learn patterns from the customers who did and didn’t make purchases.
- Prediction and Evaluation: The remaining 20% of the data is used to test the model. In this case, the model was able to predict customer behavior with 100% accuracy. This means that, based on the customer’s age and salary, the model could perfectly guess whether they would make a purchase.

CONCLUSION:

This project demonstrates how machine learning, even with basic data, can predict important customer behaviors. With just age and salary information, we can build a model that helps businesses target potential customers more effectively. The Random Forest Classifier provides a flexible, accurate solution that can easily be scaled up with more data to handle more complex business needs.