

The Relationship Between Student Earnings and Institution Characteristics

Introduction to Databases and Data Mining

Tegveer Ghura Jayesh Jain Rahul Thairani

Introduction

Our final project report employs numeric estimation data mining techniques, namely the *M5P* regression tree, simple and multiple linear regression, to build models that estimate the mean earnings of working students 10 years after entry into an institution based on the institution's characteristics. The baseline model of simple linear regression bases its predictions on the input attribute which explains the greatest variation in mean earnings of working students 10 years after entry into an institution, which happens to be average faculty salary. The correlation coefficient, R , for the single regressor linear model through cross-validation was found to be 0.78. In addition, the *M5P* regression tree algorithm generated a regression tree with 18 leaf nodes, each leaf node containing the average output value for the training examples in that subgroup. The correlation coefficient, R , for the *M5P* regression tree through cross-validation was found to be 0.80.

Finally, the multiple regressor linear model was formulated through the linear regression algorithm with *noAttributeSelection* set to False, resulting in the linear model being forced to base its predictions on all input attributes. The correlation coefficient, R , of the multiple regressor linear model through cross-validation was found to be 0.83. Therefore, the addition of input attributes resulted in predictive models which progressively generalized better, culminating in the multiple regressor linear model with all input attributes explaining approximately 69 percent of the variation in mean earnings of students working and not enrolled 10 years after entry into an institution.

Dataset description

The dataset used is Post-School Earnings Data, which comprises 2018 cross-sectional data on every degree-granting institution in the United States and is trimmed to include only certain attributes of interest and only those institutions containing no missing values for these attributes. The dataset is obtained from the College Scorecard, a database of institution characteristics.¹ The attributes include:

- *median_hh_inc* : Median household income (numeric input)
- *avgfacsal* : Average faculty salary (numeric input)
- *ugds_black* : Total share of enrollment of undergraduate-degree seeking students who are black (numeric input)
- *ugds_white* : Total share of enrollment of undergraduate-degree seeking students who are white (numeric input)
- *sat_avg* : Average SAT equivalent score of students admitted (numeric input)
- *instnm* : Institution name (nominal)
- *unitid* : Unit ID of the institution (nominal)
- *mn_earn_wne_p10* : Mean earnings of students working and not enrolled 10 years after entry into the institution (numeric output)

The SQL database constructed consists of a single relational table entitled ‘University’, which consists of the attributes stated above as columns and institutions with no missing values for these attributes as rows. The primary key for this relational table is *unitid*, the Unit ID of the Institution and the attributes are of the following data types.

- *median_hh_inc* : REAL
- *avgfacsal* : REAL
- *ugds_black* : REAL
- *ugds_white* : REAL
- *sat_avg* : REAL
- *mn_earn_wne_p10* : REAL
- *instnm* : VARCHAR
- *unitid* : VARCHAR

¹ <https://collegescorecard.ed.gov/data/>

Data Preparation

The following steps were taken to pre-process the dataset for data mining.

- The original Post-School Earnings dataset of the College Scorecard was trimmed to include only 8 attributes of interest for use in the Numeric Estimation model.
- The resulting dataset was trimmed using a Python program to only include institutions with no missing values for these attributes of interest, resulting in a dataset with 1256 institutions.
- The resulting dataset was trimmed to remove all institutions with problematic characters in their names and institutions with the same name, resulting in a dataset with 1223 institutions.
- In order to obtain the desired class attribute, the `.csv` file was imported into WEKA, *Edit* under the *Preprocess* tab was clicked, the desired class attribute was right-clicked and the *Attribute as class* option was chosen. Finally, this modified `.csv` file was saved in `.arff` format.
- In order to import the file into DB Browser for SQLite, the `.arff` file was converted into a `.csv` file by using a Python program obtained from GitHub².
- The dataset was randomly shuffled through the *Unsupervised > Instance.Randomize* filter in WEKA before being split 90/10 for training/testing respectively. The training data consists of 1101 institutions while the remaining 122 institutions were used for testing.

A brief description³ of the Python program used to trim the dataset has been provided below.

- The Python program obtains the file name for the input data file from the user and opens a connection to both the input file and the output file, hard-coded to `'results.csv.'`
- The program obtains a list of all lines in the input data file through `.readlines()` and copies the header to the output file.
- The program splits every line in the input file into its various comma-separated fields through `.split(',')` and determines if the value for any particular field is empty. The program copies all lines with no empty fields into the output file.
- The program returns the number of lines with no missing fields and closes the connection to both input and output files.

² <https://github.com/haloboy777/arfftocsv/blob/master/arffToCsv.py>

³ The complete Python program is provided in Appendix A

A brief description⁴ of the Python program used to create Figure 2 has been provided below.

- The program (*discretize.py*) checks each tuple and classifies it as a black or white college based on student enrollment. It then further classifies tuples into bins for several input attributes on the basis of regression tree decision nodes. In the case of *sat_avg* and *median_hh_inc*, we used equal width binning whereas for *avgfacsal*, we used equal height binning.
- The program classifies the attributes of *median_hh_inc*, *avgfacsal* and *sat_avg* into the following categories:
 1. For *median_hh_inc* the categories are black majority and below \$40,000, black majority and above \$40,000, white majority and below \$40,000, and white majority and above \$40,000.
 2. For *sat_avg* the categories are black majority and below 800, black majority and above 800, white majority and below 800, and white majority and above 800.
 3. For *avgfacsal* the categories are black majority and below 7041.5, black majority and above 7041.5, white majority and below 7041.5, and white majority and above 7041.5.
- The program then creates 3 numerical arrays that allow us to generate a graph in excel.⁵

Data Analysis

The following numeric estimation data-mining techniques were employed for data analysis.

- Simple Linear Regression

Simple Linear Regression produces a regression model that is a linear function of the single explanatory input attribute which explains the greatest variation in the output attribute. In other words, the single regressor linear model learned in simple linear regression bases its predictions of the output attribute of previously unseen instances on one of its input attributes. The general form of a single regressor linear model is $y = w_1x_1 + c$, where y is the output attribute, w_1 is a numeric weight, x_1 is the single explanatory input attribute and c is a numeric constant. Simple Linear Regression finds the parameter values (w_1 and c) that minimize the sum of squared differences between the actual y values and the y values estimated by the regression function. In other words, simple linear regression finds the line that ‘best fits’ the training examples, ordered pairs of (x, y) .

⁴ The complete Python program is provided in Appendix A.

⁵ The three numerical arrays are provided in Appendix A.

In order to perform simple linear regression in WEKA, we choose *SimpleLinearRegression* under *Classifiers > Functions*, with the output attribute as mean earnings of students working and not enrolled 10 years after entry into the institution (*mn_earn_wne_p10*). The single regressor linear model output corresponds to the input attribute of average faculty salary.

- Regression Tree

A regression tree refers to a decision tree with leaf nodes containing the average value of the output attribute for the training examples in that subgroup. In other words, a regression tree builds a regression model in the form of a tree structure, with input attribute ranges corresponding to branches of the tree.⁶ During its formation, the regression tree breaks a dataset down into smaller subgroups, while at the same time incrementally developing an associated decision tree.⁷ The final regression tree consists of decision nodes and leaf nodes.⁸ A decision node comprises of two or more branches, each corresponding to a particular range of values for an input attribute whereas a leaf node represents a numerical decision on the output attribute for the training examples in that subgroup.⁹

In order to build a regression tree in WEKA, we choose *M5P* under *Classifiers > Trees*, with the output attribute as mean earnings of students working and not enrolled 10 years after entry into the institution (*mn_earn_wne_p10*). In addition, to make sure that WEKA builds a regression tree with leaf nodes corresponding to output attribute values rather than a model tree with leaf nodes corresponding to regression equations, we change the *buildRegressionTree* parameter from *False* to *True*.

- Multiple Linear Regression

Multiple Linear Regression produces a regression model that is a linear function (i.e. a weighted sum) of several input attributes. In other words, the multiple regressor linear model learned in multiple linear regression bases its predictions of the output attribute of previously unseen instances on several of its explanatory input attributes. The general form of a multiple regressor linear model is:

$y = w_1x_1 + w_2x_2 + \dots + w_nx_n + c$, where y is the output attribute, x_1, \dots, x_n are the input attributes, w_1, \dots, w_n are numeric weights and c is an additional numeric constant. Multiple Linear Regression finds the parameters w_1, \dots, w_n and c that minimize the sum of squared differences between the actual y values

⁶ Sayad, Saed. *Decision Tree Regression*, https://www.saedsayad.com/decision_tree_reg.htm.

⁷ Sayad, Saed. *Decision Tree Regression*, https://www.saedsayad.com/decision_tree_reg.htm.

⁸ Sayad, Saed. *Decision Tree Regression*, https://www.saedsayad.com/decision_tree_reg.htm.

⁹ Sayad, Saed. *Decision Tree Regression*, https://www.saedsayad.com/decision_tree_reg.htm.

and the y values estimated by the regression function. In other words, multiple linear regression finds the line that ‘best fits’ the training examples, ordered pairs of $(x_1, x_2, \dots, x_n, y)$.

In order to perform multiple linear regression in WEKA, we chose *LinearRegression* under *Classifiers > Functions*, with the output attribute as mean earnings of students working and not enrolled 10 years after entry into the institution (*mn_earn_wne_p10*). In addition, to make sure that WEKA uses all numeric attributes while forming the regression equation, we change the *attributeSelectionMethod* parameter from *M5 Method* to *No attribute selection*, which forces the output regression model to be a function of all other numeric attributes.

Additionally, our analysis employs SQL database queries to obtain summary statistics for our dataset. The dataset is queried to find the average, minimum and maximum values for the attributes of median household income (*median_hh_inc*), average faculty salary (*avgfacsal*), average SAT equivalent score of students admitted (*sat_avg*), mean earnings of students working and not enrolled 10 years after entry into the institution (*mn_earn_wne_p10*), total share of enrollment of undergraduate degree-seeking students who are black (*ugds_black*) and total share of enrollment of undergraduate degree-seeking students who are white (*ugds_white*).

- SQL query for finding the average value of *mn_earn_wne_p10*:
SELECT AVG(mn_earn_wne_p10)
FROM University;
- SQL query for finding the minimum value of *mn_earn_wne_p10* and the corresponding institution:
SELECT instnm, mn_earn_wne_p10
FROM University
WHERE mn_earn_wne_p10 = (SELECT min(mn_earn_wne_p10)
FROM University);
- SQL query for finding the maximum value of *mn_earn_wne_p10* and the corresponding institution:

```

SELECT instnm, mn_earn_wne_p10
FROM University
WHERE mn_earn_wne_p10 = (SELECT max(mn_earn_wne_p10)
                           FROM University);

```

In order to obtain summary statistics for the remaining attributes, we simply replace *mn_earn_wne_p10* with the name of the corresponding attribute in the SQL query.

Results

The summary statistics generated through the SQL queries stated above have been presented below.

<i>attribute of interest</i>	<i>minimum</i>	<i>average</i>	<i>maximum</i>
median household income (<i>median_hh_inc</i>)	31,403.34 Alice Lloyd College	63,575.39	95,275.15 George Mason University
average SAT equivalent score of students admitted (<i>sat_avg</i>)	730 Livingstone College	1061.10	1555 California Institute of Technology
average faculty salary (<i>avgfacsal</i>)	3,052 Appalachian Bible College	7949.70	22,146 Stanford University
mean earnings of students working and not enrolled 10 years after entry (<i>mn_earn_wne_p10</i>)	26,100 Livingstone College	50,892.15	153,600 Massachusetts Institute of Technology
total share of enrollment of undergraduate-degree seeking students who are white (<i>ugds_white</i>)	0.0 LeMoyne-Owen College	0.5975	0.9849 Hebrew Theological College
total share of enrollment of undergraduate-degree seeking students who are black (<i>ugds_black</i>)	0.0 Hebrew Theological College	0.1364	0.9875 LeMoyne-Owen College

Numeric Estimation

The numeric estimation data mining techniques produced the following results.

- Simple Linear Regression

The single regressor linear model with 10 fold cross-validation is stated below.

$$y = 11,712.1 + 4.92 x_1$$

where y corresponds to the output attribute of mean earnings of students working and not enrolled 10 years after entry into the institution ($mn_earn_wne_p10$) and x_1 corresponds to the single explanatory input attribute of average faculty salary ($avgfacsal$). The positive coefficient 4.92 on the variable x_1 implies that a unit increase in the average faculty salary ($avgfacsal$) results in an increase of 4.92 in the mean earnings of students working and not enrolled 10 years after entry into the institution ($mn_earn_wne_p10$) on average. The intercept 11,712.1 implies that the mean earnings of students working and not enrolled 10 years after entry into an institution ($mn_earn_wne_p10$) is 11,712.1 on average when the average faculty salary ($avgfacsal$) is 0. Since an average faculty salary ($avgfacsal$) of 0 is not within the scope of the model, the intercept of 11,712.1 has no meaningful interpretation.

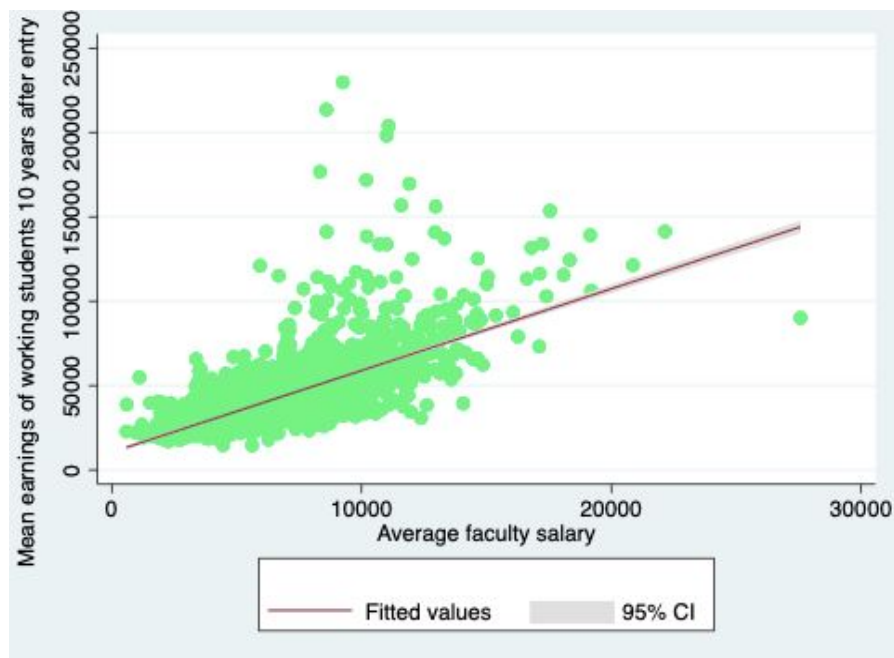


Figure 1: Scatter plot of mean earnings of working students against average faculty salary

The two way scatter plot of the output attribute mean earnings of students working and not enrolled 10 years after entry into the institution (*mn_earn_wne_p10*) against the single explanatory input attribute of average faculty salary (*avgfacsal*), along with the line of best fit, is shown above.

The performance of the single regressor linear model on the training instances¹⁰ is characterised by the correlation coefficient of $R = 0.786$. The performance of the single regressor linear model on the test instances¹¹ is characterised by the correlation coefficient of $R = 0.799$.

Training Results

Correlation coefficient	0.786
Mean absolute error	6598.4169
Root mean squared error	9254.771
Relative absolute error	64.878%
Root relative squared error	61.8178%
Total Number of Instances	1101

Testing Results

Correlation coefficient	0.799
Mean absolute error	7269.9627
Root mean squared error	9850.9142
Relative absolute error	65.1214%
Root relative squared error	60.2302%
Total Number of Instances	122

¹⁰ The complete performance of the single regressor linear model on the training instances is provided in Figure 3 of Appendix B

¹¹ The complete performance of the single regressor linear model on the test instances is provided in Figure 4 of Appendix B

The fact that the correlation coefficients depict the performance of the single regressor linear model on the test instances to be higher than on the training instances, is consistent with better performance on previously unseen examples. Therefore, this implies that the learned single regressor linear model generalizes well. This claim is further reinforced by the correlation coefficient $R = 0.748$ for the cross-validation test¹², which provides a preliminary measure of the ability of the single regressor linear model to generalize on previously unseen instances. The single regressor linear model does make intuitive sense given that the maximum values of average faculty salary (*avgfacsal*) and mean earnings of students working and not enrolled 10 years after entry into the institution (*mn_earn_wne_p10*) are 22,146 and 153,600 respectively. Such measures are consistent with a single regressor linear model having an intercept of 11,712.1 and an increase of 4.92 on average in the mean earnings of students working and not enrolled 10 years after entry into an institution (*mn_earn_wne_p10*) for a unit increase in the average faculty salary (*avgfacsal*).

- Regression Tree

The regression tree generated consists of all input attributes except for total share of enrollment of undergraduate degree-seeking students who are white (*ugds_white*) and results in a decision tree with 18 leaf nodes, with each such node containing the average value of the output attribute for the training examples in that subgroup.¹³ For instance, the leaf node corresponding to the path *sat_avg* > 1149.5, *sat_avg* ≤ 1337.5 and *avgfacsal* ≤ 8413 contains the value 56,760.42. This suggests that the average of the output attribute, mean earnings of students working and not enrolled 10 years after entry into an institution (*mn_earn_wne_p10*), for the training examples in the subgroup defined by the above mentioned path is 56,760.42.

The performance of the regression tree on the training instances¹⁴ is characterised by the correlation coefficient of $R = 0.827$. The performance of the regression tree on the test instances¹⁵ is characterised by the correlation coefficient of $R = 0.828$.

¹² The cross-validation summary of the single regressor linear model is provided in Figure 2 of Appendix B

¹³ The visual and textual representation of the entire regression tree is provided in Appendix D

¹⁴ The complete performance of the regression tree on the training instances is provided in Figure 3 of Appendix C

¹⁵ The complete performance of the regression tree on the test instances is provided in Figure 4 of Appendix C

Training Results

Correlation coefficient	0.8276
Mean absolute error	5793.6897
Root mean squared error	8595.8545
Relative absolute error	56.9656 %
Root relative squared error	57.4165 %
Total Number of Instances	1101

Testing Results

Correlation coefficient	0.8285
Mean absolute error	6429.8181
Root mean squared error	9557.1374
Relative absolute error	57.5957 %
Root relative squared error	58.434 %
Total Number of Instances	122

The fact that the correlation coefficients depict the performance of the *MP5* regression tree model on the test instances to be higher than on the training instances, is consistent with better performance on previously unseen examples and implies that the learned *MP5* regression tree model generalizes well. This claim is further substantiated by the correlation coefficient $R = 0.800$ for the cross-validation test¹⁶, which provides a preliminary measure of the ability of the regression tree to generalize on previously unseen instances. The regression tree makes intuitive sense given that it bases its top two decision nodes on the input attributes which explain the greatest amount of variation in the output attribute and therefore, provide the greatest differentiation among resulting decision nodes, namely the average SAT equivalent score of students admitted (*sat_avg*) and average faculty salary (*avgfacsal*). In addition, decision nodes at lower levels of the decision tree are based on the input attribute total share of enrollment of undergraduate degree-seeking students who are black (*ugds_black*), an input attribute which explains little variation in

¹⁶ The cross-validation summary of the regression tree is provided in Figure 2 of Appendix C

the output attribute, implying that such decisions provide only slight differentiation among resulting leaf nodes.

- Multiple Linear Regression

The multiple regressor linear model with 10 fold cross-validation is stated below.

$$y = -20,789.19 + 0.19 x_1 + 2.63 x_2 + 39.67 x_3 - 5,523.93 x_4 - 3,954.82 x_5$$

where y corresponds to the output attribute of mean earnings of students working and not enrolled 10 years after entry into the institution ($mn_earn_wne_p10$), x_1 corresponds to the input attribute of median household income ($median_hh_inc$), x_2 corresponds to the input attribute of average faculty salary ($avgfacsal$), x_3 corresponds to the input attribute of average SAT equivalent score of students admitted (sat_avg), x_4 corresponds to the input attribute of total share of enrollment of undergraduate-degree seeking students who are black ($ugds_black$) and x_5 corresponds to the input attribute of total share of enrollment of undergraduate-degree seeking students who are white ($ugds_white$).

The coefficient 0.1956 on the variable x_1 implies that for a unit increase in the median household income ($median_hh_inc$), the mean earnings of students working and not enrolled 10 years after entry into the institution ($mn_earn_wne_p10$) increases by 0.1956 on average, holding constant all other input attributes. The coefficient 2.635 on the variable x_2 implies that for a unit increase in the average faculty salary ($avgfacsal$), the mean earnings of students working and not enrolled 10 years after entry into the institution ($mn_earn_wne_p10$) increases by 2.635 on average, holding constant all other input attributes. The coefficient 39.67 on the variable x_3 implies that for a unit increase in the average SAT equivalent score of students admitted (sat_avg), the mean earnings of students working and not enrolled 10 years after entry into the institution ($mn_earn_wne_p10$) increases by 39.67 on average, holding constant all other input attributes. The coefficient -5,523.93 on the variable x_4 implies that for a unit increase in the total share of enrollment of undergraduate-degree seeking students who are black ($ugds_black$), the mean earnings of students working and not enrolled 10 years after entry into the institution ($mn_earn_wne_p10$) decreases by 5,523.93 on average, holding constant all other input attributes. The coefficient -3,954.82 on the variable x_5 implies that for a unit increase in the total share of enrollment of undergraduate-degree seeking students who are white ($ugds_white$), the mean earnings of students working and not enrolled 10 years after entry into the institution ($mn_earn_wne_p10$) decreases by 3,954.82 on average, holding

constant all other input attributes. The intercept -20,789.19 implies that the mean earnings of students working and not enrolled 10 years after entry into an institution (*mn_earn_wne_p10*) is -20,789.19 on average when all the input attributes of the multiple regressor linear model have a value of 0. Since a negative value for the output attribute of mean earnings of students working and not enrolled 10 years after entry into the institution (*mn_earn_wne_p10*) is not feasible and all input attributes having a value of 0 is not within the scope of the model, the intercept of -20,789.19 has no meaningful interpretation.

The performance of the multiple regressor linear model on the training instances¹⁷ is characterised by the correlation coefficient of $R = 0.841$. The performance of the multiple regressor linear model on the test instances¹⁸ is characterised by the correlation coefficient of $R = 0.849$.

Training Results

Correlation coefficient	0.8410
Mean absolute error	5536.1453
Root mean squared error	8099.5095
Relative absolute error	54.4333 %
Root relative squared error	54.1011 %
Total Number of Instances	1101

Testing Results

Correlation coefficient	0.8496
Mean absolute error	6218.2609
Root mean squared error	8696.0992
Relative absolute error	55.7006 %
Root relative squared error	53.1695 %

¹⁷ The complete performance of the multiple regressor linear model on the training instances is provided in Figure 2 of Appendix D

¹⁸ The complete performance of the multiple regressor linear model on the test instances is provided in Figure 3 of Appendix D

Total Number of Instances	122
---------------------------	-----

The fact that the correlation coefficients, R , depict the performance of the multiple regressor linear model on the test instances to be higher than that on the training instances, is consistent with better performance on previously unseen examples and implies that the learned multiple regressor linear model generalizes well. This claim is further substantiated by the correlation coefficient $R = 0.83$ for the cross-validation test, which provides a preliminary measure of the ability of the multiple regressor linear model to generalize on previously unseen instances. The multiple regressor linear model does make intuitive sense since the input attributes of median household income (*media_hh_inc*), average faculty salary (*avgfacsal*) and average SAT equivalent score of students admitted (*sat_avg*) have positive coefficients implying that an increase in their values results in an increase in the value of the output attribute mean earnings of students working and not enrolled 10 years after entry into an institution (*mn_earn_wne_p10*) whereas the input attributes of total share of enrollment of undergraduate-degree seeking students who are black (*ugds_black*) and total share of enrollment of undergraduate-degree seeking students who are white (*ugds_white*) have negative coefficients implying that an increase in their values results in a decrease in the value of the output attribute mean earnings of students working and not enrolled 10 years after entry into an institution (*mn_earn_wne_p10*).

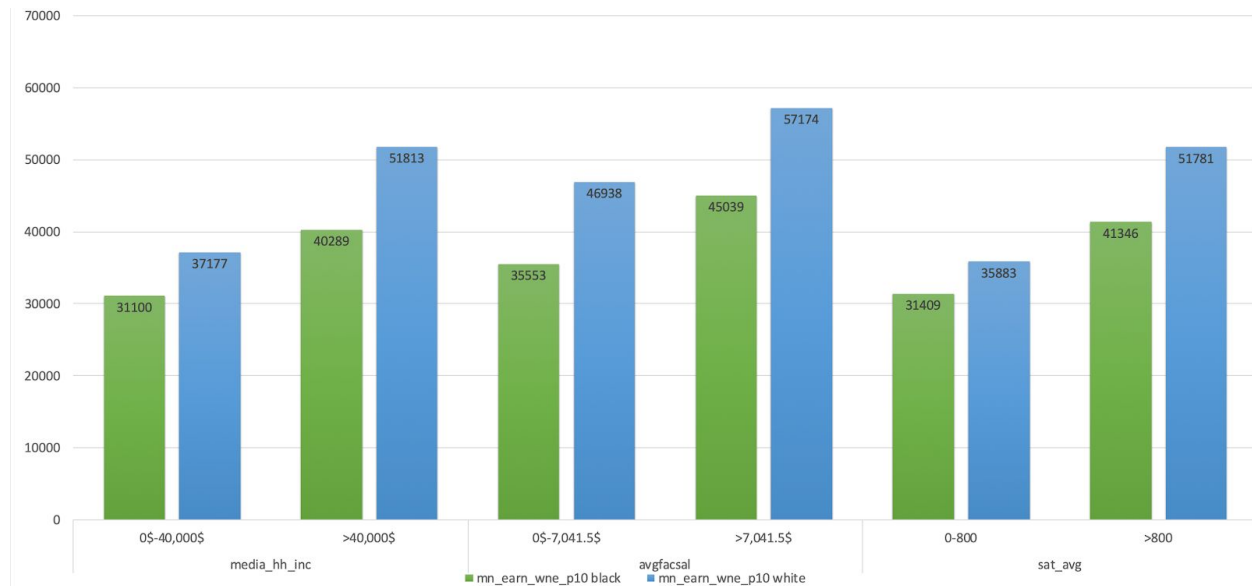


Figure 2: Input attribute averages for institutions with primarily black and white student enrollment

The above figure shows how the output attribute of mean earnings of students working and not enrolled 10 years after entry into an institution (*mn_earn_wne_p10*) compares within subcategories of institutions comprising of enrollment with primarily black or white students for the attributes of median household income (*median_hh_inc*), average faculty salary (*avgfacsal*) and average SAT equivalent score of students admitted (*sat_avg*). The averages with respect to these variables were computed using a Python program provided in the Appendix¹⁹. As can be seen from the above graph, each input attribute is further subdivided into bins based on ranges obtained from either the regression tree or equal width discretization depending on which provided the greatest differentiation for that attribute. For instance, we consulted the regression tree to determine the bins for *avgfacsal*. We decided to discretize *median_hh_inc* in 2 equal width bins and for *sat_avg*, we initially determined bins from the regression tree. However, the results showed great inequality and, consequently, we discretized *sat_avg* into equal width bins.

¹⁹The complete Python program is provided in Appendix A

Conclusion

The primary objective of this report was to understand the impact that several characteristics of an institution, such as the median household income (*median_hh_inc*), average faculty salary (*avgfacsal*), average SAT equivalent score of students admitted (*sat_avg*), total share of enrollment of undergraduate degree-seeking students who are white (*ugds_white*) and total share of enrollment of undergraduate degree-seeking students who are black (*ugds_black*) have on the mean earnings its students who are working and not enrolled 10 years after entry into the institution (*mn_earn_wne_p10*). We began by obtaining data through the College Scorecard, a public database of institution characteristics. The initial dataset had several irrelevant attributes and instances which were removed through the application of a Python program during pre-processing. After splitting our dataset for testing purposes, we proceeded with the application of numeric estimation data mining techniques, resulting in the formation of predictive models and their predictive measures, signifying their ability to generalise.




	Simple Linear Regression				MP5 Algorithm				Multiple Linear Regression			
Correlation Coeff	0.76	0.77	0.78	0.79	0.80	0.81	0.82	0.83	0.84	0.85	0.86	0.87
Training Data	Difference = 0.013 or 1.3% 				Difference = 0.0009 or 0.09% 				Difference = 0.0086 or 0.86% 			
Testing Data												

Figure 3: Correlation coefficients of training and testing data for numeric estimation techniques

The numeric estimation data mining algorithms of the regression tree, simple and multiple linear regression were applied to the training data set, with the correlation coefficients of the resulting models summarised in the figure shown above. The correlation coefficient is a numerical measure of the statistical relationship between the actual and predicted values of the output attribute of mean earnings of students working and not enrolled 10 years after entry into an institution (*mn_earn_wne_p10*). As can be

seen from the figure above, the regression tree has a 0.09% difference between its correlation coefficients, the single regressor linear model has a 1.3% difference between its correlation coefficients and the multiple regressor linear model has a 0.86% difference between its correlation coefficients. To conclude, the model produced through multiple linear regression has the best ability to generalise on previously unseen instances given its correlation coefficient through cross-validation of $R=0.83$.

Appendix A

preprocess_data.py

```
import random

file = input('Enter the name of the file: ')      # input file
name
file2 = 'results.csv'      # results file hard-coded to
'results.csv'

infile = open(file, 'r')      # open connection to input
file
outfile = open(file2, 'w')      # open connection to output
file

lines = infile.readlines()      # read lines into a list from input
file
print(lines[0], file = outfile)      # print header to output
file
lines = lines[1:]      # skip
header
counter = 0      # initialize counter for number of
institutions

for k in range(len(lines)):      # for all lines in input
file
    components = lines[k].split(',')      # split line into
attributes
    count1 = 0      # initialize counter for missing
values
    for i in range(len(components)):      # for every
institution
        if len(components[i]) == 0:      # if there is a missing
value...
            count1 += 1      # increase missing value counter by
1
        if count1 == 0:      # if there are no missing
values
            counter += 1      # increment institution counter by
1
        x = lines[k]      # assign institution line to
variable
```

```

        print(x, file = outfile)          # print variable to output
file

print('There are', counter, 'institutions with no missing values')

infile.close()                          # close connection to input
file
outfile.close()                         # close connection to output
file

```

discretize_data.py

```

file = input('Enter the name of the file: ')      # input file
name
file2 = 'results.csv'                            # name of the output
file

infile = open(file, 'r')                        # opening the input
file
outfile = open(file2, 'w')                      # opening the output
file

lines = infile.readlines()                      # reading the lines in the input
file
lines = lines[1:]                              # removing the
header
k = 1      # setting the counter to 1, will be used to end the
loop
data_hh = [0,0,0,0]                            # keeps a count for all the
categories
mean_hh = [0,0,0,0]                            # stores the values for mean of the
categories

```

```

data_afs = [0,0,0,0]          # keeps a count for all the
categories
mean_afs = [0,0,0,0]         # stores the values for mean of the
categories

data_s = [0,0,0,0]           # keeps a count for all the
categories
mean_s = [0,0,0,0]           # stores the values for mean of the
categories

while k < len(lines):         # while loop to go through the
file
    lines[k] = lines[k][:-1]   # removing '/n' from the
line
    components = lines[k].split(',')
    if components[-2] > components[-3]: # classifying the
institution
        components[-2] = 'black'
    else:
        components[-2] = 'white'

    integer = float(components[2]) # converting median_hh_inc to
float
    if integer <= 40000.0 and components[-2] == 'black':#
discretizing
        data_hh[0] += 1
        mean_hh[0] = mean_hh[0] + float(components[-1])
    elif integer > 40000.0 and components[-2] == 'black':
        data_hh[1] += 1
        mean_hh[1] = mean_hh[1] + float(components[-1])
    elif integer <= 40000.0 and components[-2] == 'white':
        data_hh[2] += 1
        mean_hh[2] = mean_hh[2] + float(components[-1])
    elif integer > 40000.0 and components[-2] == 'white':
        data_hh[3] += 1
        mean_hh[3] = mean_hh[3] + float(components[-1])

integer1 = float(components[3]) # converting avgfacsal to float

```

```

        if integer1 <= 7041.5 and components[-2] == 'black':#
discretizing
            data_afs[0] += 1
            mean_afs[0] = mean_afs[0] + float(components[-1])
        elif integer1 > 7041.5 and components[-2] == 'black':
            data_afs[1] += 1
            mean_afs[1] = mean_afs[1] + float(components[-1])
        elif integer1 <= 7041.5 and components[-2] == 'white':
            data_afs[2] += 1
            mean_afs[2] = mean_afs[2] + float(components[-1])
        elif integer1 > 7041.5 and components[-2] == 'white':
            data_afs[3] += 1
            mean_afs[3] = mean_afs[3] + float(components[-1])

integer2 = float(components[4]) # converting sat_avg to float
if integer2 <= 800 and components[-2] == 'black': # discretizing
    data_s[0] += 1
    mean_s[0] = mean_s[0] + float(components[-1])
elif integer2 > 800 and components[-2] == 'black':
    data_s[1] += 1
    mean_s[1] = mean_s[1] + float(components[-1])
elif integer2 <= 800 and components[-2] == 'white':
    data_s[2] += 1
    mean_s[2] = mean_s[2] + float(components[-1])
elif integer2 > 800 and components[-2] == 'white':
    data_s[3] += 1
    mean_s[3] = mean_s[3] + float(components[-1])

k = k + 1

for x in range(len(mean_hh)):          # loop to calculate the mean
data
    if mean_hh[x] != 0 and data_hh[x] != 0:
        mean_hh[x] = mean_hh[x] // data_hh[x]
    Else:                                # ignore any 0
values
        mean_hh[x] = mean_hh[x]

for y in range(len(mean_afs)):
    if mean_afs[y] != 0 and data_afs[y] != 0:

```

```

        mean_afs[y] = mean_afs[y] // data_afs[y]
    else:
        mean_afs[y] = mean_afs[y]

for z in range(len(mean_s)):
    if mean_s[z] != 0 and data_s[z] != 0:
        mean_s[z] = mean_s[z] // data_s[z]
    else:
        mean_s[z] = mean_s[z]

print('mean_hh = ', mean_hh)      # printing the array for
median_hh_inc
print('mean_afs = ', mean_afs)    # printing the array for avgfacsal
print('mean_ss = ', mean_s)       # printing the array for sat_avg

infile.close()                    # closing the input file

outfile.close()                   # closing the output file

```

The arrays created through the program resulted in the following table:

		<i>media_hh_inc</i>		<i>avgfacsal</i>		<i>sat_avg</i>	
		<40,000\$	>40,000\$	<7,041.5\$	>7,041.5\$	<800	>800
<i>mn_earn_wne_p10</i>	Black majority	31,100	40,289	35,553	45,039	31,409	41,346
	White majority	37,177	51,813	46,938	57,174	35,883	51,781

Appendix B

Simple Linear Regression

```
Attributes: 6
            median_hh_inc
            avgfacsal
            sat_avg
            ugds_white
            ugds_black
            mn_earn_wne_p10
Test mode:  10-fold cross-validation

=== Classifier model (full training set) ===

Linear regression on avgfacsal

4.92 * avgfacsal + 11712.1

Predicting 0 if attribute value is missing.
```

Figure 1: Single Regressor Linear Model

```
=== Cross-validation ===
=== Summary ===

Correlation coefficient          0.7848
Mean absolute error             6613.8212
Root mean squared error         9277.7713
Relative absolute error         64.9425 %
Root relative squared error     61.8918 %
Total Number of Instances      1101
```

Figure 2: Cross-Validation Summary of Single Regressor Linear Model

=== Evaluation on training set ===

Time taken to test model on training data: 0.02 seconds

=== Summary ===

Correlation coefficient	0.786
Mean absolute error	6598.4169
Root mean squared error	9254.771
Relative absolute error	64.878 %
Root relative squared error	61.8178 %
Total Number of Instances	1101

Figure 3: Performance of Single Regressor Linear Model on Training Instances

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correlation coefficient	0.7992
Mean absolute error	7269.9627
Root mean squared error	9850.9142
Relative absolute error	65.1214 %
Root relative squared error	60.2302 %
Total Number of Instances	122

Figure 4: Performance of Single Regressor Linear Model on Test Instances

Appendix C

Regression Tree

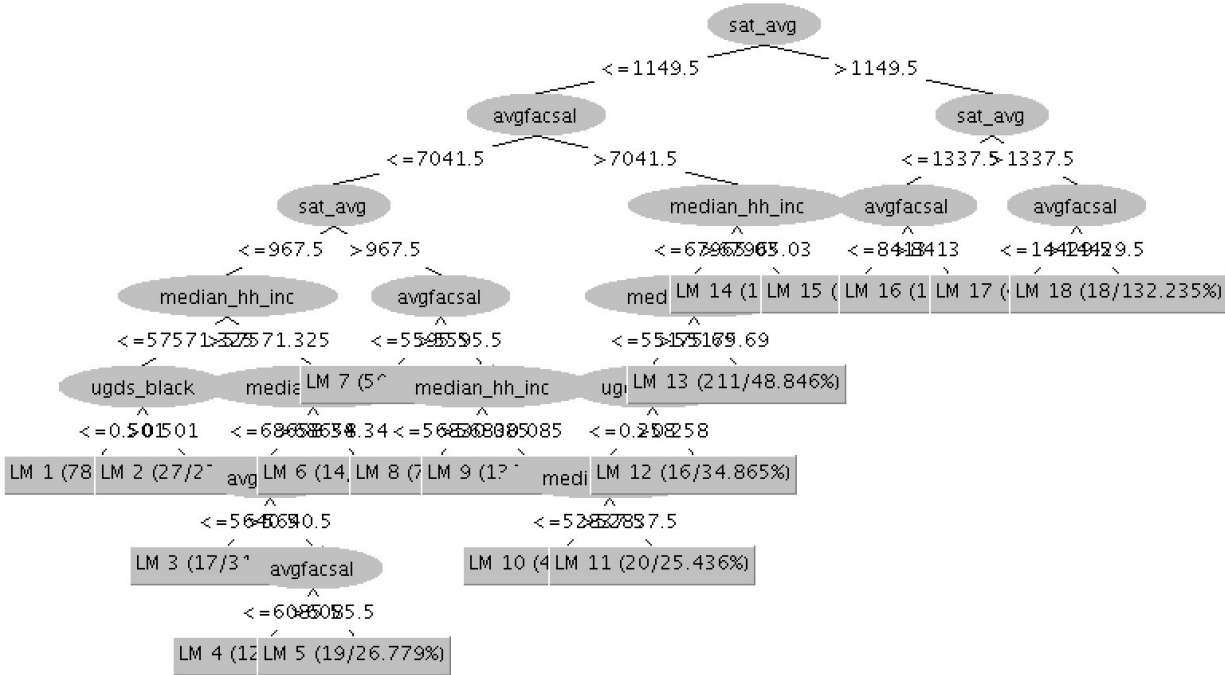


Figure 1: Visual Representation of Regression Tree

Textual Representation of M5 pruned regression tree:
(using smoothed linear models)

```

sat_avg <= 1149.5 :
| avgfacsal <= 7041.5 :
| | sat_avg <= 967.5 :
| | | median_hh_inc <= 57571.325 :
| | | | ugd_black <= 0.501 : LM1 (78/33.299%)
| | | | ugd_black > 0.501 : LM2 (27/23.21%)
| | | median_hh_inc > 57571.325 :
| | | | median_hh_inc <= 68658.34 :
| | | | | avgfacsal <= 5640.5 : LM3 (17/31.973%)
| | | | | avgfacsal > 5640.5 :
| | | | | | avgfacsal <= 6085.5 : LM4 (12/37.843%)
| | | | | | avgfacsal > 6085.5 : LM5 (19/26.779%)
| | | | median_hh_inc > 68658.34 : LM6 (14/34.463%)

```

```

| | sat_avg > 967.5 :
| | | avgfacsal <= 5595.5 : LM7 (59/42.261%)
| | | avgfacsal > 5595.5 :
| | | | median_hh_inc <= 56830.085 : LM8 (70/28.24%)
| | | | median_hh_inc > 56830.085 : LM9 (136/33.044%)
| avgfacsal > 7041.5 :
| | median_hh_inc <= 67965.03 :
| | | median_hh_inc <= 55175.69 :
| | | | ugds_black <= 0.258 :
| | | | | median_hh_inc <= 52837.5 : LM10 (42/28.923%)
| | | | | median_hh_inc > 52837.5 : LM11 (20/25.436%)
| | | | ugds_black > 0.258 : LM12 (16/34.865%)
| | | median_hh_inc > 55175.69 : LM13 (211/48.846%)
| | median_hh_inc > 67965.03 : LM14 (156/62.306%)
sat_avg > 1149.5 :
| sat_avg <= 1337.5 :
| | avgfacsal <= 8413 : LM15 (38/53.878%)
| | avgfacsal > 8413 : LM16 (127/77.549%)
| sat_avg > 1337.5 :
| | avgfacsal <= 14429.5 : LM17 (41/95.677%)
| | avgfacsal > 14429.5 : LM18 (18/132.235%)

```

LM num: 1

```

mn_earn_wne_p10 =
    + 39300.2803

```

LM num: 2

```

mn_earn_wne_p10 =
    + 35989.6777

```

LM num: 3

```

mn_earn_wne_p10 =
    + 40307.3055

```

LM num: 4

```

mn_earn_wne_p10 =
    + 42832.7767

```

LM num: 5

```

mn_earn_wne_p10 =

```

+ 41670.3135

LM num: 6

mn_earn_wne_p10 =
+ 43606.4724

LM num: 7

mn_earn_wne_p10 =
+ 41881.8997

LM num: 8

mn_earn_wne_p10 =
+ 43002.8019

LM num: 9

mn_earn_wne_p10 =
+ 45604.5325

LM num: 10

mn_earn_wne_p10 =
+ 45129.804

LM num: 11

mn_earn_wne_p10 =
+ 46683.0962

LM num: 12

mn_earn_wne_p10 =
+ 42920.5313

LM num: 13

mn_earn_wne_p10 =
+ 49276.4016

LM num: 14

mn_earn_wne_p10 =
+ 53919.2421

LM num: 15

mn_earn_wne_p10 =

```

+ 56760.4264

LM num: 16
mn_earn_wne_p10 =
+ 64279.284

LM num: 17
mn_earn_wne_p10 =
+ 77517.0952

LM num: 18
mn_earn_wne_p10 =
+ 92650.0748

Number of Rules : 18

```

```

=== Cross-validation ===
=== Summary ===

```

Correlation coefficient	0.8003
Mean absolute error	6172.2016
Root mean squared error	9155.6423
Relative absolute error	60.6061 %
Root relative squared error	61.0771 %
Total Number of Instances	1101

Figure 2: Cross-Validation Summary of Regression Tree

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correlation coefficient	0.8276
Mean absolute error	5793.6897
Root mean squared error	8595.8545
Relative absolute error	56.9656 %
Root relative squared error	57.4165 %
Total Number of Instances	1101

Figure 3: Performance of Regression Tree on Training Instances

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correlation coefficient	0.8285
Mean absolute error	6429.8181
Root mean squared error	9557.1374
Relative absolute error	57.5957 %
Root relative squared error	58.434 %
Total Number of Instances	122

Figure 4: Performance of Regression Tree on Test Instances

Appendix D

Multiple Linear Regression

=== Classifier model (full training set) ===

Linear Regression Model

mn_earn_wne_p10 =

0.1956 * median_hh_inc +
2.635 * avgfacsal +
39.6723 * sat_avg +
-5523.9383 * ugds_white +
-3954.8272 * ugds_black +
-20789.1913

Time taken to build model: 0.01 seconds

=== Cross-validation ===

=== Summary ===

Correlation coefficient	0.8385
Mean absolute error	5573.8759
Root mean squared error	8158.164
Relative absolute error	54.731 %
Root relative squared error	54.4229 %
Total Number of Instances	1101

Figure 1: Cross-Validation Summary and Model for Multiple Regressor Linear Model

=== Evaluation on training set ===

Time taken to test model on training data: 0.01 seconds

=== Summary ===

Correlation coefficient	0.841
Mean absolute error	5536.1453
Root mean squared error	8099.5095
Relative absolute error	54.4333 %
Root relative squared error	54.1011 %
Total Number of Instances	1101

Figure 2: Performance of Multiple Regressor Linear Model on Training Instances

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correlation coefficient	0.8496
Mean absolute error	6218.2609
Root mean squared error	8696.0992
Relative absolute error	55.7006 %
Root relative squared error	53.1695 %
Total Number of Instances	122

Figure 3: Performance of Multiple Regressor Linear Model on Test Instances