regression

June 15, 2024

[ ]:
```
import pandas as pd
from sklearn.preprocessing import LabelEncoder # type: ignore
from sklearn.linear_model import LogisticRegression # type: ignore
from sklearn.model_selection import train_test_split # type: ignore
from sklearn.metrics import classification_report # type: ignore
from sklearn.preprocessing import StandardScaler # type: ignore
import numpy as np
```

[ ]:
```
# Load the datasets
df = pd.read_csv('data/binary_dataset.csv')
display(df)
```

```
        Student ID  # Logins  # Content Reads  # Forum Reads  # Forum Posts  \
0    student000000       143              344             58              0
1    student000001        70              342              0              0
2    student000002        42              219              0              0
3    student000003        92              271              2              0
4    student000004       116              379              0              0
..             …       …                …              …              …
481  student000481        98              281              0              0
482  student000482        85              258              1              0
483  student000483        99              206              0              0
484  student000484        51              158              0              0
485  student000485        89              289              0              0

     # Quiz Reviews before submission  Assignment 1 lateness indicator  \
0                                   3                                0
1                                   4                                0
2                                   3                                0
3                                   6                                0
4                                   1                                0
..                                …                                …
481                                 1                                0
482                                 2                                0
483                                 6                                0
484                                 2                                0
485                                 0                                0
```

```
     Assignment 2 lateness indicator  Assignment 3 lateness indicator  \
0                                  0                                 0
1                                  0                                 0
2                                  0                                 0
3                                  0                                 0
4                                  0                                 0
..                               …                                 …
481                                0                                 0
482                                0                                 0
483                                0                                 0
484                                0                                 0
485                                0                                 0

     Assignment 1 duration to submit (in hours)  …  Engagement Level  \
0                                    178.166667  …                 H
1                                    294.033333  …                 M
2                                    169.600000  …                 M
3                                    341.150000  …                 M
4                                    325.500000  …                 M
..                                          …  …               …
481                                  175.200000  …                 H
482                                  127.633333  …                 H
483                                  177.850000  …                 M
484                                  125.900000  …                 H
485                                  313.183333  …                 M

     Quiz01 [10]  Assignment01 [8] Midterm Exam [20]  Assignment02 [12]  \
0             95                91                70                  90
1             85                76                65                  61
2             85                41                73                  61
3             80                78                80                  79
4             85                91                78                  80
..           …                 …               …                 …
481           90                85                93                  87
482           80                68                93                  70
483           80                86                93                  95
484           75                68                80                  76
485           85                81                88                  86

     Assignment03 [25]  Final Exam [35]  Course Grade  Total [100]  Class
0                   84                64            85           85      G
1                   73                64            76           76      G
2                   73                61            73           73      G
3                   79                57            80           79      G
4                   84                67            85           85      G
..                 …                 …             …          …    …
481                 81                77            92           92      G
482                 83                83            90           90      G
```

```
483                90                83                96                96      G
484                88                50                78                78      G
485                96                60                88                88      G

[486 rows x 23 columns]
```

[ ]:
```
# Encode the 'Class' target variable
label_encoder = LabelEncoder()
df['Class'] = label_encoder.fit_transform(df['Class'])

# Drop rows with missing values if any
df.dropna(inplace=True)
```

[ ]:
```
X = df.drop(columns=['Class', 'Student ID', 'Engagement Level', 'Course Grade',␣
 ↪'Total [100]'])  # Assuming 'Student ID' should be excluded
y = df['Class']
```

[ ]:
```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

[ ]:
```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3,␣
 ↪random_state=42)
```

[ ]:
```
# Fit the logistic regression model
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)
```

[ ]:
```
LogisticRegression(max_iter=1000)
```

[ ]:
```
# Get feature importance
feature_importance = np.abs(model.coef_[0])
feature_names = X.columns
importance_df = pd.DataFrame({'Feature': feature_names, 'Importance':␣
 ↪feature_importance})
importance_df = importance_df.sort_values(by='Importance', ascending=False)
print(importance_df)
```

```
                                    Feature  Importance
16                        Assignment03 [25]    1.485992
17                          Final Exam [35]    1.417478
12                              Quiz01 [10]    1.067126
13                         Assignment01 [8]    0.777754
15                        Assignment02 [12]    0.715703
6          Assignment 2 lateness indicator    0.551737
14                        Midterm Exam [20]    0.297650
4        # Quiz Reviews before submission    0.266007
1                           # Content Reads    0.249976
```

```
5                Assignment 1 lateness indicator    0.232207
2                                  # Forum Reads    0.227474
3                                  # Forum Posts    0.190200
7                Assignment 3 lateness indicator    0.137718
10    Assignment 3 duration to submit (in hours)    0.080883
8     Assignment 1 duration to submit (in hours)    0.076837
11  Average time to submit assignment (in hours)    0.051538
0                                       # Logins    0.029599
9     Assignment 2 duration to submit (in hours)    0.023930
```

[ ]:
```
# Make predictions
y_pred = model.predict(X_test)
```

[ ]:
```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      0.99      0.99       141
           1       0.71      1.00      0.83         5

    accuracy                           0.99       146
   macro avg       0.86      0.99      0.91       146
weighted avg       0.99      0.99      0.99       146
```