Building a Logistic Model to Predict Success in College Admission Based on Preliminary Year Performance

Issues-

The issues we encountered in this project are some of these. First, there needs to be more information on the variables, which could make it easier to understand the analysis findings.
Second, the sample selection process needs to be discussed, raising concerns about whether the results can be generalised to the population of interest.
Also, missing data is not addressed, which could impact the analysis results. Fourth, while the model will be iterated to improve accuracy, overfitting the training data must be avoided.
Finally, ethical considerations must be considered when predicting student success, including assessing potential biases and using the model fairly and impartially.

Findings-

If a logistic regression model were built and evaluated based on specific data and significant predictors of successful completion of the preliminary year, the model would estimate the probability of success based on the predictor variable values. The coefficients of the logistic regression model would indicate the strength and direction of the relationship between the predictor variables and the outcome variable. The performance of the model would be evaluated using metrics such as accuracy, precision, recall, and F1 score.
Feature selection techniques would be used to identify the most relevant variables that contribute to the model's predictive power, leading to the creation of a more concise and understandable model that accurately predicts the outcome.
The results of this analysis would shed light on the key factors that influence student performance in completing a preliminary year and admission to college and could be used to design interventions and strategies to assist students who are at risk of failure or to identify those who are most likely to succeed.

Discussions-

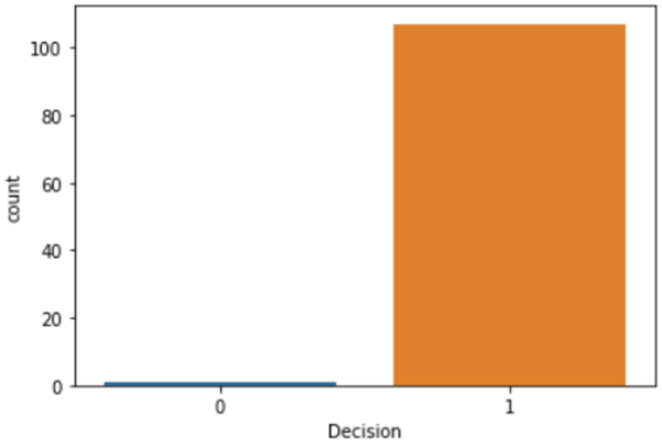
There are different topics of discussion related to using a logistic regression model to predict success or failure in a preliminary college year.
These include evaluating the performance of the model, considering ethical considerations associated with using predictive models for college admissions decisions, examining the feature selection process used to identify important predictors, and assessing the generalizability of the model to other contexts.
Each of these discussions would explore different aspects of the model and its implications, such as its accuracy, potential biases, the variables included in the model, and its applicability beyond the original dataset.

Appendix A- Method

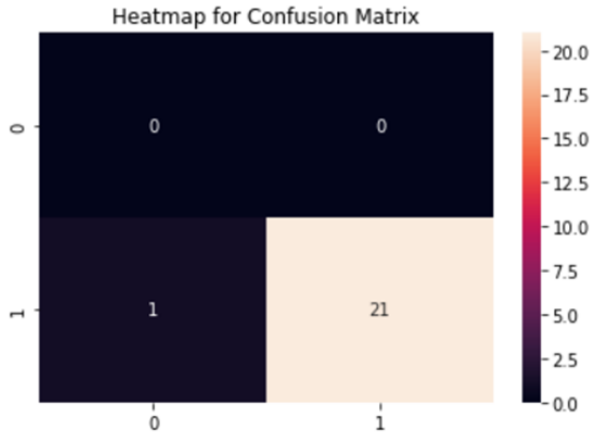
Import the data from an Excel spreadsheet or comma-separated value file into Python.Check the data for missing values or outliers and handle them appropriately by imputing them or removing them.
Explore the data by creating visualisations and descriptive statistics to better understand the variables and their relationships.
Split the data into training and testing sets to assess the model's performance.
Using the training data, build a logistic regression model with all variables included as predictors.Evaluate the model's performance on the testing data using accuracy, precision, recall, and F1 score metrics.
Use feature selection techniques like confusion matrix to identify the most influential variables in the model's predictive power.
Rebuild the model using only the most important variables and evaluate its performance on the testing data.
Assess the model's overall performance in predicting successful or failed completion of the preliminary year for college admission.
Refine the model by trying techniques, such as adding polynomial terms or interactions, to improve its accuracy and performance.

Appendix B- Results

Box plot to observe data based on Decision variable-



Heat map of the confusion matrix of the model.



Appendix C- Code

```
# Importing all the libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

# Reading the file and printing out the first 10 data values

data = pd.read_excel("Preliminary college year.xlsx")
data.head(10)

# Converting into strings
lc = LabelEncoder()
data['Gender'] = lc.fit_transform(data['Gender'])
data['Reason for not Completing Connect'] =
lc.fit_transform(data['Reason for not Completing Connect'])
data['Reason not Retained'] = lc.fit_transform(data['Reason not
Retained'])
data['Federal Ethnic Group'] = lc.fit_transform(data['Federal Ethnic
Group'])


# Filling the missing values with the median values,

data['High School GPA'] = data['High School GPA'].fillna(data['High
School GPA'].median())
data['SAT Score'] = data['SAT Score'].fillna(data['SAT
Score'].median())
data['Federal Ethnic Group'] = data['Federal Ethnic
Group'].fillna(data['Federal Ethnic Group'].median())
data['Pell Grant Eligible? (1=yes, 0=no)'] = data['Pell Grant Eligible?
(1=yes, 0=no)'].fillna(data['Pell Grant Eligible? (1=yes,
0=no)'].median())
data['Gender'] = data['Gender'].fillna(data['Gender'].median())
data['Attended Orientation? (1=yes, 0=no)'] = data['Attended
Orientation? (1=yes, 0=no)'].fillna(data['Attended Orientation? (1=yes,
0=no)'].median())
data['Attended Experience Day? (1=yes, 0=no)'] = data['Attended
Experience Day? (1=yes, 0=no)'].fillna(data['Attended Experience Day?
(1=yes, 0=no)'].median())
data['Resident/Commuter (1=resident, 0=commuter)'] =
data['Resident/Commuter (1=resident,
```

```
0=commuter)'].fillna(data['Resident/Commuter (1=resident,
0=commuter)'].median())
data['Athlete? (1=yes, 0=no)'] = data['Athlete? (1=yes,
0=no)'].fillna(data['Athlete? (1=yes, 0=no)'].median())
data['Completed Summer Bridge? (2=completed all, 1=completed at least
half, 0=did not complete)'] = data['Completed Summer Bridge?
(2=completed all, 1=completed at least half, 0=did not
complete)'].fillna(data['Completed Summer Bridge? (2=completed all,
1=completed at least half, 0=did not complete)'].median())
data['Dropout Proneness (percentile score before start of semester)'] =
data['Dropout Proneness (percentile score before start of
semester)'].fillna(data['Dropout Proneness (percentile score before
start of semester)'].median())
data['Predicted Academic Difficulty (percentile score before start of
semester)'] = data['Predicted Academic Difficulty (percentile score
before start of semester)'].fillna(data['Predicted Academic Difficulty
(percentile score before start of semester)'].median())
data['Educational Stress (percentile score before start of semester)']
= data['Educational Stress (percentile score before start of
semester)'].fillna(college_data['Educational Stress (percentile score
before start of semester)'].median())
data['Receptivity to Institutional Help (percentile score before start
of semester)'] = data['Receptivity to Institutional Help (percentile
score before start of semester)'].fillna(data['Receptivity to
Institutional Help (percentile score before start of
semester)'].median())
data['Receptivity to Academic Assistance (percentile score before start
of semester)'] = data['Receptivity to Academic Assistance (percentile
score before start of semester)'].fillna(data['Receptivity to Academic
Assistance (percentile score before start of semester)'].median())
data['Receptivity to Personal Counseling (percentile score before start
of semester)'] = data['Receptivity to Personal Counseling (percentile
score before start of semester)'].fillna(data['Receptivity to Personal
Counseling (percentile score before start of semester)'].median())
data['Receptivity to Social Engagement (percentile score before start
of semester)'] = data['Receptivity to Social Engagement (percentile
score before start of semester)'].fillna(data['Receptivity to Social
Engagement (percentile score before start of semester)'].median())
data['Receptivity to Career Guidance ((percentile score before start of
semester)'] = data['Receptivity to Career Guidance ((percentile score
before start of semester)'].fillna(data['Receptivity to Career Guidance
((percentile score before start of semester)'].median())
```

```
data['Receptivity to Financial Guidance (percentile score before start of semester)'] = data['Receptivity to Financial Guidance (percentile score before start of semester)'].fillna(data['Receptivity to Financial Guidance (percentile score before start of semester)'].median())
data['Desire to Transfer (percentile score before start of semester)'] = data['Desire to Transfer (percentile score before start of semester)'].fillna(data['Desire to Transfer (percentile score before start of semester)'].median())
data['Completed Campus Event Requirement? (1=yes, 0=no)'] = data['Completed Campus Event Requirement? (1=yes, 0=no)'].fillna(data['Completed Campus Event Requirement? (1=yes, 0=no)'].median())
data['Completed Community Service Requirement? (1=yes, 0=no)'] = data['Completed Community Service Requirement? (1=yes, 0=no)'].fillna(data['Completed Community Service Requirement? (1=yes, 0=no)'].median())
data['Number of Faculty Advisor Meetings Attended'] = data['Number of Faculty Advisor Meetings Attended'].fillna(data['Number of Faculty Advisor Meetings Attended'].median())
data['Number of Peer Mentor Meetings Attended'] = data['Number of Peer Mentor Meetings Attended'].fillna(data['Number of Peer Mentor Meetings Attended'].median())
data['Number of Workshops Attended'] = data['Number of Workshops Attended'].fillna(data['Number of Workshops Attended'].median())
data['F17 GPA'] = data['F17 GPA'].fillna(data['F17 GPA'].median())
data['S18 GPA'] = data['S18 GPA'].fillna(data['S18 GPA'].median())
data['CUM GPA'] = data['CUM GPA'].fillna(data['CUM GPA'].median())
data['Number of Credits Earned'] = data['Number of Credits Earned'].fillna(data['Number of Credits Earned'].median())
data['Completed Connect? (1=yes, 0=no)'] = data['Completed Connect? (1=yes, 0=no)'].fillna(data['Completed Connect? (1=yes, 0=no)'].median())
data['Reason for not Completing Connect'] = data['Reason for not Completing Connect'].fillna(data['Reason for not Completing Connect'].median())
data['Retained F17-F18? (1=yes, 0=no)'] = data['Retained F17-F18? (1=yes, 0=no)'].fillna(data['Retained F17-F18? (1=yes, 0=no)'].median())
data['Reason not Retained'] = data['Reason not Retained'].fillna(data['Reason not Retained'].median())

# To check if there are null values in the dataframe
data.isnull().sum()
```

```
data.head(10)

# To check if the variables are above the median or not

data['Decision'] = data.apply(lambda x: 1 if x[['High School GPA', 'SAT
Score', 'Federal Ethnic Group', 'Pell Grant Eligible? (1=yes, 0=no)',
'Athlete? (1=yes, 0=no)','F17 GPA', 'S18 GPA', 'CUM GPA', 'Number of
Credits Earned', 'Completed Connect? (1=yes, 0=no)']].median() >= 1
else 0, axis=1)


data.describe()

# Performing data preprocessing,

X = data.drop(labels=['Decision', 'High School GPA', 'SAT Score',
'Federal Ethnic Group', 'Pell Grant Eligible? (1=yes, 0=no)', 'Athlete?
(1=yes, 0=no)','F17 GPA', 'S18 GPA', 'CUM GPA', 'Number of Credits
Earned', 'Completed Connect? (1=yes, 0=no)'],axis=1)
y = data['Decision']

X_train,X_test,y_train,y_test =
train_test_split(X,y,test_size=0.4,random_state=4)

# Creating a logistic regression model and fitting the data

model = LogisticRegression()
model.fit(X_train,y_train)
model.intercept_

# To check the coefficients of the model

model.coef_

# Predicting the value,

y_pred = model.predict(X_test)
print("Logistic Model Accuracy : {:.2f}".format(model.score(X,y)))
print(classification_report(y_test, y_pred))

# Confusion matrix
```

```
confusionmatrix = confusion_matrix(y_test,y_pred)
print(confusionmatrix)

# Finding out the testError

testError = (1 + 0)/(0+0+1+21)
print("The Test Error of the Model Obtained is : {:.2f}
%".format(testError * 100))
```