JAIN UNIVERISTY

FUNDAMENTALS OF MACHINE LEARNING

NAME – KRISH DODHIA

USN - 22BTRAD020

LAB - 5

CODE:

Import pandas pd

f1 = pd.read_csv(“User_Data.csv”)

print(f1.head())

```
import pandas as pd
f1 = pd.read_csv('User_Data.csv')
print(f1.head())
```

```
    User ID  Gender  Age  EstimatedSalary  Purchased
0  15624510    Male   19            19000          0
1  15810944    Male   35            20000          0
2  15668575  Female   26            43000          0
3  15603246  Female   27            57000          0
4  15804002    Male   19            76000          0
```

EXPLANATION:

The code snippet uses Pandas, a Python library for data manipulation, to read the contents of a CSV file named "User_Data.csv" into a DataFrame called **f1**. Subsequently, it prints the first five rows of this DataFrame, showcasing a preview of the data's structure and initial values. This action aids in understanding the dataset's columns, data types, and the nature of information contained within, assisting in further analysis and processing of the data.

CODE:

f2 = pd.get_dummies(f1,columns=[“Gender”])

print(f2.head())

```
f2 = pd.get_dummies(f1,columns=['Gender'])
print(f2.head())
```

```
   User ID  Age  EstimatedSalary  Purchased  Gender_Female  Gender_Male
0  15624510   19            19000          0              0            1
1  15810944   35            20000          0              0            1
2  15668575   26            43000          0              1            0
3  15603246   27            57000          0              1            0
4  15804002   19            76000          0              0            1
```

\+ Code + Text

EXPLAINATION:

The code utilizes Pandas' **get_dummies()** function to create binary (dummy) columns for categorical variables, specifically for the "Gender" column in the DataFrame **f1**. It transforms categorical data into numerical representations, where each unique category becomes a new binary column (0 or 1) indicating the presence or absence of that category. The resulting DataFrame **f2** contains these newly created binary columns for "Gender," allowing for easier analysis or machine learning tasks that require numerical inputs. The **print(f2.head())** statement displays the initial rows of the modified DataFrame **f2**, offering a glimpse of the transformed data structure.

```
[7]  from sklearn.linear_model import LogisticRegression
     from sklearn.model_selection import train_test_split
     from sklearn.preprocessing import StandardScaler
```

```
[8]  X_data = f2.iloc[:,[1,2,4,5]].values
     Y_data = f1.iloc[:, 3].values
```

```
[11] X_train, X_test, Y_train, Y_test = train_test_split(X_data, Y_data, test_size = 0.2)
```

```
ss = StandardScaler()
x_train_ss = ss.fit_transform(X_train)
x_test_ss = ss.fit_transform(X_test)
x_train_ss, x_test_ss, Y_train, Y_test = train_test_split(f1[["Age","EstimatedSalary"]],
f1.Purchased, test_size = 0.2)
```

```
[18] model = LogisticRegression()
     #model.fit(X_train, Y_train)
     model.fit(x_train_ss, Y_train)
```

This code snippet prepares data for a logistic regression model by selecting specific columns from the DataFrame **f2** to form the feature set (**X_data**) and extracting the target variable (**Y_data**) from the original DataFrame **f1**. It then splits the data into training and testing sets using **train_test_split()**. Next, it scales the numerical features using **StandardScaler** separately for the training and testing sets. Finally, it initializes a Logistic Regression model, fits it using the scaled training data (**x_train_ss**), and the corresponding target values (**Y_train**). The commented-out section (**#model.fit(X_train, Y_train)**) suggests an alternative approach using different columns for training.

CODE:

print(model.predict(x_test_ss))

```
print(model.predict(x_test_ss))
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0]
```

EXPLANATION:

The code **model.predict(x_test_ss)** uses the trained Logistic Regression model (**model**) to make predictions on the scaled test data (**x_test_ss**). It returns the predicted binary labels based on the logistic regression algorithm applied to the test set's features.

CODE:

print(model.predict_probab(x_test_ss))

```
probabilities = model.predict_proba(x_test_ss)
print(probabilities)
```

```
[0.60946134 0.39053866]
[0.57476884 0.42523116]
[0.51737774 0.48262226]
[0.58155267 0.41844733]
[0.5496718  0.4503282 ]
[0.58942817 0.41057183]
[0.57023042 0.42976958]
[0.65154772 0.34845228]
[0.51853521 0.48146479]
[0.57929468 0.42070532]
[0.55883422 0.44116578]
[0.64201626 0.35798374]
[0.53471353 0.46528647]
[0.66613425 0.33386575]
[0.55769097 0.44230903]
[0.63237121 0.36762879]
[0.55425765 0.44574235]
[0.65049449 0.34950551]
[0.59614245 0.40385755]
[0.58042407 0.41957593]
[0.5916701  0.4083299 ]
[0.61825213 0.38174787]
[0.60282095 0.39717905]
[0.63237119 0.36762881]
[0.53586672 0.46413328]
[0.53240599 0.46759401]
[0.62152913 0.37847087]
[0.66095945 0.33904055]
[0.5939082  0.4060918 ]
```

EXPLANATION:

The code **model.predict_proba(x_test_ss)** employs a trained Logistic Regression model (**model**) to calculate the predicted probabilities for each class label pertaining to the test data (**x_test_ss**). The resulting **probabilities** variable holds an array where each row represents a sample from the test set, and each column indicates the probability of that sample belonging to a specific class

CODE:

print(model.score(x_test_ss,Y_test))

```
print(model.score(x_test_ss,Y_test))
```

```
0.55
```

EXPLANATION:

The **model.score(x_test_ss, Y_test)** computes the accuracy of the Logistic Regression model (**model**) on the provided test data (**x_test_ss**) by comparing the predicted labels against the true labels (**Y_test**). It returns a value representing the accuracy score, indicating the proportion of correctly predicted labels in the test set.

LINK OF CODE:

https://colab.research.google.com/drive/1LY7KOfyH-gUp3jgBrVAD38rgyv3ZdsBZ#scrollTo=UXqKPKaLoJBP