Julia Donato

Drug Persistency Project- Week 8: Data Problems

Individual Project- Julia Donato, julia.donato21@gmail.com, USA, Indiana University-Bloomington, Data Science

Github Repo Link:
https://github.com/julia-donato/DG-Final-Project

Problem Description:
Drug persistency is a challenge for pharmaceutical companies and understanding the factors that impact it can be difficult. My goal is to develop a machine learning model that can predict drug persistency based on physician prescription data.

Data Cleaning/Transformation Code:

```
def cleaning1(input_file, output_file):
    import pandas as pd
    import numpy as np
    # Load the dataset
    data = pd.read_excel(str(input_file), sheet_name=1, engine='openpyxl')

    #### Replace "Other/Unkown" with "NaN" for selected columns
    data.replace(['Other/Unknown', 'Unkown'], np.nan)

    #### Impute missing values with mode for each column
    for column in data.columns:
        data[column].fillna(data[column].mode()[0], inplace=True)
   #### Transforming Y and N variables to 0 and 1
    data.map({'Y': 1, 'N': 0})

    # Transforming the Age_Bucket variable to numeric
    data['Age_Bucket'] = data['Age_Bucket'].map({'>75': 0, '65-75': 1, '55-65': 2, '<55': 3})


    data.to_csv(str(output_file), index=False)



def cleaning2(input_file, output_file):
    import pandas as pd
    import numpy as np
    from sklearn.tree import DecisionTreeRegressor
    from sklearn.preprocessing import LabelEncoder, OneHotEncoder
    # Load the dataset
    data = pd.read_excel(str(input_file), sheet_name=1, engine='openpyxl')
```

```
# Elimination of variables with more than 40% missing values
data = data.drop(columns=['Risk_Segment_During_Rx',
                          'Tscore_Bucket_During_Rx',
                          'Change_T_Score',
                          'Change_Risk_Segment'])

# replacing the missing values into actual null values. "Unknown" => "NULL"
data.replace(["Other/Unknown", "Unknown"], np.nan)

# Transforming Y and N variables to 0 and 1
data.replace({'Y': 1, 'N': 0}, inplace=True)

# Transforming the Age_Bucket variable to numeric
data['Age_Bucket'] = data['Age_Bucket'].map({'>75': 0, '65-75': 1, '55-65': 2, '<55': 3})

# splitting the descriptive variables from the target variable
features = data.iloc[:, 2:]
target = data.Persistency_Flag

# transformations
label_encoder = LabelEncoder()
ohe = OneHotEncoder()

# fit transformations
label_encoder.fit(target)
ohe.fit(features)

# transform
features = ohe.transform(features).toarray()
target = label_encoder.transform(target)

# Train a decision tree to impute missing values
impute_indices = np.where(np.isnan(features))
impute_features = np.delete(features, impute_indices[0], axis=0)
impute_target = np.delete(target, impute_indices[0], axis=0)
tree_model = DecisionTreeRegressor(random_state=42)
tree_model.fit(impute_features, impute_target)
imputed_values = tree_model.predict(features[impute_indices])
features[impute_indices] = imputed_values

# Assigning the variables X and Y
X = features
Y = target
```

```
    # output file
    df = pd.concat([data.iloc[:, :2], pd.DataFrame(X)], axis=1)
    df.to_csv(str(output_file), index=False)
```