CP468 Project Report

Brandon Niles, Tony Yoon

Table of Contents

Executive Summary

Datasets

We are using two datasets from the StatLib—Datasets Archive called AAUP and USNEWS.

1. AAUP is from the March-April 1994 issue of Academe which contains information about faculty salaries for 1161 different American universities and colleges.

2. USNEWS is from 1995 US News and World Report's Guide to America's Best Colleges which contains information about the academic performances and expenses for students in 1302 different American colleges.

Methods

We are using two different methods two analyse the two datasets.

1. Decision Tree Classification for USNEWS.
    a. We determine a hierarchical order of variables based on how each variable affects another specific variable of the dataset.
2. Linear Regression for AAUP.
    a. We predict the coefficients of four linear equations which determine the average compensation for different ranked faculty members based on their salary and faculty size.

Results Summary

For the decision tree classification method, we determined that students who manage higher amounts of expenses may academically perform lower than students with lower costs.
For the regression models, we determined that there was correlation between the two independent variables with the dependent variable. As a result, our models based on predictive data had low mean absolute error, but high mean squared error due to possible large errors in predictions.

Introduction

In this report, we will be using methods from decision tree classification and linear regression to view correlation, predictions, and identify patterns amongst different features from the two datasets; aaup.data and usnews.data. Both of these datasets were provided from the StatLib—Datasets Archive for colleges and universities in America. Throughout our processes for analysing these datasets, we will be answering questions on feature selection, missing values, algorithm implementation, training and testing, potential outliers, and predicted feature relationships.

Project Description

Data Collection

AAUP

The AAUP dataset contains information about faculty salaries for 1161 different American colleges and universities. For the purpose of this project, the dataset was transformed and converted into a csv file. The raw data for the aaup.data file consisted of comma delimited fields with a single data line for each school. The order of the variables were given in the same format, which allowed us to convert it into a csv for fixed columns. The data was collected from the March-April 1994 issue of Academe.

Variable Descriptions

Feature Name	Description	Value Type
FICE	Federal ID number	Double
Name	Name of college or university	String
Postal	State postal code	String
Type	I, IIA, or IIB	String
Full Professor Salary	Average salary for full time professors at institution	Double
Associate Professor Salary	Average salary for associate professors at institution	Double
Assistant Professor Salary	Average salary for assistant professors at institution	Double
Full Rank Salary	Average salary for all ranks at institution	Double

Full Professor Comp	Average compensation for full time professors at institution	Double
Associate Professor Comp	Average compensation for associate professors at institution	Double
Assistant Professor Comp	Average compensation for assistant professors	Double
All Rank Comp	Average compensation for all ranks at institution	Double
Number of Full Professors	Number of full time professors at institution	Integer
Number of Associate Professors	Number of associate professors at institution	Integer
Number of Assistant Professors	Number of assistant professors at institution	Integer
Number of Instructors	Number of instructors at institution	Integer
Number of All Rank Faculty	Number of all ranks at institution.	Integer

USNEWS

The USNEWS dataset contains a wide variety of information for post secondary schools in the United States. Some of the information has to do with tuition and expenses, applications to the school, faculty information, and exam scores. This dataset was also converted to .csv format for use with Python libraries.

Variable Descriptions

Feature Name	Description	Value Type
FICE	Federal ID Number	Integer
Name	College name	String
Postal	State abbreviation	String
Indicator	Numerical indicator	Integer
Average Math SAT Score	Test score data	Integer

Average Verbal SAT Score	Test score data	Integer
Average Combined SAT Score	Test score data	Integer
Average ACT Score	Test score data	Integer
First Quartile Math SAT	Test score data	Integer
Third Quarter Math SAT	Test score data	Integer
First Quartile Verbal SAT	Test score data	Integer
Third Quartile Verbal SAT	Test score data	Integer
First Quartile ACT	Test score data	Integer
Third Quartile ACT	Test score data	Integer
Number of Applications Received	Application data	Integer
Number of Applicants Accepted	Application data	Integer
Number of New Students Enrolled	Application data	Integer
Top 10 percent HS	New students from top 10% of H.S. class	Integer
Top 25 percent HS	New students from top 25% of H.S. class	Integer

Number of Fulltime Undergrad	Student count info	Integer
Number of Parttime Undergrad	Student count info	Integer
In-State Tuition	Expenses info per student	Integer
Out-State Tuition	Expenses info per student	Integer
Room and Board Costs	Expenses info per student	Integer
Room Costs	Expenses info per student	Integer
Board Costs	Expenses info per student	Integer
Additional Fees	Expenses info per student	Integer
Estimated Book Costs	Expenses info per student	Integer
Estimated Personal Spending	Expenses info per student	Integer
Percent of Faculty with Ph.D	Faculty information	Float
Percent of Faculty with Terminal Degree	Faculty information	Float
Student Faculty Ratio	Faculty information	Float
Percent Alumni Donate	Faculty information	Float
Instructional	Faculty information / college expenses	Integer

Expenditure per Student		
Graduation Rate	% of students graduating	Integer

Feature Selection

AAUP

Linear regression modelling will be used to analyse the aaup dataset. When dealing with the different features and coefficients in linear regression models, we need to ensure that the independent variables do not have high collinearity values. High correlation in linear regression models can cause problems with multicollinearity and make regression coefficients unreliable, and as a result, would need to remove one of the highly collinear variables. Using the Pearson correlation method, we were able to create a map of the correlation coefficients in the correlationCoefficient.csv file. We can see that many of the independent variables have high correlation coefficients, which would force us to remove many of the variables. However, rather than limiting the features that we work with, we can generate multiple linear regression models based on independent variables with lower correlation coefficients.

For the aaup dataset, we have decided to use the following 12 features: full professor salary, associate professor salary, assistant professor salary, all rank salary, full professor comp, associate professor comp, assistant professor comp, all rank comp, number of full professors, number of associate professors, number of assistant professors, number of all rank faculty. The following features will be used to determine the linear regression models predictions for professor compensations based on their respective salaries and faculty size.

USNEWS

Decision Tree Classification will be used to analyze the usnews dataset. Due to the nature of decision trees, the classification algorithm runs into trouble when dealing with missing values. Unfortunately, usnews contains plenty of rows with missing values. The dataset contains 1302 rows. However, if we were to remove all rows containing a missing entry, we would be left with only 154 rows. After analyzing the dataset further, we determined that most of the missing values originate in a few columns (with some exceptions of course). The resulting solution was to first remove the problematic columns entirely, and then remove all rows containing a missing entry. This solution yielded 940 total rows, a staggering 786 more than the initial method.

The columns in question that were removed were:
"Row", "FICE", "Name", "Postal", "Indicator", "Average Math SAT Score", "Average Verbal SAT Score", "Average Combined SAT Score", "Average ACT Score", "First Quartile Math SAT", "Third Quarter Math SAT", "First Quartile Verbal SAT", "Third Quartile Verbal SAT", "First Quartile ACT", "Third Quartile ACT", "Top 10 percent HS", "Top 25 percent HS", "Room Costs", "Board Costs", "Additional Fees", "Percent Alumni Donate"

Unfortunately, we would have liked to have used these features in the analysis, but they were too unreliable in terms of missing values. Some of these columns had roughly 70% missing values. Attempting to include these features as factors in analysis would have led to unreliable results.

The remaining features after removal were:

Number of Applications Received
Number of Applicants Accepted
Number of New Students Enrolled
Number of Fulltime Undergrad
Number of Parttime Undergrad
In-State Tuition
Out-State Tuition
Room and Board Costs
Estimated Book Costs
Estimated Personal Spending
Percent of Faculty with Ph.D
Percent of Faculty with Terminal Degree
Student Faculty Ratio
Instructional Expenditure per Student
Majority Passing

Everything discussed in this subsection can be seen as demonstrated in the test_data.py file:

```
data/usnews.csv has 36 columns and 1302 rows.
If we remove rows with missing values we get 36 columns and 154 rows
However, if we first remove problematic columns (consistantly missing values among rows), we get:
15 columns and 940 rows
Thus we have 786 more entries to analyze without causing issues with missing data
```

Methodology

For the purpose of analyzing the dataset, we have opted to use 2 different methods.

Decision Tree Classification

Description

Decision tree classification is the concept of taking a set of data and determining a hierarchical order of variables based on how each variable affects another specific variable, variable x. In this scenario, variable x is the classification variable, being either true or false. The resulting structure as the name suggests is a tree, where each internal node is a decision node representing a comparison of a given variable in the tree, and each leaf node

is a given classification, indicating the expected label that an input value should take. The tree structure forms a hierarchy of data such that the tree is organised where internal nodes with the least depth are highest in information gain for a particular path. Two important terms are essential for classification trees:

Gini Impurity

Gini Impurity is one such method of determining how “pure” a specific set of data is at a given split. Sets of data that are greatly mixed are higher in impurity, resulting in a higher Gini Impurity value - the opposite is true for sets of data with little mixing. This test value is very important for later stages for determining the information gain of a specific split, as both sides (true and false) of a split must be evaluated. The general form of Gini Impurity is as follows:

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

Where p_i is the probability of element i.

There are other methods of determining impurity, such as entropy. We have opted out of using entropy calculations as it contains logarithmic functions which are typically higher in processing time. This is an especially important consideration when considering the size of potential test and training datasets.

See gini_impurity() method in decision_tree.py for detailed implementation.

Information Gain

Information gain is a method of comparing potential splits with one another. Decision trees will aim to pick the split with the highest amount of information gain. Information gain is a double value, ranging from 0.0 to 1.0. The calculation equation takes an input from the Gini Impurity score, comparing the results to that of the children of a split. The method can be seen in detail below:

```
def information_gain(node, right, left):
    right_weight = len(right) / len(node) #ratio of child to parent
    left_weight = len(left) / len(node)
    information = gini_impurity(node) - (right_weight * gini_impurity(right)) - \
        (left_weight * gini_impurity(left))

    return information
```

Adaptation

Best Split

Because each column of the chosen dataset can take on a wide range of values, the classification branches for each column must be determined. This is accomplished by looping through every value for a column in the test data to determine which value provides the best split (the split with the most information gain).

Left child: results where column value is false in decision node
Right child: results where column value is true in decision node

Classifier

When working with classification trees, it is important to determine the classification variable. In our case, we chose to use “Graduation Rate”. This column would act as our classification variable, where data points are classed as one of the following:

Positive classification: Graduation rate >= 50 -> TRUE
Negative classification: Graduation rate < 50 -> FALSE

As a result, the “Graduation Rate” column name has been renamed to “Majority Passing”, and entries in the column are replaced with “True” or “False” based on the condition specified above. This data modification is interpreted internally (without modifying the original .csv file directly), as can be seen in data_loading.py (see make_tree_data() method). Thus, the goal of using a decision tree classifier on this dataset is to determine how the remaining features in the dataset influence the result of Majority Passing

Linear Regression

Linear regression is a form of data analysis used to predict the coefficients of a linear equation; dependent variables based on the value of related independent variables. The regression model fits a straight line that minimises the discrepancies between predicted and actual output values.

In the case of the AAUP dataset, we’ll be using multiple instances of multiple linear regression to determine and interpret the relation between a single dependent variable to a single independent variable. As mentioned in previous sections, the data set consists of independent salaries and number of staff in different university faculty positions, as well as dependent compensations.

Clean Data

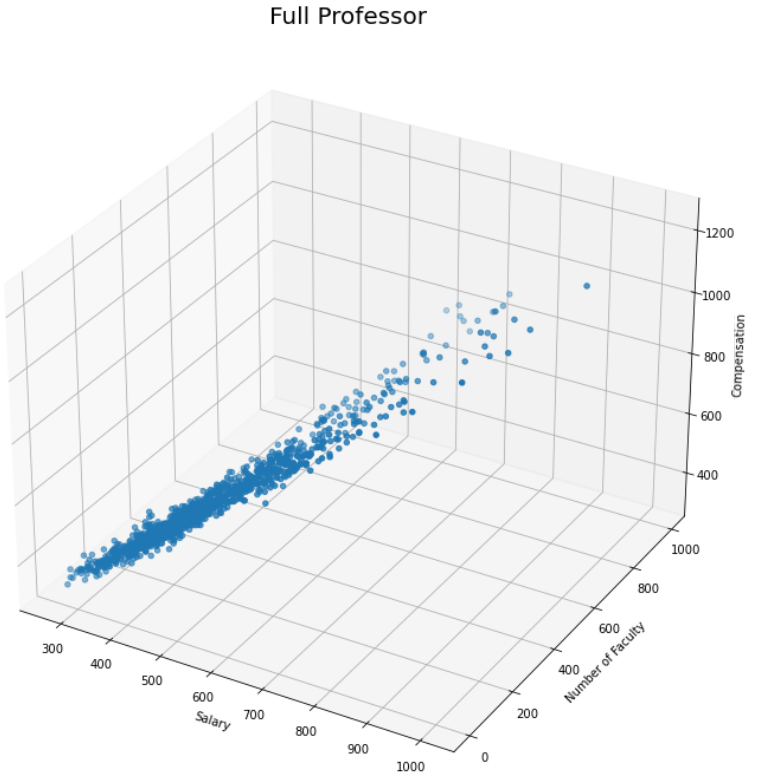
Before proceeding with building linear regression models for the aaup dataset, we must first prepare the data to ensure that we get the most accurate regression coefficients. This includes:

1. Linear assumptions (dependent vs independent variables)
2. Removing noise (outliers)
3. Removing multicollinearity

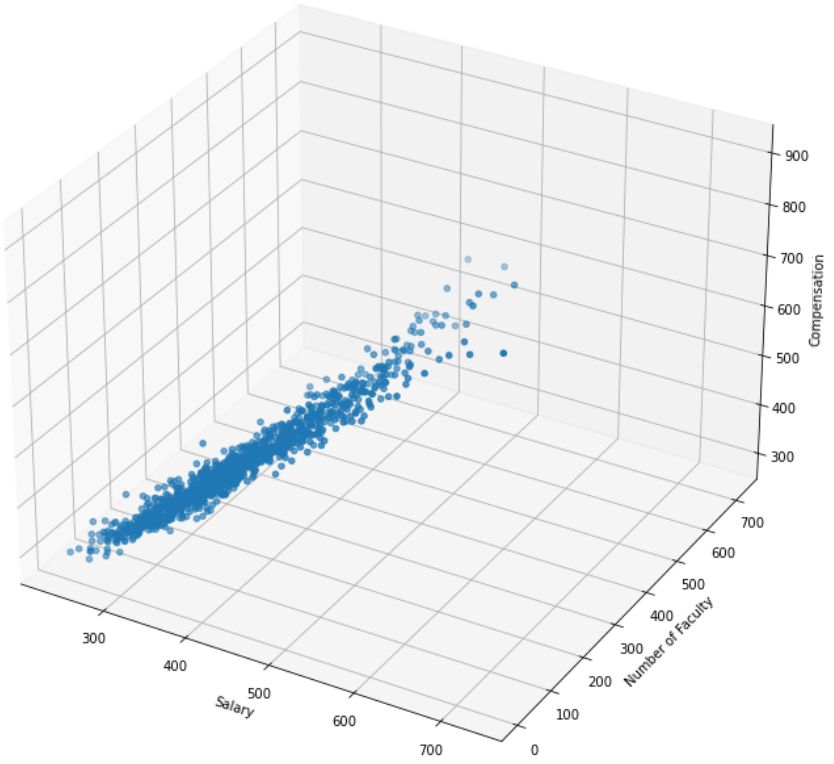
We first clean up the dataset by removing any incomplete entries with the dropna() function from the python pandas library. From this, we go from 1161 school's in the data down to 1074. We then determine what we would like to look for in this dataset.

Of the features available, we can see that end of year compensations for different ranking professors can be interpreted as dependent on the independent variables of their respective salaries and faculty size. We can set our independent variables as Professor Salary and Number of Professors for the different ranks, and our dependent variable as the Professor Comp for each respective type of professor.

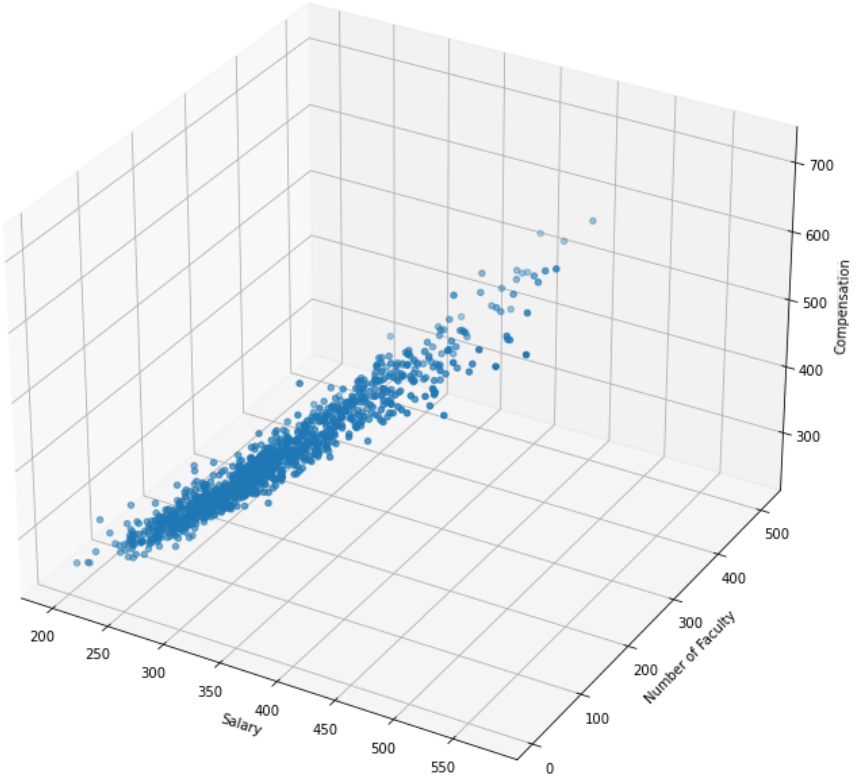
When plotting the different histograms, shown below:

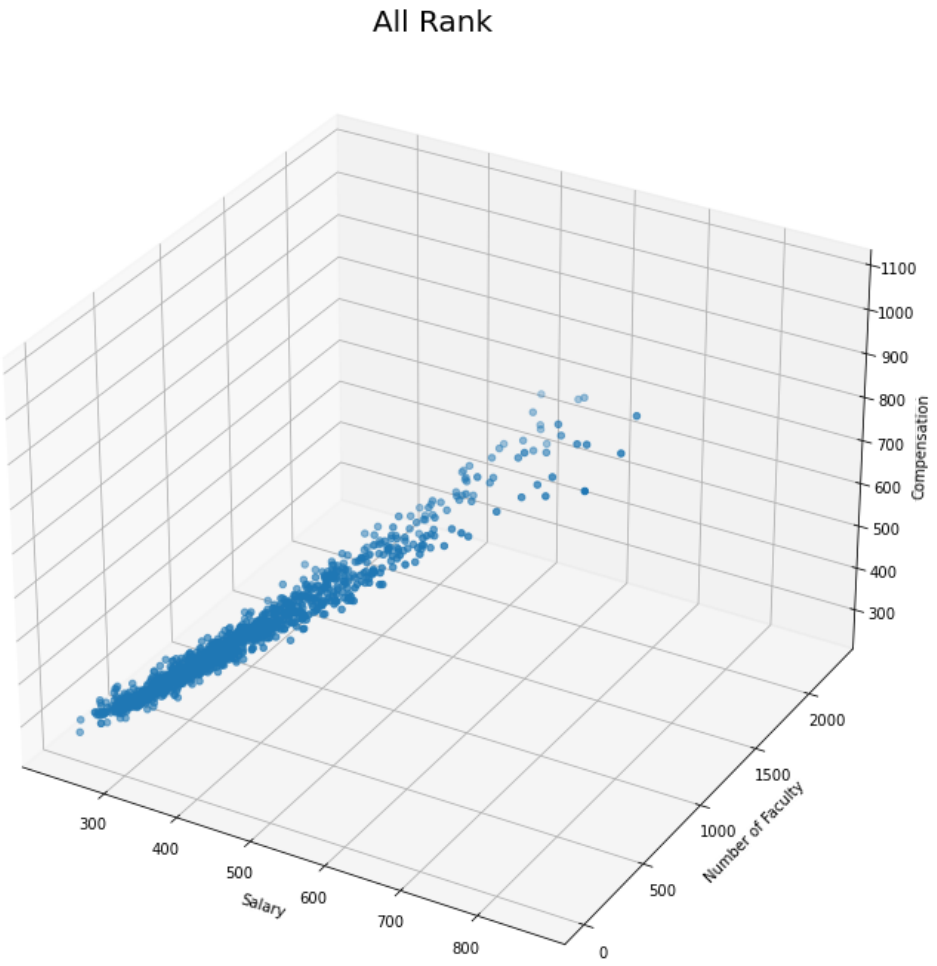


Associate Professor



Assistant Professor





We can plot the points on a 3-dimensional scatter plot with (x,y,z) as (salary, comp, size) to exemplify that there exists outliers in the dataset. With this being said, we can compute the z-score for each respective column and remove all the rows where the z-scores are above 3, effectively removing all the possible outliers.

```
# Computes Z-score of each respective column, and removes rows with Z-scores above 3. 1058 data entries now.
df3 = df2[(np.abs(stats.zscore(df2['Full Professor Comp'])) < 3)]
df3 = df3[(np.abs(stats.zscore(df3['Associate Professor Comp'])) < 3)]
df3 = df3[(np.abs(stats.zscore(df3['Assistant Professor Comp'])) < 3)]
df3 = df3[(np.abs(stats.zscore(df3['All Rank Comp'])) < 3)]
print("{:.0f} data entries (removed outliers)".format(len(df3.index)))
```

```
1058 data entries (removed outliers)
```

After cleaning, making linear assumptions, removing noise, and dealing with the multicollinearity, our dataset is now ready for linear regression.

Results/Experimental Analysis

Decision Tree Classification

File Output

How to Interpret

Since the resulting output is printed out in string format, some explanation on how to interpret the results may be required. As previously mentioned, the decision tree is structures with 2 types of nodes: internal nodes and leaf nodes.

Internal nodes - those representing a splitting condition, are represented as follows:

```
right:D:5 Number of Fulltime Undergrad >= 1673.0 Gain: 0.023262430087638064
```

Where the naming “right” represents which child the node is (right or left). Right children will satisfy the comparison, while left children do not (<).

D:x represents the depth of the node (5 in this case). Where the root node has depth 0.

“Number of Fulltime Undergrad” is a column feature name. This is the feature in question. This feature is compared to (>=) to 1673.0 in this case. Where 1673.0 is the threshold chosen by the split determining method.

Additionally, the corresponding information gain the split is listed at the end.

Leaf nodes - those representing a classification label, are represented as follows:

```
left:D:11 False
```

Again, “left” means left child.
D:x is the depth.

The resulting label is either “False” or “True”, referring to the classification at hand.

Output

The full output can be found in the file sample_output/decision_tree.txt. Below is a brief sample:

```
D:0 Out-State Tuition >= 8644.0 Gain: 0.06167821698331974
  left:D:1 Number of Applications Received >= 5548.0 Gain: 0.026716521440515673
    left:D:2 Number of Parttime Undergrad >= 1256.0 Gain: 0.040723593964334726
      left:D:3 Estimated Personal Spending >= 2500.0 Gain: 0.031648351648351614
        left:D:4 In-State Tuition >= 1044.0 Gain: 0.02311498972165895
          left:D:5 Estimated Book Costs >= 350.0 Gain: 0.21875
            left:D:6 True
            right:D:6 False
          right:D:5 Number of Fulltime Undergrad >= 1673.0 Gain: 0.023262430087638064
            left:D:6 In-State Tuition >= 5150.0 Gain: 0.05743100578781325
              left:D:7 Estimated Book Costs >= 675.0 Gain: 0.12244897959183687
                left:D:8 In-State Tuition >= 3020.0 Gain: 0.0738461538461537
                  left:D:9 Room and Board Costs >= 3264.0 Gain: 0.21568047337278107
                    left:D:10 Number of Parttime Undergrad >= 117.0 Gain: 0.21875
                      left:D:11 True
                      right:D:11 False
                    right:D:10 Estimated Book Costs >= 500.0 Gain: 0.31999999999999984
                      left:D:11 False
                      right:D:11 True
                right:D:9 False
              right:D:8 True
```

Analysis

As seen from the above information, the first split that the algorithm chooses has to do with Out of state tuition. From this observation, if we had to choose one feature that best separates the data, it would be Out-State tuition. With that being said, the information gain is only approximately 0.06, which still allows for a significant level of impurity. Additionally, there are numerous other additions to the lower-depth area of the tree in which the splitting feature has to do with student finances. While we cannot point to one specific feature which may be a cause for graduation rates, we can look at groupings of features. Student finances could play several roles in determining the outcome of graduation rates at a given school.

One outlook that could be made is that the stress of managing higher amounts of expenses may cause students to perform poorly. Expanding further, some students may even choose to drop out if expenses become too difficult to manage, or if they cannot continue to pay for their education.

A secondary outlook on the scenario is actually the inverse of the above statement. There are a few parent-child node pairs in the tree which actually suggest that some higher payments actually lead to higher graduation rates (not as many as suggest otherwise, but still noteworthy). One possible real-world scenario may be that a student is more determined to do better because they know that more money is on the line. Perhaps a student will do better because they have access to higher-quality educational materials (books, etc).

With these observations, it is important to keep in mind that there is no one reason why outcomes are how they are. It is impossible to pick out one specific feature or even set of features that may affect every student's choice or outcomes. The real world is full of factors, and it is currently unfeasible to determine exactly how a student will react when facing any variety of circumstances.

Linear Regression

Output

Upon dividing the aaup dataset and building/training our machine learning regression model, we have the following outputs.

Full Professor

```
[ 1.29825179 -0.02867791]
-23.828804243035506
16.57703863349319
477.2632379161607
21.846355254736675
```

$$y = -23.828804243035506 + 1.29825179X_1 - 0.02867791X_2$$

where

$y = Full\ Professor\ Comp,\ X_1 = Full\ Professor\ Salary,\ X_2 = Number\ of\ Full\ Professors$

$Absolute\ Mean\ Error = 16.57703863349319$

$Mean\ Squared\ Error = 477.2632379161607$

$\sqrt{MSE} = 21.846355254736675$

Associate Professor

```
[ 1.35568907 -0.03292018]
-38.434958446472706
14.766902946740414
347.3189022099251
18.63649382823725
```

$$y = -38.434958446472706 + 1.35568907X_1 - 0.03292018X_2$$

where

$y = Associate\ Professor\ Comp,\ X_1 = Associate\ Professor\ Salary,\ X_2 = Number\ of\ Associate\ Professors$

$Absolute\ Mean\ Error = 14.766902946740414$

$Mean\ Squared\ Error = 347.3189022099251$

$\sqrt{MSE} = 18.63649382823725$

Assistant Professor

```
[ 1.34783096 -0.01453419]
-30.95948053714858
12.609667987741437
263.7953112963445
16.24177672843536
```

$$y = -30.95948053714858 + 1.34783096X_1 - 0.01453419X_2$$

where

$y = Assistant\ Professor\ Comp,\ X_1 = Assistant\ Professor\ Salary,\ X_2 = Number\ of\ Assistant\ Professors$

$Absolute\ Mean\ Error = 12.609667987741437$

$Mean\ Squared\ Error = 263.7953112963445$

$\sqrt{MSE} = 16.24177672843536$

All Rank

```
[ 1.31780312 -0.00999107]
-24.863650668791138
14.232404096131768
325.64548647113395
18.045650070616297
```

$$y = -24.863650668791138 + 1.31780312X_1 - 0.00999107X_2$$

where

$$y = All\ Rank\ Comp,\ X_1 = All\ Rank\ Salary,\ X_2 = Number\ of\ All\ Rank\ Faculty$$

$$Absolute\ Mean\ Error = 14.232404096131768$$

$$Mean\ Squared\ Error = 325.64548647113395$$

$$\sqrt{MSE} = 18.045650070616297$$

Analysis

The four linear regression models above showcase the results from our training data. As we can see, the Absolute Mean Error for each model ranges from between 12 and 17, which indicates that the predicted results from the trained data were approximately off from the initial dataset by $1200 to $1700 average compensation. Although the absolute mean error is relatively small, the mean squared error is very high, ranging between 263 and 480, which indicates that either the data may not be normally distributed or that there exists quite a few larger errors in our predictive data.

Conclusion

Throughout this report, we have taken the two datasets, aaup and usnews, to analyse through two different machine learning techniques. The first one was a decision tree classification method where we determined that students that manage higher amounts of expenses may academically perform lower than students with lower costs. The second method was linear regression, where we predicted the compensations of different ranking faculty members based on their salary and faculty size. We determined that there was correlation between these features, however, the dataset used may not have been distributed normally which resulted in high mean squared error in our predictive data.