Read Data

In [ ]:
```
import pandas as pd
```

In [ ]:
```
df = pd.read_excel('Data Scientist Project Data Evaluation of Test Optional.xlsx')
```

In [ ]:
```
df.head()
```

Out[ ]:

	Starting Term	Low Income	School	Race/Ethnicity	Gender	HS GPA	FIRST_GENERATION_DESCR	Test Optional	First Fall GPA	First Spring GPA	Retention to Next Fall
0	Fall 2021	No	Dietrich Sch Arts and Sciences	Non-White	M	3.675	Not First Generation	Admitted without test scores	1.700	2.192	Not Retained
1	Fall 2021	No	Sch Computing and Information	Non-White	M	3.460	Unknown	Admitted without test scores	1.000	1.727	Not Retained
2	Fall 2021	No	College of Business Admin	Non-White	F	4.020	Not First Generation	Admitted without test scores	2.019	NaN	Not Retained
3	Fall 2021	No	Dietrich Sch Arts and Sciences	Non-White	F	3.740	Not First Generation	Admitted without test scores	3.462	3.304	Not Retained
4	Fall 2021	No	Dietrich Sch Arts and Sciences	Non-White	M	3.719	First Generation	Admitted without test scores	3.500	NaN	Not Retained

In [ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4869 entries, 0 to 4868
Data columns (total 11 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Starting Term           4869 non-null   object 
 1   Low Income              4869 non-null   object 
 2   School                  4869 non-null   object 
 3   Race/Ethnicity          4869 non-null   object 
 4   Gender                  4767 non-null   object 
 5   HS GPA                  4859 non-null   float64
 6   FIRST_GENERATION_DESCR  4869 non-null   object 
 7   Test Optional           4869 non-null   object 
 8   First Fall GPA          4868 non-null   float64
 9   First Spring GPA        4742 non-null   float64
 10  Retention to Next Fall  4869 non-null   object 
dtypes: float64(3), object(8)
memory usage: 418.6+ KB
```

Data Pre-processing

Starting Term

In [ ]:
```
df['Starting Term'].isna().sum()
```

Out[ ]:
```
0
```

In [ ]:
```
df['Starting Term'].value_counts()
```

Out[ ]:
```
Fall 2021    4869
Name: Starting Term, dtype: int64
```

Low Income

In [ ]:
```
df['Low Income'].isna().sum()
```

Out[ ]:
```
0
```

In [ ]:
```
df['Low Income'].value_counts()
```

Out[ ]:
```
No     3995
Yes     874
Name: Low Income, dtype: int64
```

School

In [ ]:
```
df['School'].isna().sum()
```

Out[ ]:
```
0
```

In [ ]:
```
df['School'].value_counts()
```

Out[ ]:
```
Dietrich Sch Arts and Sciences    3386
Swanson School of Engineering      656
College of Business Admin          400
Sch Computing and Information      248
School of Nursing                  179
Name: School, dtype: int64
```

Race/Ethnicity

In [ ]:
```
df['Race/Ethnicity'].isna().sum()
```

Out[ ]:
```
0
```

In [ ]:
```
df['Race/Ethnicity'].value_counts()
```

Out[ ]:
```
White            2965
Non-White        1660
International     136
Unknown           108
Name: Race/Ethnicity, dtype: int64
```

In [ ]:
```
df = df[df['Race/Ethnicity']!='Unknown']
df = df.reset_index(drop=True)
```

In [ ]:
```
df['Race/Ethnicity'].value_counts()
```

Out[ ]:
```
White            2965
Non-White        1660
International     136
Name: Race/Ethnicity, dtype: int64
```

Gender

In [ ]:
```
df['Gender'].isna().sum()
```

Out[ ]:
```
102
```

In [ ]:
```
df = df.dropna(axis=0,subset=['Gender'])
df = df.reset_index(drop=True)
```

In [ ]:
```
df['Gender'].isna().sum()
```

Out[ ]:
```
0
```

In [ ]:
```
df['Gender'].value_counts()
```

Out[ ]:
```
F    2793
M    1866
Name: Gender, dtype: int64
```

HS GPA

In [ ]:
```
df['HS GPA'].isna().sum()
```

Out[ ]:
```
9
```

In [ ]:
```
df = df.dropna(axis=0,subset=['HS GPA'])
df = df.reset_index(drop=True)
```

In [ ]:
```
df['HS GPA'].isna().sum()
```

Out[ ]:
```
0
```

In [ ]:
```
df['HS GPA'].describe()
```

Out[ ]:
```
count    4650.000000
mean        4.102985
std         0.427446
min         0.000000
25%         3.841000
50%         4.140000
75%         4.410000
max         6.656000
Name: HS GPA, dtype: float64
```

FIRST_GENERATION_DESCR

In [ ]:
```
df['FIRST_GENERATION_DESCR'].isna().sum()
```

Out[ ]:
```
0
```

In [ ]:
```
df['FIRST_GENERATION_DESCR'].value_counts()
```

Out[ ]:
```
Not First Generation    3369
First Generation         663
Unknown                  618
Name: FIRST_GENERATION_DESCR, dtype: int64
```

In [ ]:
```
df = df[df['FIRST_GENERATION_DESCR'] != 'Unknown']
df = df.reset_index(drop=True)
```

In [ ]:
```
df['FIRST_GENERATION_DESCR'].value_counts()
```

Out[ ]:
```
Not First Generation    3369
First Generation         663
Name: FIRST_GENERATION_DESCR, dtype: int64
```

Test Optional

In [ ]:
```
df['Test Optional'].isna().sum()
```

Out[ ]:
```
0
```

In [ ]:
```
df['Test Optional'].value_counts()
```

Out[ ]:
```
Admitted with test scores       2083
Admitted without test scores    1949
Name: Test Optional, dtype: int64
```

First Fall GPA

In [ ]:
```
df['First Fall GPA'].isna().sum()
```

Out[ ]:
```
1
```

In [ ]:
```
df = df.dropna(axis=0,subset=['First Fall GPA'])
df = df.reset_index(drop=True)
```

In [ ]:
```
df['First Fall GPA'].isna().sum()
```

Out[ ]:
```
0
```

In [ ]:
```
df['First Fall GPA'].describe()
```

Out[ ]:
```
count    4031.000000
mean        3.182444
std         0.791998
min         0.000000
25%         2.833000
50%         3.404000
75%         3.769000
max         4.000000
Name: First Fall GPA, dtype: float64
```

First Spring GPA

In [ ]:
```
df['First Spring GPA'].isna().sum()
```

Out[ ]:
```
97
```

In [ ]:
```
df = df.dropna(axis=0,subset=['First Spring GPA'])
df = df.reset_index(drop=True)
```

In [ ]:
```
df['First Spring GPA'].isna().sum()
```

Out[ ]:
```
0
```

In [ ]:
```
df['First Spring GPA'].describe()
```

Out[ ]:
```
count    3934.000000
mean        3.178097
std         0.776574
min         0.000000
25%         2.859000
50%         3.359000
75%         3.750000
max         4.000000
Name: First Spring GPA, dtype: float64
```

Retention to Next Fall

In [ ]:
```
df['Retention to Next Fall'].isna().sum()
```

Out[ ]:
```
0
```

In [ ]:
```
df['Retention to Next Fall'].value_counts()
```

Out[ ]:
```
Retained        3718
Not Retained     216
Name: Retention to Next Fall, dtype: int64
```

Regression Analysis

In [ ]:
```
import statsmodels.api as sm
```

Feature encoding

In [ ]:
```
# Binary features encoding

df['Low Income'] = df['Low Income'].map({'Yes': 1, 'No': 0})
df['Gender'] = df['Gender'].map({'M':1, 'F':0})
df['FIRST_GENERATION_DESCR'] = df['FIRST_GENERATION_DESCR'].map({'First Generation':1, 'Not First Generation':0})
df['Test Optional'] = df['Test Optional'].map({'Admitted with test scores':1, 'Admitted without test scores':0})
df['Retention to Next Fall'] = df['Retention to Next Fall'].map({'Retained':1, 'Not Retained':0})
```

In [ ]:
```
# Multi-levels features encoding

df = pd.get_dummies(df, columns=['School', 'Race/Ethnicity'], drop_first=True)
```

In [ ]:
```
df.head()
```

Out[ ]:

	Starting Term	Low Income	Gender	HS GPA	FIRST_GENERATION_DESCR	Test Optional	First Fall GPA	First Spring GPA	Retention to Next Fall	School_Dietrich Sch Arts and Sciences	School_Sch Computing and Information	School_School of Nursing	School_Swanson School of Engineering	Race/Ethnicity_Non-White	Race/Ethnicity_White
0	Fall 2021	0	1	3.675	0	0	1.700	2.192	0	1	0	0	0	1	0
1	Fall 2021	0	0	3.740	0	0	3.462	3.304	0	1	0	0	0	1	0
2	Fall 2021	0	1	4.410	0	0	3.397	3.132	0	0	0	0	1	1	0
3	Fall 2021	0	0	3.745	0	0	2.000	0.000	0	1	0	0	0	1	0
4	Fall 2021	0	0	4.330	0	0	3.481	0.000	0	1	0	0	0	1	0

In [ ]:
```
# add a constant to the dataset

df['constant'] = 1
```

Set Up X AND Y

In [ ]:
```
X_features = ['Low Income', 'Gender', 'HS GPA', 'FIRST_GENERATION_DESCR', 'Test Optional', 'School_Dietrich Sch Arts and Sciences',
       'School_Sch Computing and Information', 'School_School of Nursing',
       'School_Swanson School of Engineering', 'Race/Ethnicity_Non-White',
       'Race/Ethnicity_White', 'constant']
```

In [ ]:
```
X = df[X_features]
```

In [ ]:
```
Y_fall_GPA = df['First Fall GPA']
Y_spring_GPA = df['First Spring GPA']
Y_retention = df['Retention to Next Fall']
```

Regression Fit

model 1 - First Fall GPA

In [ ]:
```
fall_GPA_model = sm.OLS(Y_fall_GPA, X)
```

In [ ]:
```
fall_results = fall_GPA_model.fit()
```

In [ ]:
```
print(fall_results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:         First Fall GPA   R-squared:                       0.243
Model:                            OLS   Adj. R-squared:                  0.241
Method:                 Least Squares   F-statistic:                     114.7
Date:                Wed, 05 Jul 2023   Prob (F-statistic):          5.53e-228
Time:                        16:32:07   Log-Likelihood:                -3991.6
No. Observations:                3934   AIC:                             8007.
Df Residuals:                    3922   BIC:                             8083.
Df Model:                          11
Covariance Type:            nonrobust
=========================================================================================================
                                            coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------------------------
Low Income                                -0.0798      0.029     -2.730      0.006      -0.137      -0.022
Gender                                    -0.0062      0.024     -0.262      0.794      -0.052       0.040
HS GPA                                     0.6779      0.027     25.292      0.000       0.625       0.730
FIRST_GENERATION_DESCR                    -0.1361      0.032     -4.257      0.000      -0.199      -0.073
Test Optional                              0.2249      0.023      9.899      0.000       0.180       0.269
School_Dietrich Sch Arts and Sciences      0.0672      0.040      1.673      0.094      -0.012       0.146
School_Sch Computing and Information       0.0994      0.063      1.587      0.113      -0.023       0.222
School_School of Nursing                   0.0838      0.068      1.227      0.220      -0.050       0.218
School_Swanson School of Engineering      -0.4384      0.047     -9.230      0.000      -0.531      -0.345
Race/Ethnicity_Non-White                  -0.2357      0.133     -1.766      0.078      -0.497       0.026
Race/Ethnicity_White                      -0.1507      0.133     -1.131      0.258      -0.412       0.111
constant                                   0.5163      0.168      3.081      0.002       0.188       0.845
==============================================================================
Omnibus:                      845.901   Durbin-Watson:                   1.870
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2319.312
Skew:                          -1.136   Prob(JB):                         0.00
Kurtosis:                       5.998   Cond. No.                         97.3
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

model 2 - First Spring GPA

In [ ]:
```
spring_GPA_model = sm.OLS(Y_spring_GPA, X)
```

In [ ]:
```
spring_results = spring_GPA_model.fit()
```

In [ ]:
```
print(spring_results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:       First Spring GPA   R-squared:                       0.189
Model:                            OLS   Adj. R-squared:                  0.187
Method:                 Least Squares   F-statistic:                     83.03
Date:                Wed, 05 Jul 2023   Prob (F-statistic):          3.60e-169
Time:                        16:32:15   Log-Likelihood:                -4175.1
No. Observations:                3934   AIC:                             8374.
Df Residuals:                    3922   BIC:                             8449.
Df Model:                          11
Covariance Type:            nonrobust
=============================================================================================================
                                              coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------------------------------
Low Income                                 -0.0772      0.031     -2.520      0.012      -0.137      -0.017
Gender                                     -0.0288      0.025     -1.166      0.244      -0.077       0.020
HS GPA                                      0.6332      0.028     22.546      0.000       0.578       0.688
FIRST_GENERATION_DESCR                     -0.1192      0.033     -3.558      0.000      -0.185      -0.053
Test Optional                               0.1505      0.024      6.324      0.000       0.104       0.197
School_Dietrich Sch Arts and Sciences       0.0043      0.042      0.103      0.918      -0.078       0.087
School_Sch Computing and Information       -0.0670      0.066     -1.021      0.307      -0.196       0.062
School_School of Nursing                    0.2058      0.072      2.878      0.004       0.066       0.346
School_Swanson School of Engineering       -0.3946      0.050     -7.930      0.000      -0.492      -0.297
Race/Ethnicity_Non-White                   -0.3134      0.140     -2.241      0.025      -0.588      -0.039
Race/Ethnicity_White                       -0.2418      0.140     -1.732      0.083      -0.516       0.032
constant                                    0.8479      0.176      4.829      0.000       0.504       1.192
==============================================================================
Omnibus:                     1367.216   Durbin-Watson:                   1.737
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             5779.007
Skew:                          -1.658   Prob(JB):                         0.00
Kurtosis:                       7.925   Cond. No.                         97.3
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

model 3 - Retention

In [ ]:
```
retention_model = sm.Logit(Y_retention, X)
```

In [ ]:
```
retention_results = retention_model.fit()
```

```
Optimization terminated successfully.
         Current function value: 0.204819
         Iterations 8
```

In [ ]:
```
print(retention_results.summary())
```

Logit Regression Results

Dep. Variable:	Retention to Next Fall	No. Observations:	3934
Model:	Logit	Df Residuals:	3922
Method:	MLE	Df Model:	11
Date:	Wed, 05 Jul 2023	Pseudo R-squ.:	0.03712
Time:	16:32:19	Log-Likelihood:	-805.76
converged:	True	LL-Null:	-836.82
Covariance Type:	nonrobust	LLR p-value:	3.733e-09

	coef	std err	z	P>|z|	[0.025	0.975]
Low Income	-0.4501	0.171	-2.629	0.009	-0.786	-0.115
Gender	-0.2176	0.152	-1.432	0.152	-0.515	0.080
HS GPA	0.6524	0.148	4.419	0.000	0.363	0.942
FIRST_GENERATION_DESCR	-0.5881	0.178	-3.305	0.001	-0.937	-0.239
Test Optional	-0.1034	0.149	-0.692	0.489	-0.396	0.189
School_Dietrich Sch Arts and Sciences	-0.3431	0.300	-1.145	0.252	-0.930	0.244
School_Sch Computing and Information	-0.5033	0.410	-1.228	0.220	-1.307	0.300
School_School of Nursing	0.5104	0.657	0.776	0.437	-0.778	1.799
School_Swanson School of Engineering	-0.3159	0.343	-0.920	0.357	-0.989	0.357
Race/Ethnicity_Non-White	-1.1120	1.040	-1.069	0.285	-3.151	0.927
Race/Ethnicity_White	-1.0849	1.039	-1.044	0.297	-3.122	0.952
constant	1.9713	1.195	1.649	0.099	-0.372	4.314