CAP 5768: Homework 3

Place name here: Krushal Kalkani

Preliminary instructions

All analyses must be performed in Python using the packages that we discussed in class. Fill in all your solutions in the appropriate spaces provided in this Word document, and then upload a PDF copy of your solutions to Canvas. **Only PDF copies will be graded**.

Brief overview of the assignment

In this assignment, you will analyze the **`college`** dataset that is available with this assignment in Canvas and under the datasets module. This dataset has information on 18 features for 777 US colleges obtained from the 1995 issue of US News and World Reports. The columns in the dataset are:

Name	Description
`Private`	A factor with levels **`No`** and **`Yes`** including private or public university
`Apps`	Number of applications received
`Accept`	Number of applications accepted
`Enroll`	Number of new students enrolled
`Top10perc`	Percent new students from top 10% of high school class
`Top25perc`	Percent new students from top 25% of high school class
`F.Undergrad`	Number of full-time undergraduates
`P.Undergrad`	Number of part-time undergraduates
`Outstate`	Out-of-state tuition
`Room.Board`	Room and board costs
`Books`	Estimated book costs
`Personal`	Estimated personal spending
`PhD`	Percent of faculty with a Ph.D.
`Terminal`	Percent of faculty with a terminal degree
`S.F.Ratio`	Student/faculty ratio
`perc.alumni`	Percent alumni who donate
`Expend`	Instructional expenditure per student

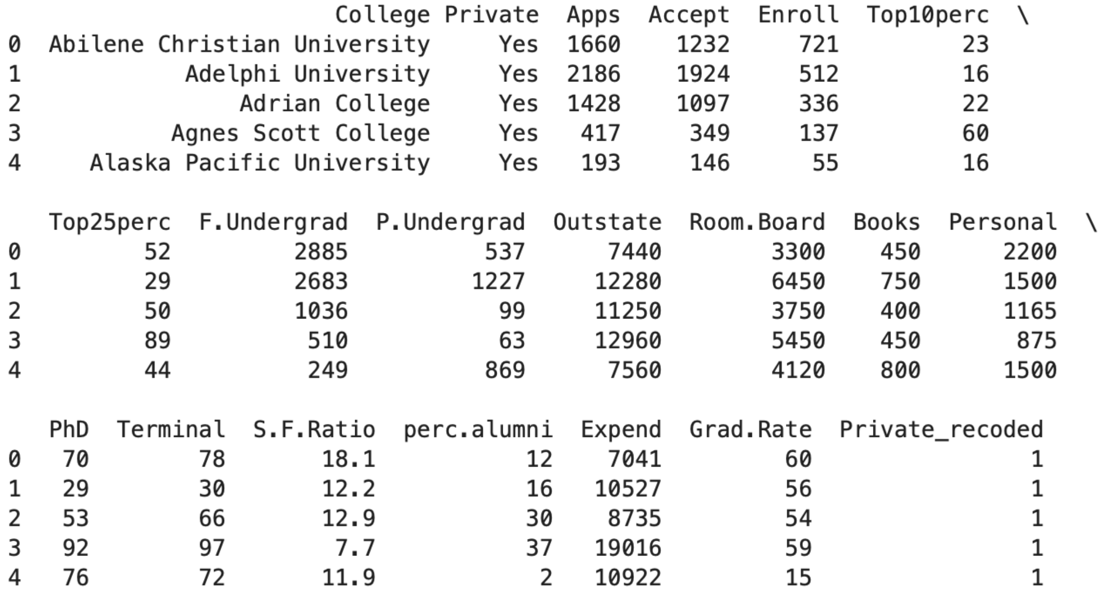
Grad.Rate Graduation rate

Questions and problems

1. [6%] Recode the binary feature `Private` with values `0` and `1` instead of `No` and `Yes` and store it in a new data frame called `College_recoded`.

Provide the code below:

```
import pandas as pd
file_path = 'college.csv'
college_data = pd.read_csv('college.csv')
college_data['Private_recoded'] = college_data['Private'].map({'Yes': 1, 'No': 0})
college_recoded = college_data.copy()
print(college_recoded.head())
```



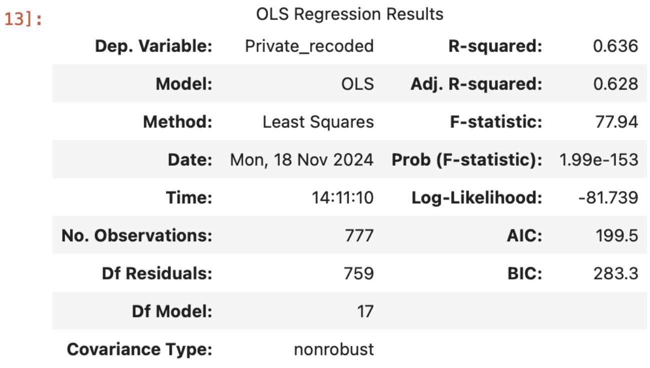
```
                      College Private  Apps  Accept  Enroll  Top10perc  \
0  Abilene Christian University     Yes  1660    1232     721         23
1            Adelphi University     Yes  2186    1924     512         16
2                Adrian College     Yes  1428    1097     336         22
3           Agnes Scott College     Yes   417     349     137         60
4     Alaska Pacific University     Yes   193     146      55         16

   Top25perc  F.Undergrad  P.Undergrad  Outstate  Room.Board  Books  Personal  \
0         52         2885          537      7440        3300    450      2200
1         29         2683         1227     12280        6450    750      1500
2         50         1036           99     11250        3750    400      1165
3         89          510           63     12960        5450    450       875
4         44          249          869      7560        4120    800      1500

   PhD  Terminal  S.F.Ratio  perc.alumni  Expend  Grad.Rate  Private_recoded
0   70        78       18.1           12    7041         60                1
1   29        30       12.2           16   10527         56                1
2   53        66       12.9           30    8735         54                1
3   92        97        7.7           37   19016         59                1
4   76        72       11.9            2   10922         15                1
```

. [

2 **10%]** Fit a multiple linear regression model to predict private school status with the 17 other features. Which feature is most important in this model, and what evidence tells you that?

Provide the code below:

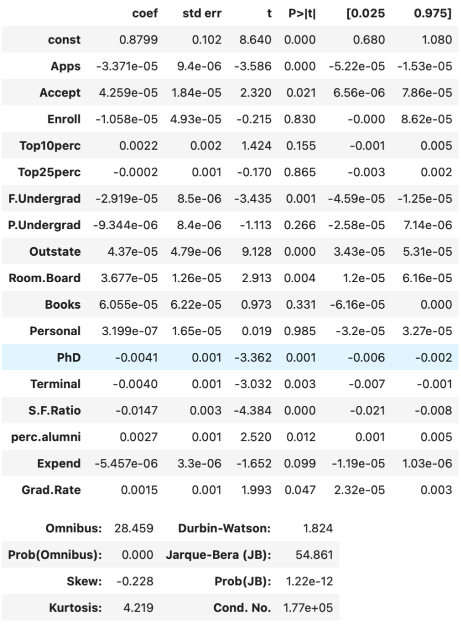
```
import statsmodels.api as sm
X = college_recoded.drop(columns=['College', 'Private', 'Private_recoded'])
y = college_recoded['Private_recoded']
X = sm.add_constant(X)
model = sm.OLS(y, X).fit()
model_summary = model.summary()
model_summary
```



13]:

OLS Regression Results

Dep. Variable:	Private_recoded	**R-squared:**	0.636
Model:	OLS	**Adj. R-squared:**	0.628
Method:	Least Squares	**F-statistic:**	77.94
Date:	Mon, 18 Nov 2024	**Prob (F-statistic):**	1.99e-153
Time:	14:11:10	**Log-Likelihood:**	-81.739
No. Observations:	777	**AIC:**	199.5
Df Residuals:	759	**BIC:**	283.3
Df Model:	17		
Covariance Type:	nonrobust		



	coef	std err	t	P>|t|	[0.025	0.975]
const	0.8799	0.102	8.640	0.000	0.680	1.080
Apps	-3.371e-05	9.4e-06	-3.586	0.000	-5.22e-05	-1.53e-05
Accept	4.259e-05	1.84e-05	2.320	0.021	6.56e-06	7.86e-05
Enroll	-1.058e-05	4.93e-05	-0.215	0.830	-0.000	8.62e-05
Top10perc	0.0022	0.002	1.424	0.155	-0.001	0.005
Top25perc	-0.0002	0.001	-0.170	0.865	-0.003	0.002
F.Undergrad	-2.919e-05	8.5e-06	-3.435	0.001	-4.59e-05	-1.25e-05
P.Undergrad	-9.344e-06	8.4e-06	-1.113	0.266	-2.58e-05	7.14e-06
Outstate	4.37e-05	4.79e-06	9.128	0.000	3.43e-05	5.31e-05
Room.Board	3.677e-05	1.26e-05	2.913	0.004	1.2e-05	6.16e-05
Books	6.055e-05	6.22e-05	0.973	0.331	-6.16e-05	0.000
Personal	3.199e-07	1.65e-05	0.019	0.985	-3.2e-05	3.27e-05
PhD	-0.0041	0.001	-3.362	0.001	-0.006	-0.002
Terminal	-0.0040	0.001	-3.032	0.003	-0.007	-0.001
S.F.Ratio	-0.0147	0.003	-4.384	0.000	-0.021	-0.008
perc.alumni	0.0027	0.001	2.520	0.012	0.001	0.005
Expend	-5.457e-06	3.3e-06	-1.652	0.099	-1.19e-05	1.03e-06
Grad.Rate	0.0015	0.001	1.993	0.047	2.32e-05	0.003

Omnibus:	28.459	**Durbin-Watson:**	1.824
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	54.861
Skew:	-0.228	**Prob(JB):**	1.22e-12
Kurtosis:	4.219	**Cond. No.**	1.77e+05

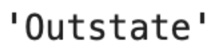
Provide the answer to the question below:

. [

3 5%] Fit a simple linear regression model to predict private school status based on the most important feature from Question 2. Is this feature still important in this model, and what evidence tells you that?

Provide the code below:

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler
import numpy as np

file_path = 'college.csv'
college_data = pd.read_csv(‘college.csv’)

college_data['Private_recoded'] = college_data['Private'].map({'Yes':
1, 'No': 0})

X = college_data.drop(columns=["College", "Private",
"Private_recoded"])
y = college_data["Private_recoded"]

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

model = LinearRegression()
model.fit(X_scaled, y)

feature_importance = np.abs(model.coef_)
most_important_feature_index = np.argmax(feature_importance)
most_important_feature = X.columns[most_important_feature_index]

most_important_feature
```



```
'Outstate'
```

Provide the answer to the question below:

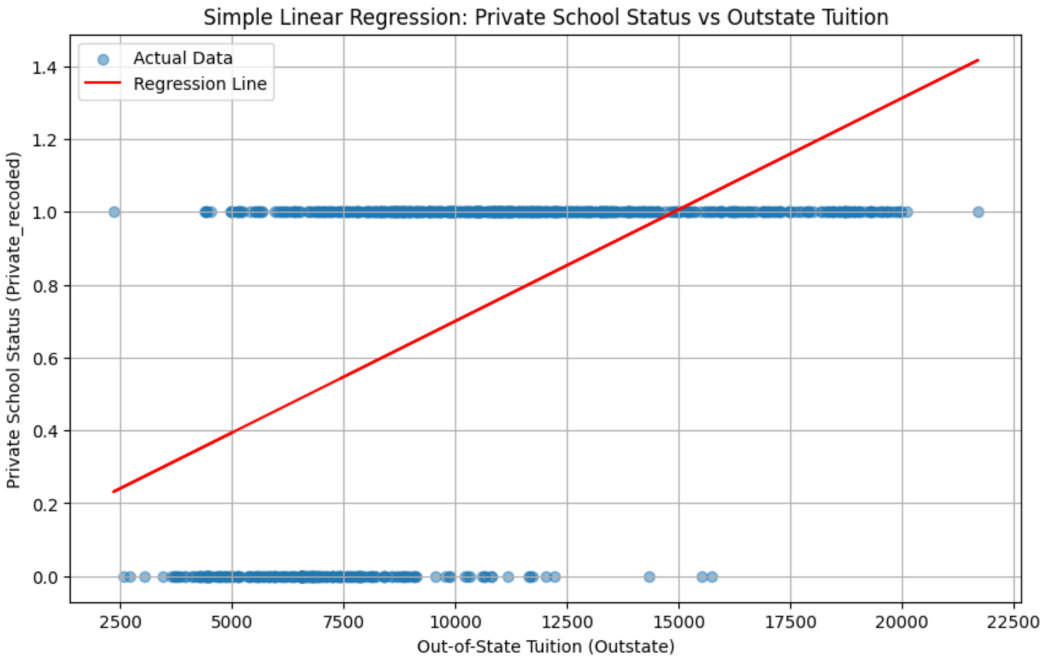
4 10%] Visualize the simple linear regression model from Question 3 using a scatter plot and the fitted linear model.

. [

Provide the code below:

```
plt.figure(figsize=(10, 6))

plt.scatter(college_data['Outstate'], college_data['Private_recoded'], alpha=0.5, label='Actual Data')

predicted_values = simple_model.predict(X_simple)
plt.plot(college_data['Outstate'], predicted_values, color='red', label='Regression Line')

plt.xlabel('Out-of-State Tuition (Outstate)')
plt.ylabel('Private School Status (Private_recoded)')
plt.title('Simple Linear Regression: Private School Status vs Outstate Tuition')
plt.legend()

plt.grid(True)
plt.show()
```

Provide the figure below:



. [10%]

5 Make predictions of the classes for the training dataset using your simple linear regression model from Question 3, and add these predictions to the data frame `College_recoded` that you created in Question 1. Create a confusion matrix and estimate classification accuracy for the training dataset.

Provide the code below:

```
 from sklearn.metrics import confusion_matrix, accuracy_score
college_data['Predicted_Private'] = (predicted_values >= 0.5).astype(int)
conf_matrix = confusion_matrix(college_data['Private_recoded'],
college_data['Predicted_Private'])
accuracy = accuracy_score(college_data['Private_recoded'], college_data['Predicted_Private'])
print("Confusion Matrix:\n", conf_matrix)
print("Accuracy:", accuracy)
```

Provide the confusion matrix below:

```
[[111 101]
 [ 38 527]]
```

- True Negatives: 111
- False Positives: 101
- False Negatives: 38
- True Positives: 527

Provide the accuracy estimate below:

Accuracy = 82.11%

6 **12%]** Perform the same operations as in Question 5, except use the multiple linear regression model from Question 2. Has the classifier improved in performance on the training data compared to the results from Question 5? Explain why you conclude this and provide a reason as to why this model did or did not improve upon the training error from Question 5.

Provide the code below:

```
 from sklearn.metrics import confusion_matrix, accuracy_score
```

. [

```
X_multiple = college_data.drop(columns=["College", "Private", "Private_recoded"])
X_multiple = sm.add_constant(X_multiple)  # Add intercept
y_multiple = college_data['Private_recoded']

multiple_model = sm.OLS(y_multiple, X_multiple).fit()
multiple_predicted_values = multiple_model.predict(X_multiple)

college_data['Predicted_Private_Multiple'] = (multiple_predicted_values >= 0.5).astype(int)

conf_matrix_multiple = confusion_matrix(college_data['Private_recoded'],
college_data['Predicted_Private_Multiple'])

accuracy_multiple = accuracy_score(college_data['Private_recoded'],
college_data['Predicted_Private_Multiple'])

print("Confusion Matrix:\n", conf_matrix_multiple)
print("Accuracy:", accuracy_multiple)
```

Provide the confusion matrix below:

```
Confusion Matrix:
 [[182  30]
 [ 18 547]]
```

Provide the accuracy estimate below:

Accuracy: 0.9382239382239382

Provide answers to the questions below:

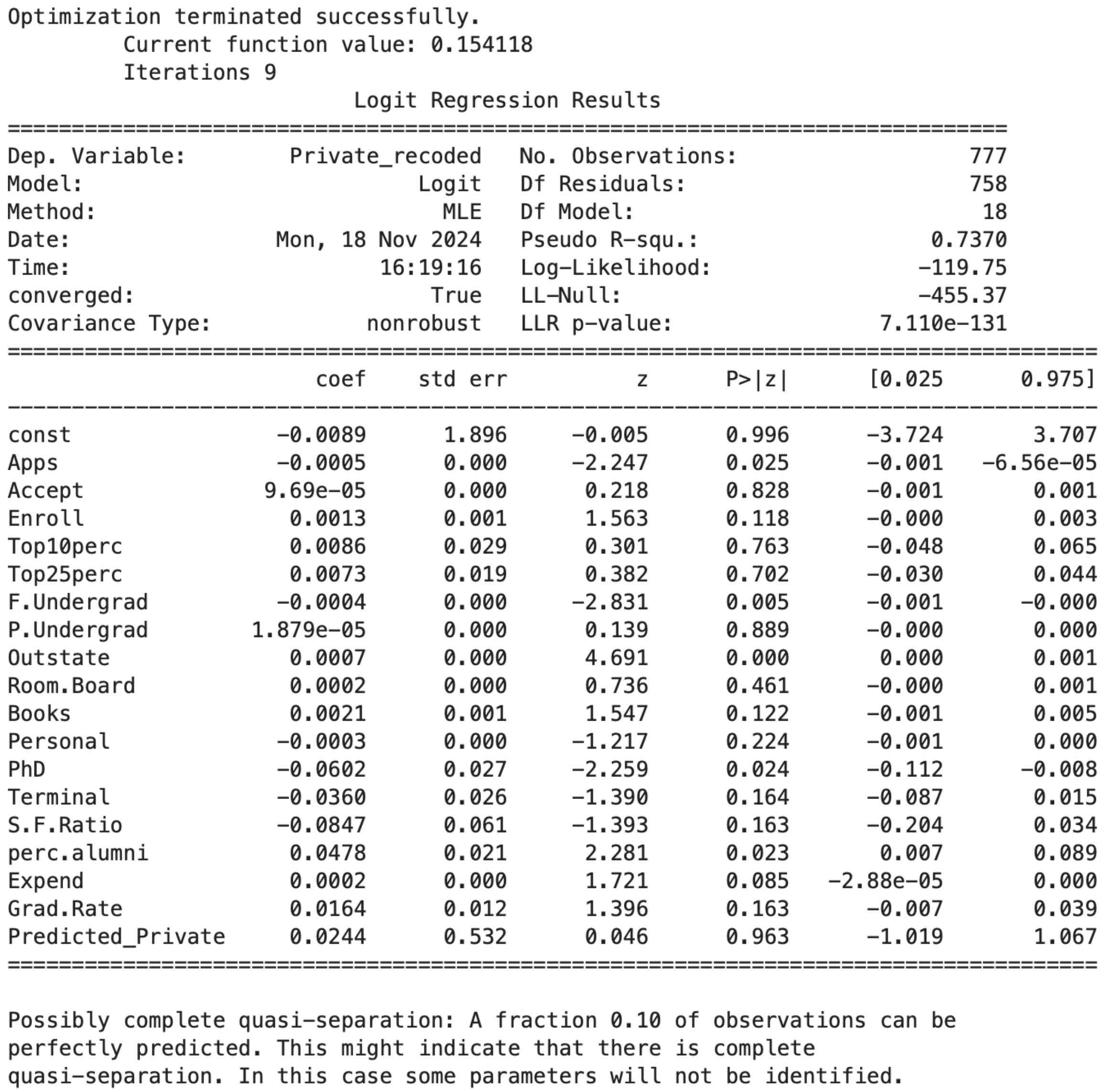
7 Fit a multiple logistic regression model to predict private school status with the 17 other features. Which feature is most important in this model, and what evidence tells you that?

Provide the code below:

```
 import statsmodels.api as sm

logit_model = sm.Logit(y_multiple, X_multiple)
logit_model_fitted = logit_model.fit()
```

. [10%]

```
print(logit_model_fitted.summary())
```



```
Optimization terminated successfully.
         Current function value: 0.154118
         Iterations 9
                           Logit Regression Results
==============================================================================
Dep. Variable:        Private_recoded   No. Observations:                  777
Model:                          Logit   Df Residuals:                      758
Method:                           MLE   Df Model:                           18
Date:                Mon, 18 Nov 2024   Pseudo R-squ.:                  0.7370
Time:                        16:19:16   Log-Likelihood:                -119.75
converged:                       True   LL-Null:                       -455.37
Covariance Type:            nonrobust   LLR p-value:                7.110e-131
=====================================================================================
                        coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------
const                -0.0089      1.896     -0.005      0.996      -3.724       3.707
Apps                 -0.0005      0.000     -2.247      0.025      -0.001   -6.56e-05
Accept              9.69e-05      0.000      0.218      0.828      -0.001       0.001
Enroll                0.0013      0.001      1.563      0.118      -0.000       0.003
Top10perc             0.0086      0.029      0.301      0.763      -0.048       0.065
Top25perc             0.0073      0.019      0.382      0.702      -0.030       0.044
F.Undergrad          -0.0004      0.000     -2.831      0.005      -0.001      -0.000
P.Undergrad        1.879e-05      0.000      0.139      0.889      -0.000       0.000
Outstate              0.0007      0.000      4.691      0.000       0.000       0.001
Room.Board            0.0002      0.000      0.736      0.461      -0.000       0.001
Books                 0.0021      0.001      1.547      0.122      -0.001       0.005
Personal             -0.0003      0.000     -1.217      0.224      -0.001       0.000
PhD                  -0.0602      0.027     -2.259      0.024      -0.112      -0.008
Terminal             -0.0360      0.026     -1.390      0.164      -0.087       0.015
S.F.Ratio            -0.0847      0.061     -1.393      0.163      -0.204       0.034
perc.alumni           0.0478      0.021      2.281      0.023       0.007       0.089
Expend                0.0002      0.000      1.721      0.085   -2.88e-05       0.000
Grad.Rate             0.0164      0.012      1.396      0.163      -0.007       0.039
Predicted_Private     0.0244      0.532      0.046      0.963      -1.019       1.067
=====================================================================================

Possibly complete quasi-separation: A fraction 0.10 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
```

. [

Provide the answer to the question below:

8 5%] Fit a simple logistic regression model to predict private school status based on the most important feature from Question 7. Is this feature still important in this model, and what evidence tells you that?

Provide the code below:

```
 import statsmodels.api as sm

X_logistic_simple = college_data[['Outstate']]
X_logistic_simple = sm.add_constant(X_logistic_simple)  # Add intercept

simple_logit_model = sm.Logit(y_multiple, X_logistic_simple)
simple_logit_model_fitted = simple_logit_model.fit()

print(simple_logit_model_fitted.summary())
```

Provide the answer to the question below:

9 Visualize the simple logistic regression model from Question 8 using a scatter plot and the fitted logistic model.

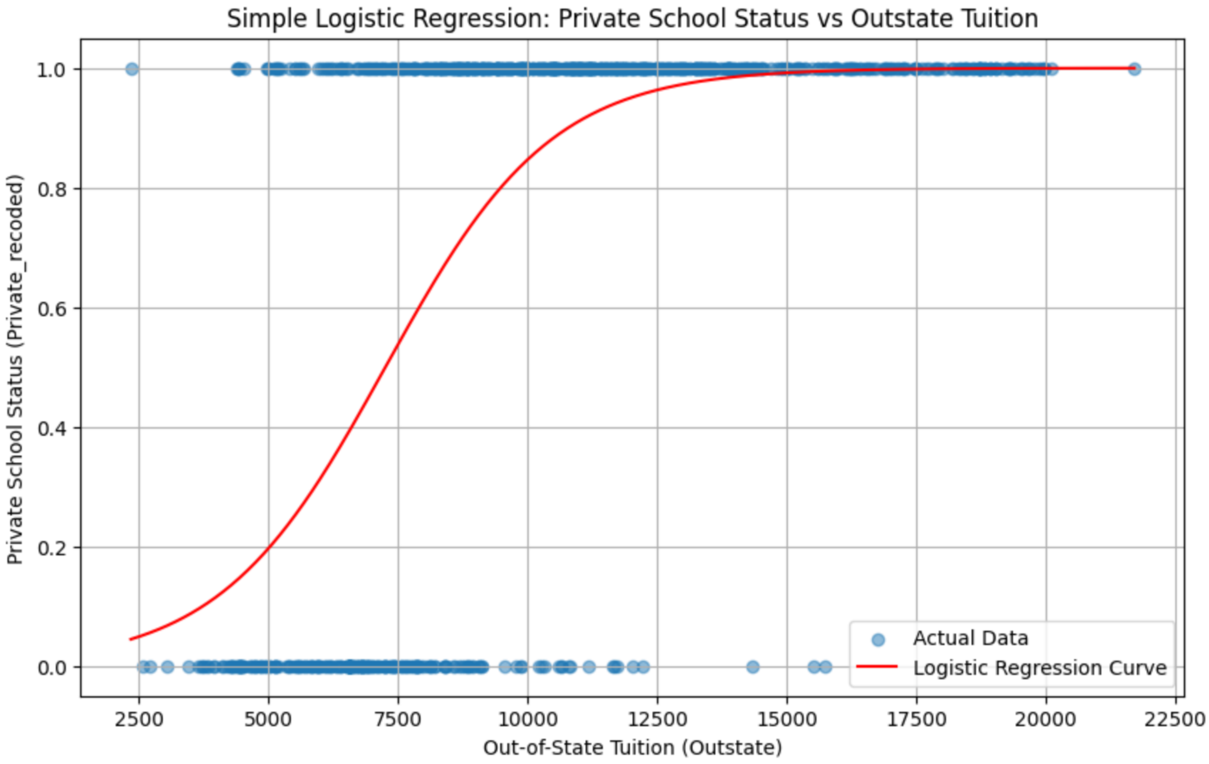
Provide the code below:

```
 import numpy as np
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))

plt.scatter(college_data['Outstate'], college_data['Private_recoded'], alpha=0.5, label='Actual Data')

outstate_values = np.linspace(college_data['Outstate'].min(), college_data['Outstate'].max(), 500)
X_logistic_curve = sm.add_constant(outstate_values)
logistic_predictions = simple_logit_model_fitted.predict(X_logistic_curve)

plt.plot(outstate_values, logistic_predictions, color='red', label='Logistic Regression Curve')
```

. [10%]

```
plt.xlabel('Out-of-State Tuition (Outstate)')
plt.ylabel('Private School Status (Private_recoded)')
plt.title('Simple Logistic Regression: Private School Status vs Outstate Tuition')
plt.legend()

plt.grid(True)
plt.show()
```

<u>Provide the figure below</u>:



10. **[10%]** Make predictions of the classes for the training dataset using your simple logistic regression model from Question 8 and add these predictions to the data frame **`College_recoded`** that you created in Question 1. Create a confusion matrix and estimate classification accuracy for the training dataset.

Provide the code below:

```
from sklearn.metrics import confusion_matrix, accuracy_score

college_data['Predicted_Private_Logistic'] =
(simple_logit_model_fitted.predict(X_logistic_simple) >= 0.5).astype(int)

conf_matrix_logistic = confusion_matrix(college_data['Private_recoded'],
college_data['Predicted_Private_Logistic'])

accuracy_logistic = accuracy_score(college_data['Private_recoded'],
college_data['Predicted_Private_Logistic'])

print("Confusion Matrix:\n", conf_matrix_logistic)
print("Accuracy:", accuracy_logistic)
```

Provide the confusion matrix below:

```
Confusion Matrix:
 [[140  72]
 [ 53 512]]
```

Provide the accuracy estimate below:

Accuracy = 83.91%

11. **[12%]** Perform the same operations as in Question 10, except use the multiple logistic regression model from Question 7. Has the classifier improved in training accuracy compared to the results of the multiple linear regression model from Question 6? Explain why you conclude this and provide a reason as to why this model did or did not improve upon the training error from Question 6.

Provide the code below:

```
from sklearn.metrics import confusion_matrix, accuracy_score

college_data['Predicted_Private_Multiple_Logistic'] = (logit_model_fitted.predict(X_multiple) >=
0.5).astype(int)

conf_matrix_multiple_logistic = confusion_matrix(college_data['Private_recoded'],
college_data['Predicted_Private_Multiple_Logistic'])

accuracy_multiple_logistic = accuracy_score(college_data['Private_recoded'],
college_data['Predicted_Private_Multiple_Logistic'])

print("Confusion Matrix:\n", conf_matrix_multiple_logistic)
print("Accuracy:", accuracy_multiple_logistic)
```

Provide the confusion matrix below:

```
Confusion Matrix:
[[191  21]
 [ 22 543]]
```

Provide the accuracy estimate below:

Accuracy: 0.9446589446589446

Provide answers to the questions below: