Model

December 3, 2023

[1]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.metrics import accuracy_score
import pandas as pd
```

[2]:
```
# Read data from Excel file into a Pandas DataFrame
file_path = 'dm_mimic_pathways.csv'
df = pd.read_csv(file_path)
```

[3]:
```
column_name_mapping = {'person_id': 'Person',
                       'race_concept_id': 'Race',
                       'gender_concept_id':'Gender',
                       'age_group':'Age Group',
                       'pathways':'Treatment Regimen'}

race_mapping = {8527: 'White/ Hispanic',
                8516: 'Black',
                8515: 'Asian',
                0:'Unknown',
                38003592: 'Asian',
                4077359: 'Other',
                4218674: 'Unknown',
                4188159: 'White/ Hispanic',
                38003599: 'Black',
                38003574: 'Asian',
                4212311: 'Asian',
                38003600: 'Black',
                8557: 'Other',
                38003584: 'Asian',
                38003578: 'Asian',
                4087921: 'Other',
                38003615: 'Other',
                38003581: 'Asian',
                8657: 'Other',
                38003579: 'Asian',
```

```
                 38003605: 'Black',
                 38003614: 'White',
                 4213463: 'White'}

gender_mapping = {8507: 'Male',
                  8532: 'Female'}

age_mapping = {'10 - 19': 'Teens',
               '20 - 29': 'Twenties',
               '30 - 39': 'Thirties',
               '40 - 49': 'Forties',
               '50 - 59': 'Fifties',
               '60 - 69': 'Sixties',
               '70 - 79': 'Seventies',
               '80 - 89': 'Eighties',
              '> 90': 'Nineties'}
```

[4]:
```
df = df.rename(columns=column_name_mapping)
df['Race'] = df['Race'].replace(race_mapping)
df['Gender'] = df['Gender'].replace(gender_mapping)
df['Age Group'] = df['Age Group'].replace(age_mapping)
df['Age Group'].fillna('Unknown', inplace=True)
```

[5]:
```
df = df[(df['Age Group'] != 'Unknown') & (df['Race'] != 'Unknown')]
```

[6]:
```
print(len(df))
n = 6
values_to_preserve = df['Treatment Regimen'].value_counts().head(n)
print(values_to_preserve)
```

```
1746
Treatment Regimen
19071700             463
19071700,40166274    197
40164929              73
40164930              62
40166274              61
19071700,40164929     47
Name: count, dtype: int64
```

[7]:
```
def preserve_or_change(value, value_set, replacement_value):
    return value if value in value_set else replacement_value
```

[8]:
```
df['Treatment Regimen'] = df['Treatment Regimen'].apply(lambda x:␣
 ↪preserve_or_change(x, values_to_preserve, 'Other'))
df.head(5)
len(df['Treatment Regimen'].unique())
```

[8]: 7

[9]:
```
X = df[['Age Group', 'Race', 'Gender']]
y = df['Treatment Regimen']
```

[10]:
```
preprocessor = ColumnTransformer(
    transformers=[
            ('cat', OneHotEncoder(), ['Age Group', 'Race', 'Gender'])
        ],
        remainder='passthrough'

)
pipeline = Pipeline([
    ('preprocessor', preprocessor),
    ('classifier', LogisticRegression(multi_class='multinomial', class_weight =␣
↪'balanced'))
])
```

[11]:
```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=42)
```

[12]:
```
# Train the model
pipeline.fit(X_train, y_train)
```

[12]:
```
Pipeline(steps=[('preprocessor',
                 ColumnTransformer(remainder='passthrough',
                                   transformers=[('cat', OneHotEncoder(),
                                                  ['Age Group', 'Race',
                                                   'Gender'])])),
                ('classifier',
                 LogisticRegression(class_weight='balanced',
                                    multi_class='multinomial'))])
```

[13]:
```
# Make predictions on the test set
y_pred = pipeline.predict(X_test)

# Evaluate the accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')

# Create a DataFrame with actual and predicted values
df_predictions = pd.DataFrame({
    'Actual': y_test,
    'Predicted': y_pred
})
```

```
print("Actual vs Predicted:")
print(df_predictions)
```

```
Accuracy: 0.12
Actual vs Predicted:
                  Actual          Predicted
408             19071700              Other
387             19071700           40164929
803             19071700           19071700
81                 Other           40164929
942             19071700  19071700,40164929
…                      …                  …
596    19071700,40166274           40166274
1710               Other              Other
894    19071700,40164929           40164930
1226               Other  19071700,40164929
1466               Other           40166274

[350 rows x 2 columns]
```

[14]:
```
# Access the one-hot encoder from the pipeline
encoder = pipeline.named_steps['preprocessor'].named_transformers_['cat']

# Get feature names after one-hot encoding
feature_names_after_encoding = list(encoder.get_feature_names_out(X.
 ↪select_dtypes(include=['object']).columns))

# Concatenate feature names with numeric features
all_feature_names = X.select_dtypes(include=['number']).columns.tolist() +␣
 ↪feature_names_after_encoding

# Access the model from the pipeline
model = pipeline.named_steps['classifier']

# Get coefficients
coefficients = model.coef_

# Display coefficients in a DataFrame
df_coefficients = pd.DataFrame(coefficients, columns=all_feature_names)
df_coefficients['Intercept'] = model.intercept_
df_coefficients['Class'] = model.classes_
df_coefficients.set_index('Class', inplace=True)

print("Coefficients:")
print(df_coefficients)
```

```
Coefficients:
                   Age Group_< 90  Age Group_Eighties  Age Group_Fifties  \
```

```
Class
19071700            -0.836426         -0.155413          0.361676
19071700,40164929   -0.680596          0.462782         -0.869301
19071700,40166274   -0.755326         -0.652004         -0.183342
40164929             0.286274          0.435708         -0.002231
40164930             0.509524          0.531097          0.445836
40166274             0.889572         -0.957442         -0.012424
Other                0.586978          0.335273          0.259785

                   Age Group_Forties  Age Group_Seventies  Age Group_Sixties  \
Class
19071700                    0.284115            -0.345830          -0.281696
19071700,40164929           0.542313             0.881072           0.995801
19071700,40166274           0.402388            -0.563268          -0.541726
40164929                    0.054001             0.112197           0.209576
40164930                   -0.787298            -0.047166           0.144199
40166274                   -0.603099            -0.494980          -0.502859
Other                       0.107581             0.457974          -0.023294

                   Age Group_Teens  Age Group_Thirties  Age Group_Twenties  \
Class
19071700                 -0.151121            0.317982            0.807451
19071700,40164929        -0.029563           -0.818414           -0.488420
19071700,40166274         0.718709            0.652825            0.923044
40164929                 -0.140869           -0.121502           -0.832597
40164930                 -0.050468            0.023009           -0.768359
40166274                 -0.264193            0.822605            1.124241
Other                    -0.082496           -0.876505           -0.765360

                   Race_Asian  Race_Black  Race_Other  Race_White  \
Class
19071700             0.248994   -0.022959    0.034765   -0.037835
19071700,40164929   -0.900061    0.150853   -0.979728    1.181880
19071700,40166274   -0.697013    0.301572    0.518592   -0.108390
40164929             0.051193   -0.138297    0.372902    0.233525
40164930             0.006826   -0.678208    1.267986   -0.704338
40166274             1.242767    0.577683   -1.295671   -0.624841
Other                0.047294   -0.190644    0.081154    0.059999

                   Race_White/ Hispanic  Gender_Female  Gender_Male  Intercept
Class
19071700                      -0.222226       0.028364    -0.027626   0.383200
19071700,40164929              0.542729      -0.319360     0.315033  -1.205051
19071700,40166274             -0.013461       0.000196     0.001104   0.412097
40164929                      -0.518768       0.066906    -0.066351   0.379076
40164930                       0.108109       0.016626    -0.016251  -0.152944
40166274                       0.101484       0.222303    -0.220882   0.268092
Other                          0.002133      -0.015035     0.014971  -0.084470
```