hajira-imran-44594-lab-11

April 22, 2025

[24]:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, accuracy_score
from sklearn.preprocessing import LabelEncoder
```

[2]:

```
# Load the dataset
df = pd.read_csv("/content/Public Livelihood Data.csv")
```

[4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 8 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Designation     32561 non-null  object
 1   Education       32561 non-null  object
 2   Marital Status  32561 non-null  object
 3   Field           32561 non-null  object
 4   Race            32561 non-null  object
 5   Gender          32561 non-null  object
 6   Country         32561 non-null  object
 7   Salary          32561 non-null  object
dtypes: object(8)
memory usage: 2.0+ MB
```

[23]:

```
#drop missing values
df.dropna(inplace=True)
```

[30]:

```
# Create a LabelEncoder object
le = LabelEncoder()

# Apply LabelEncoder to only categorical columns
categorical_columns = df.select_dtypes(include=['object']).columns
df[categorical_columns] = df[categorical_columns].apply(le.fit_transform)
```

[31]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 8 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Designation     32561 non-null  int64
 1   Education       32561 non-null  int64
 2   Marital Status  32561 non-null  int64
 3   Field           32561 non-null  int64
 4   Race            32561 non-null  int64
 5   Gender          32561 non-null  int64
 6   Country         32561 non-null  int64
 7   Salary          32561 non-null  int64
dtypes: int64(8)
memory usage: 2.0 MB
```

[32]:

```python
# Separate features and target
X = df.drop('Salary', axis=1)
y = df['Salary']
```

[33]:

```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

[34]:

```python
#Train the Naive Bayes Classifier
model = GaussianNB()
model.fit(X_train, y_train)
```

[34]:

```
GaussianNB()
```

[35]:

```python
#Make Predictions
y_pred = model.predict(X_test)
```

[36]:

```python
#Evaluate the Classifier
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.760325502840473
```

[37]:

```python
#Show Predicted vs Actual Labels
print("\nPredicted labels:", y_pred)
print("Actual labels:    ", y_test.values)
```

```

Predicted labels: [0 0 0 … 1 0 0]
Actual labels:    [0 0 1 … 1 0 0]
```