

Preference Prediction

This machine learning model indent to predict product preferences specific to different countries. This model will help to analyze & review product requirement associated with counties included in the model.

The dataset has been imported from UCI Machine Learning Repository

The dataset contains of 5+Lakhs data of E-Commerce Sales

1. Prerequisites

Python installed on your machine.

Required libraries: streamlit, pandas, scikit-learn, and joblib.

A pre-trained model saved as work.joblib (created using the previous code snippet).

You can install Streamlit and other dependencies using pip:

Copypip install streamlit pandas scikit-learn joblib

Application Components

2. Imports:

streamlit for creating the web application interface.

pandas for data manipulation.

joblib for loading the pre-trained model.

LabelEncoder from sklearn for encoding input data.

3. Label Encoding:

The LabelEncoder is applied to all non-datetime columns of the DataFrame df. This process converts categorical string data into numerical data that the machine learning model can process.

4. Feature and Target Selection:

The features X are selected from ['Description', 'Quantity'] columns, and the target y is the 'Country' column.

5. Splitting Dataset:

The dataset is split into training and testing sets using an 80-20 split with a fixed random state for reproducibility.

6. TimeStamp:

The dataset has date and time the encoding as been done for that .

```
def encode(df1, col, max_val):  
    df1[col + '_sin'] = np.sin(2 * np.pi * df1[col]/max_val)  
    df1[col + '_cos'] = np.cos(2 * np.pi * df1[col]/max_val)  
    return df1
```

```
df1['month'] = df1.InvoiceDate.dt.month
```

```
df1 = encode(df1, 'month', 12)
```

```
df1['day'] = df1.InvoiceDate.dt.day
```

```
df2 = encode(df1, 'day', 31)
```

```
df3=df1.drop("InvoiceDate",axis=1)
```

7. Model Training:

A RandomForestClassifier object is initialized and trained (fit) on the training dataset.

Prediction and Evaluation:

The trained model predicts the labels of the test data.

The accuracy and detailed classification report (precision, recall, f1-score) of the model are printed.

8. Model Persistence:

The trained model is saved to a file named work.joblib using **joblib.dump()**. This allows the model to be reloaded and used for predictions without retraining.

Usage

Ensure your dataset is loaded into a pandas DataFrame df with columns ['Description', 'Quantity', 'Country'].

9. Compile & Run

Run the script in a Python environment. Using `python -m streamlib run my.py`

Review the printed accuracy and classification report output.

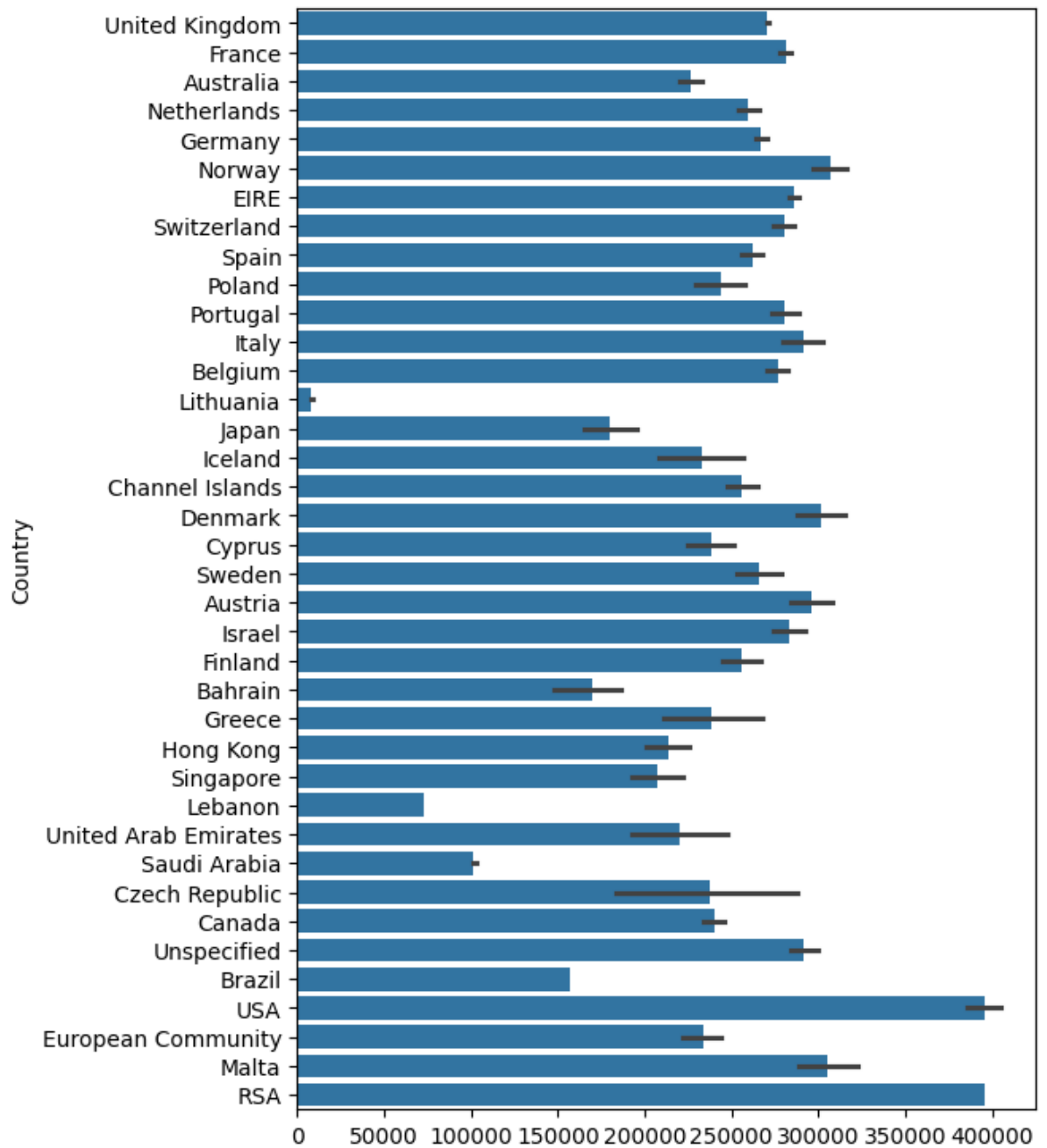
The model will be saved as **work.joblib** for future use.

10. Visuvalization

HeatMap:



Bar Chart:



Output page:

Deploy 

Preference Prediction

Enter Product :

FAMILY ALBUM WHITE PICTURE FRAME

Enter Quantity:

100

- +

Predict

Predicted Country: [17]

11. Conclusion:

I have done both regression and classification

Classification is my project requirement but as it is mentioned in the internship project I have done some Logistic Regression.