alishba-waqar-46997-lab-11

April 22, 2025

[23]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, classification_report
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
```

[24]:
```
df = pd.read_csv('/content/Public Livelihood Data.csv')
df.dropna(inplace=True)
```

[25]:
```
df.columns
```

[25]:
```
Index(['Designation', 'Education', 'Marital Status', 'Field', 'Race', 'Gender',
       'Country', 'Salary'],
      dtype='object')
```

[34]:
```
# 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)
```

[35]:
```
X = df.drop('Salary', axis=1)
y = df['Salary']
```

[36]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)
```

[37]:
```
# Step 6: Train the Naive Bayes Classifier
model = GaussianNB()
model.fit(X_train, y_train)
```

[37]:
```
GaussianNB()
```

[38]:
```
y_pred = model.predict(X_test)
```

[39]:
```
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
```

[40]:

```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.760325502840473
```

[41]:

```
print("\nPredicted labels:", y_pred)
print("Actual labels:    ", y_test.values)
print("\nClassification Report:\n", classification_report(y_test,y_pred))
```

```

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

Classification Report:
               precision    recall  f1-score   support

           0       0.87      0.80      0.84      4942
           1       0.50      0.63      0.56      1571

    accuracy                           0.76      6513
   macro avg       0.69      0.72      0.70      6513
weighted avg       0.78      0.76      0.77      6513
```