Name: Annie Mahajan

Registration No.: 23BRS1121

Lab – 03

Example 1

```
import pandas as pd
from sklearn.naive_bayes import CategoricalNB
X = pd.DataFrame({
   'Outlook': ['Sunny','Sunny','Overcast','Rainy','Rainy','Sunny'],
   'Humidity': ['High','High','High','Normal','Normal','Normal'],
   'Wind': ['Weak','Strong','Weak','Weak','Strong','Weak']
})
y = ['No','No','Yes','Yes','No','Yes']
X_encoded = X.apply(lambda col: col.astype('category').cat.codes)
model = CategoricalNB()
model.fit(X_encoded, y)
test = pd.DataFrame([['Sunny','High','Weak']], columns=X.columns)
test_encoded = test.apply(lambda col: col.astype('category').cat.codes)
print("Prediction:", model.predict(test_encoded))
```

```
PS C:\Users\scope1\Downloads\ML> python Example1.py
Prediction: ['No']
```

Example 2

```
import pandas as pd
from sklearn.preprocessing import OrdinalEncoder
from sklearn.naive_bayes import CategoricalNB
X = pd.DataFrame({
   'Outlook': ['Sunny','Sunny','Overcast','Rainy','Rainy','Sunny'],
   'Humidity': ['High','High','High','Normal','Normal','Normal'],
   'Wind': ['Weak','Strong','Weak','Weak','Strong','Weak']
})
y = ['No','No','Yes','Yes','No','Yes']
encoder = OrdinalEncoder()
```

```
X_encoded = encoder.fit_transform(X)
model = CategoricalNB()
model.fit(X_encoded, y)
test = pd.DataFrame([['Sunny','High','Weak']], columns=X.columns)
test_encoded = encoder.transform(test)
print("Prediction:", model.predict(test_encoded))
```

```
PS C:\Users\scope1\Downloads\ML> python Example2.py
Prediction: ['No']
```

Example 3

```
import pandas as pd
from sklearn.preprocessing import OrdinalEncoder
X = pd.DataFrame({
   'Color': ['Red','Blue','Green','Blue','Red'],
   'Size': ['Small','Medium','Large','Small','Large']
})
y = ['Yes','No','Yes','No','Yes']
encoder = OrdinalEncoder()
X_encoded = encoder.fit_transform(X)
print("Encoded Training Data:\n", X_encoded)
X_test = pd.DataFrame({
   'Color': ['Green','Red'],
   'Size': ['Medium','Small']
})
X_test_encoded = encoder.transform(X_test)
print("Encoded Test Data:\n", X_test_encoded)
```

```
Encoded Training Data:
 [[2. 2.]
 [0. 1.]
 [1. 0.]
 [0. 2.]
 [2. 0.]]
Encoded Test Data:
 [[1. 1.]
 [2. 2.]]
```

Example 4

```
import pandas as pd
from sklearn.preprocessing import OrdinalEncoder
X = pd.DataFrame({
   'Color': ['Red','Blue','Green','Blue','Red'],
   'Size': ['Small','Medium','Large','Small','Large']
})
encoder = OrdinalEncoder(handle_unknown='use_encoded_value', unknown_value=-1)
encoder.fit(X)
X_test_new = pd.DataFrame({
   'Color': ['Yellow'],
   'Size': ['Small']
})
X_test_new_encoded = encoder.transform(X_test_new)
print("Encoded Unseen Data:\n", X_test_new_encoded)
```

```
PS C:\Users\scope1\Downloads\ML> & 'C:/Users/scope1/Downloads/ML/venv/Scripts/python.exe' -c "import pandas as pd;
from sklearn.preprocessing import OrdinalEncoder; X = pd.DataFrame({'Color': ['Red','Blue','Green','Blue','Red'],'S
ize': ['Small','Medium','Large','Small','Large']}); encoder = OrdinalEncoder(handle_unknown='use_encoded_value', un
known_value=-1); encoder.fit(X); X_test_new = pd.DataFrame({'Color': ['Yellow'], 'Size': ['Small']}); X_test_new_en
coded = encoder.transform(X_test_new); print('Encoded Unseen Data:\n', X_test_new_encoded)"; echo Exit:$LASTEXITCOD
E
Encoded Unseen Data:
 [[-1.  2.]]
Exit:0
```

Example 5

```
import numpy as np
from sklearn.naive_bayes import GaussianNB
X = np.array([[5.1],[4.9],[5.0],[6.7],[6.5],[6.8]])
y = np.array(['A','A','A','B','B','B'])
model = GaussianNB()
model.fit(X, y)
test = np.array([[6.6]])
print("Prediction:", model.predict(test))
```

```
PS C:\Users\scope1\Downloads\ML> & 'C:/Users/scope1/Downloads/ML/venv/Scripts/python.exe' -c "import numpy as np; f
rom sklearn.naive_bayes import GaussianNB; X = np.array([[5.1],[4.9],[5.0],[6.7],[6.5],[6.8]]); y = np.array(['A','
A','A','B','B','B']); model = GaussianNB(); model.fit(X, y); test = np.array([[6.6]]); print('Prediction:', model.p
redict(test))"; echo Exit:$LASTEXITCODE
Prediction: ['B']
Exit:0
```

Example 6

```
import pandas as pd
from sklearn.preprocessing import OrdinalEncoder
from sklearn.naive_bayes import GaussianNB
data = {
   'Age': [25,40,35,50,28],
   'Income': [50000,60000,58000,80000,52000],
   'MaritalStatus': ['Single','Married','Single','Married','Single'],
   'CreditScore': [700,650,720,600,690],
   'LoanApproved': ['Yes','No','Yes','No','Yes']
}
df = pd.DataFrame(data)
X = df.drop('LoanApproved', axis=1)
y = df['LoanApproved']
categorical_cols = ['MaritalStatus']
encoder = OrdinalEncoder()
X[categorical_cols] = encoder.fit_transform(X[categorical_cols])
model = GaussianNB()
model.fit(X, y)
test = pd.DataFrame({
   'Age':[30],
   'Income':[54000],
   'MaritalStatus':['Single'],
   'CreditScore':[710]
})
test[categorical_cols] = encoder.transform(test[categorical_cols])
pred = model.predict(test)
print("Predicted Loan Approval:", pred[0])
```

```
PS C:\Users\scope1\Downloads\ML> & 'C:/Users/scope1/Downloads/ML/venv/Scripts/python.exe' -c "import pandas as pd;
from sklearn.preprocessing import OrdinalEncoder; from sklearn.naive_bayes import GaussianNB; data = {'Age': [25,40
,35,50,28], 'Income': [50000,60000,58000,80000,52000], 'MaritalStatus': ['Single','Married','Single','Married','Sin
gle'], 'CreditScore': [700,650,720,600,690], 'LoanApproved': ['Yes','No','Yes','No','Yes']}; df = pd.DataFrame(data
); X = df.drop('LoanApproved', axis=1); y = df['LoanApproved']; categorical_cols = ['MaritalStatus']; encoder = Ord
inalEncoder(); X[categorical_cols] = encoder.fit_transform(X[categorical_cols]); model = GaussianNB(); model.fit(X,
 y); test = pd.DataFrame({'Age':[30], 'Income':[54000], 'MaritalStatus':['Single'], 'CreditScore':[710]}); test[cat
egorical_cols] = encoder.transform(test[categorical_cols]); pred = model.predict(test); print('Predicted Loan Appro
val:', pred[0])"; echo Exit:$LASTEXITCODE
Predicted Loan Approval: Yes
Exit:0
```

Exercise 1

```
import pandas as pd
from sklearn.preprocessing import OrdinalEncoder
from sklearn.naive_bayes import GaussianNB
```

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
df = pd.read_csv("adult.csv")
X = df.drop('income', axis=1)
y = df['income']
cat_cols = X.select_dtypes(include='object').columns
encoder = OrdinalEncoder()
X[cat_cols] = encoder.fit_transform(X[cat_cols])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
model = GaussianNB()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

```
PS C:\Users\scope1\Downloads\ML> python Exercise1.py
Accuracy: 0.7943494728221927
```

Exercise 2

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score
df = pd.read_csv("diabetes.csv")
X = df.drop('Outcome', axis=1)
y = df['Outcome']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
model = GaussianNB()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

```
df shape (768, 9)
columns ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction',
'Age', 'Outcome']
Accuracy 0.7987012987012987
```