ML with College Major Salaries Data

Machine Learning Final Report

Jiyeong Oh, department of Data Science
Hanyang Univ.

Introduction

1. Problem addressing

Universities are institutions of higher education and research that confer degrees in various fields of study. It plays the role providing a deep understanding of the major field and at the same time, provides the way to view the world, through various liberal education. Although the historical goal of university stands on 'pure pursuit of learning', many students tend to think the college as a place to prepare for employment, as the belief that 'the degree of expertise a person has affects the wages he or she will receive later on' grows. Accordingly, the definition of 'good university' also has been various, compared to the previous atmosphere that criticized the commercialization of universities.

Amid this wave of change, it is the current state that preferences of different majors are being divided; parents of someone might be worried when their child choose philosophy as his/her major, while be happy when their child choose to work in medical area. However, when the criteria of 'good university' moves from just simple initial salary to the 'increasement' of the wage, there is a possibility that different decision would come out when choosing the college major. Besides, elements outside of university major, are also likely to be the factors that affect a person's wages; the type of university a student attends or the location of the college can also affect the amount of salaries the student would earn after graduation, as the tendency of 'university hierarchy' gets stronger. In that sense, this project aims to acquire some insights associated with salaries of the graduates depending on several elements by building various models which predicts 'the amount of increasement' between initial salaries and mid-career salaries, and comparing the performance of the models.

2. Data Description & Brief Workflow of the Study

Based on the datasets obtained from the Wall Street Journal (which is based on data from Payscale, Inc), model building processes were implemented. According to the Kaggle, which offers the datasets above, three csv files are provided: 'Salaries for Colleges by

Type', 'Salaries for Colleges by Region', and 'Degrees that Pay you Back'. The first dataset involves the type (State, Party, IvyLeague, Engineering, and Liberal Arts) of each university, followed by the various overall-income information such as median value of salaries. Similarly, 'Salaries for Colleges by Region' contains income variables depending on the location of the university(California, Western….etc). 'Degrees that Pay you Back' also shows salaries depending on the 50 different majors.

Building the model, it was difficult to build a prediction model in a sense that the characteristics of the features were almost similar each other, that there is no unified target variables in three datasets. To solve this problem, 6900 'simulated students' were created based on the datasets given instead of using raw datasets, and models were built on this processed simulation field. The goal of the model was set to 'predict the percent change of a simulated individual's salary'.

Exploratory Data Analysis

To understand the data well so that it leads to better direction-setting for building a prediction model, EDA(Exploratory Data Analysis) was implemented in advance. Visualizing the original data with python packages, several insights have come out.

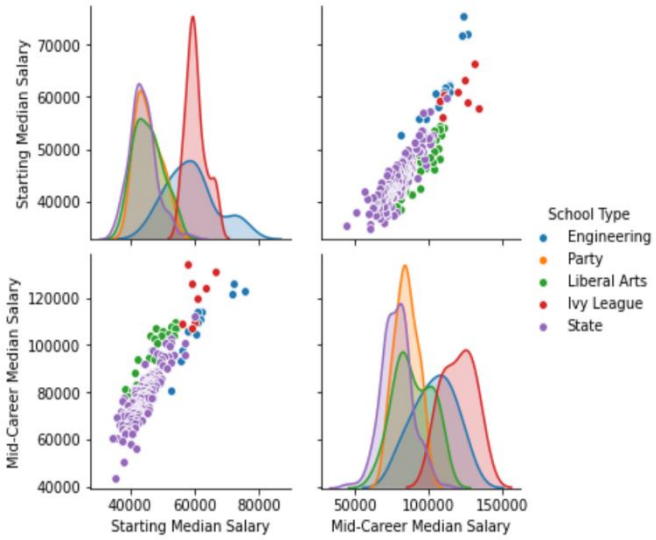

Figure 1

Figure on the left side (Figure 1) is the pair-plot between the initial salary and mid-career salary, depending on the School Type. Plotting the graphs, some significantly different distributions were observed among the type of school. For example, Ivy League tends to have high increasement of salaries, with relatively high income on average, both on starting and mid-career (median). These led to the insight that the type of the university affect the income of an individual, as one of the features of income-increasement predicting model.

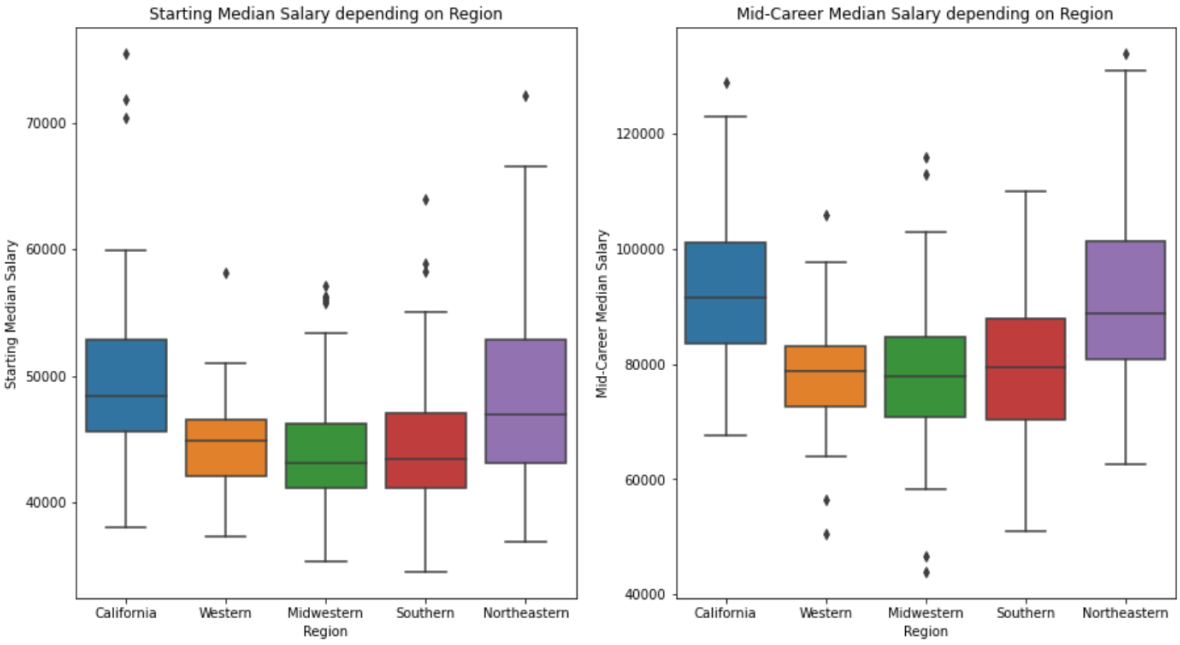

Figure 2

The location where the university is in, also was observed to be significant factor when predicting the future salary increasement. Two sets of bar-plots (Figure 2) represent the Starting Median Salary depending on Region and Mid-Career Salary depending on Region, respectively. From the graph, it is inferred that Universities located in California tends to have higher income on both starting and mid-career, though deviation ranges exist. These plots also indicates that the college region is entitled to be one of the candidate elements that affect the income of an individual.

Figure 3

Figure 3 represents the correlations among the salary information depending on the major. Calculating the correlation between the columns, it was concluded that income elements followed by the major of an individual are highly correlated. However, the results does not guarantee the excellent performance of models, but leave behind some doubts that the variables are too similar that the number of features are not

enough to predict the target, which indeed means, overfitting.

Analysis and Results

1. Preprocessing (Cleaning the Data)

Before running into the modeling process, preprocessing was preceded for easier and accurate analyzing. First, pre-existing dollar signs($) were eliminated. Then the datasets ‘Salaries for Colleges by Type’ and ‘Salaries for Colleges by Region’ were merged, since there are 268 schools which names are same on both datasets. Also, new variable called ‘school_percent_change’ was created by calculating the increasement of median salaries of each school. Based on the merged school dataset and major dataset, ‘pseudo-data’ was created as the next step. 6900 ‘simulated students’ across the 50 different majors who are attending various kind of colleges in different regions, were created by randomizing the data across the different school types and regions. Finally, the target variables, which is the percent change of a simulated student, were set by calculating the average of percent_change depending on major and school. Additionally, slight amount of noises were added to the target values for evaluating the performance of models better.

2. Modeling & Performance analysis

2.1 Decision Tree

As Decision Tree is one of the simplest decision support tools that schematizes decision rules and their results into the tree structure, Decision Tree Regressor was selected as the initial model to predict the increasement of income.

2.1.1 Single Split versus K-fold Cross-Validation

Running the code, the RMSE and Explained Variance scores with slight deviations have come out. As train set and test set are randomly selected for each turn, there are no significant pattern of performance scores among five turns. Calculating the average and the biggest deviation (from the average) scores of the five turns, the average RMSE score was about 7, with range of (+0.18492) and (-0.089397) and the average Explained Variance score was about 5
, with range of (+0.01638) and (-0.026084). (See Table_1.)

Table_1

Turn	Decision Tree RMSE	Decision Tree Expl Var
1	6.908353515261428	0.5407486856310383

2	7.130388438806357	0.5119732871762277
3	6.885368373625837	0.5367407716138609
4	6.9471749061311225	0.5463885803264309
5	6.856074701519786	0.5544331220137195

Comparing the scores with those with the cross-validations (See Table_1 and Table_2 on 2.1.2), the average scores of performances were observed to be better on cross-validation(with lower RMSE and higher Explained Variance), unless the number of the folds of cross-validation was too small (e.g. cv fold=3). Another difference is that the scores with cross-validation tend to be consistent, comparing with those with single train-test split. Looking through the scores with a single test/train split, it was observed that the scores of some turns were higher than those of the cross-validation, and some of them were lower. This involves the probability of a single test/train split to be overfitted to trainset. Although the ranges of performance score are overlapped because of the expected standard deviation, it is enough to suggest that cross-validation is preferred as it tends to be less overfitted than a single test/train split. As the prediction results are sensitively affected by the composition of the train set and test set, it is concluded the cross-validation, which evaluate the performance score in consistent state, is the 'stable' method. Therefore, from the section below, cross-validation is applied as default process.

2.1.2 Finding the Optimal CV Fold

Although it is the fact that cross-validation lowers the probability to lead the model to ovefitting, there is no use if inappropriate number of folds are used when applying cross-validation. In that sense, several decision tree regressor models with different number of cross-folds were built and their performance score as well as cv runtime, were compared.

Table_2

Number of fold	Decision Tree RMSE	Decision Tree Expl Var	CV Runtime
3	7.03 (+/- 0.03)	0.52 (+/- 0.03)	0.12857317924499512
5	6.95 (+/- 0.08)	0.53 (+/- 0.03)	0.25621700286865234
8	6.89 (+/- 0.27)	0.54 (+/- 0.06)	0.4633903503417969
10	6.86 (+/- 0.33)	0.54 (+/- 0.09)	0.5720360279083252

Running the code and comparing the results, it was showed that the model with the larger number of folds tends to have low RMSE (on average), and Explained Variance score seems to get higher as the number of folds grows, converge around the model with 8 folds, and stop increasing (of course on average.) On the other hand, it was showed that the larger the number of folds, the greater the variation in performance score, which was interpreted as it is because of the smaller the size of test set become as the number of folds grows. Runtime tended to take

much longer with larger number of folds. This is interpreted as because the number of rotations of each fold as a test set increases the runtime. Through this, it was inferred that it was not good to have too many fold, but not too few, and concluded that it was important to find the optimal number of fold. Therefore, considering not only the average performance score but also the deviation of them and cv runtime, 5 folds are selected to be optimal in this dataset. From the section below, cv=5 would be applied as default process when doing cross-validation.

2.2 Random Forest

Since the Random Forest, which a number of trees work together based on their different parts of the information, it can be observed that the performance scores with random forest model tends to be better(with higher Explained Variance score and lower RMSE) than those with decision tree model, even considering deviations of the scores. (See Table_3 with 100 trees and compare them with decision tree in 2.1) This indicates the advantage of random forest: supplementing the limits of decision tree (It does not generalize well) so that it performances better with the test set. This is because the random forest model has less possibility to be overfitted as multiple decision trees with slightly different information contained are created and results from those different decision trees are put together.

2.2.1 Finding the Optimal Number of Trees

Just like the number of cross-validation folds, finding the optimal number of trees are also important to build more tractable model. Therefore, several Random Forest Regressor models with different number of trees were built and their performance score as well as cv runtime, were compared.

Table_3

Number of Trees	Random Forest RMSE	Random Forest Expl Var	CV Runtime
5	5.71 (+/- 0.23)	0.69 (+/- 0.02)	0.34584832191467285
10	5.55 (+/- 0.18)	0.70 (+/- 0.01)	0.7255816459655762
20	5.49 (+/- 0.19)	0.71 (+/- 0.02)	1.4053142070770264
50	5.46 (+/- 0.20)	0.71 (+/- 0.02)	3.4534828662872314
100	5.45 (+/- 0.19)	0.71 (+/- 0.02)	6.639969110488892
200	5.44 (+/- 0.19)	0.71 (+/- 0.02)	12.768411874771118
500	5.44 (+/- 0.19)	0.72 (+/- 0.02)	32.907233238220215
1000	5.44 (+/- 0.19)	0.72 (+/- 0.02)	66.02632403373718

Running the code, it was observed that observed that the ranges of the performance scores are overlapped because of the expected deviations, although the average RMSE score seems to get lower as the number of trees grows, converge around the model with 200 trees, and stop increasing (of course on average.) and Explained Variance score also seems to converge (after increasing), around the model with 500 trees. Therefore, it is hasty action to judge that the model with many trees beats the model with small number of trees, based on the evidence we have. Considering the increasing runtime followed by the increasing number of trees, model with a large number of trees is not the optimal model in this case. From the section below, random forest regressor with 100 trees would be applied as default process, considering both the 'simplicity' of a model and performance.

2.2.2 Feature Selection on Random Forest

'How tractable the model is' is also one of the conditions of a good model, as well as the performance score, just like you saw on section above. Here, feature selection was implemented to see whether the model can predict the individual's increasement of income with fewer number of features. With wrapper-based feature selection, which finds the optimal feature subset by building multiple models on different sets of features, 11 features out of 72 are selected; they are 'school_start_50', 'school_mid_50', 'school_mid_10', 'school_mid_25', 'school_mid_75', 'school_mid_90', 'major_start_50', 'major_mid_50', 'major_mid_10', 'major_mid_75', and 'major_mid_90'.

Table_4 (the number of tree=100)

Random Forest RMSE	Random Forest Expl Var	CV Runtime
5.46 (+/- 0.20)	0.71 (+/- 0.02)	4.142171859741211

Running the code and comparing it with those without feature selection, (Table_3 with 100 trees) there was no significant difference, as their range of the scores are overlapped. On the other hand, it is observed that 11 features are selected out of 72, while the model without feature selection is bult with all features. Taking these inferences into accounts, it can be inferred that the model with feature selection offered almost the same performance score with fewer features, comparing with the one with all features. This indicates that choosing more tractable model, taking a risk of slight decrease in performance, is also a reliable. Therefore, when determining which is the 'good' model, not only the performance scores, but also 'how simple the model is' should be considered.

3. Neural Network, Gradient Boosting and Ada Boosting

Another several models were built additionally: Gradient Boosting and Ada Boosting, which are different types of boosting model, a sequential ensemble which improves its

performance itself by learning from its errors. The difference, if it has to be looked for, is that Ada boosting re-weights based on previously misclassified examples while gradient boosting model, which is a branch of Ada boosting, involves minimizing the loss function process as well as re-weighting. Neural Network, on the other hand, teaches layers of connected nodes by showing them their own errors to adjust connection weights.

Table_5

	Gradient Boosting	Ada Boosting	Neural Network
RMSE	5.01 (+/- 0.22)	5.30 (+/- 0.17)	10.19 (+/- 0.23)
Expl Var	0.76 (+/- 0.02)	0.73 (+/- 0.02)	-0.00 (+/- 0.00)
CV Runtime	3.2819995880126953	2.593351125717163	0.5771491527557373

Comparing each performance score of Gradient Boosting model and Ada Boosting model, it was observed that the two performed similarly as the ranges of the performance scores are overlapped because of the expected deviations. This leads to the inference here that it is difficult to say either of them is better than the other, but it is significant enough to say that they are different types of boosting model.

However, quite significant difference with Neural Network was observed as boosting methods outperformed the Neural Network, at least in terms of relative performance score, considering the ranges of expected deviation. However, this does not generalize that the Neural Network is a model that always lags behind the boosting model. As neural network tends to be used for pattern recognition problems including speech and image, there exists the possibility that Neural Network model outperforms the others if appropriate data were given instead if the diabetes data. In other words, Neural Network was not the optimal approach to implement classification on the ‘given’ diabetes dataset, as it does not deal with images or speech recognition. The conclusion is that a model suitable for the situation should be used.

Conclusion

To paraphrase the paper, the overall conclusions of the project would be following:
From EDA, it was observed that several university elements including not only the major of an individual, but also the type of school and location where the school exist, affects the increasement of someone’s income, as well as their initial salaries and mid-career salaries.
Also, several models which predict income increasement of the individuals who graduated different universities with different major, were built with techniques including cross validations and feature selection. Besides, we have come to the conclusion that in the process of determining which model is the best, the performance score is not a single element to be

reflected, but also the tractability and temporal elements should be taken into account in combination. Considering that the question of which criteria to focus on among those elements does not end with one single answer, finding the optimal model is a series of endless trade-off.

From these modeling results, it was concluded that it is possible to predict someone's future income increasement as well as their initial income before he or she graduate the university by machine learning. This gives us the inference that machine learning opens the door to the possibility to help solving the real-world problems; in this case, giving the guideline of choosing the university and major to high school students who is agonizing over choosing the college.

Limits & Future work

Based on the EDA and ML results, several limits of the study have come out. The results also played a role in providing some insights for future study, with different subject associated with university and income. Below is the ideas of future work.

- As the dataset which were created with original datasets is 'pseudo dataset', with less outliers and variance, the model has probability to be overfitted to simulated data. As we know that good models do not only perform well in refined situations, real-dataset which contains a student's university information and his/her wages, needs to be collected for the future study.

- The project did not consider the actual percentage of people per major. As simulated student sample was just simply 'randomly extracted with restoration', the real ratio depending on the major should be taken into account when building another model in the future.

- Finding out whether the major you choose or the status of your school affect the future income tendency would be interesting. As the given data only indicates that two somewhat affects the income but does not give which factor has more influence on the target, figuring out the amount of relation between target feature can be one of the subject of future study.

-Based on the pair-plot and bar-plot in EDA section, significantly different distributions of income were observed, both on the types of school and the location of the school. This led to the idea of building model which classifies the type of the university with other features or predicting the location of school based on other features.