Engineering Graduate Salary Prediction

Hyunjin Chang Aria Hamidi Sriman Manyam

Hyunjin: LASSO, KNN, Ridge and Linear Regression, ElasticNet, Data Preprocessing

Aria: Finding an Applicable Dataset, Data Transformation, KNN, SVM Model (just for testing), Interpretation

Sriman: Report Building (QOI, Results, Takeaways), LASSO, Data Preprocessing

0.1 Introduction

0.1.1 Background - Motivation and Problem at Hand

As seniors ready to graduate and move on to professional careers in engineering, we have learned that the task of applying and getting a job that not only fits our background but pays accordingly is a difficult one. Along this journey there are multiple obstacles that include reaching out to countless recruiters, submitting applications, and going through enduring interview processes. However, this experience leads to the most important question that recruiters and hiring managers place on our shoulders - what is your desired pay? Now this is a difficult question to answer. Although you can research online on websites such as levels.fyi or Glassdoor to calculate the compensation you "should" be getting is common, a lot of the times this has proven to be incapable. These websites show that there is a large range of qualified offers for professionals with similar academic backgrounds, thus indicating that it is nearly impossible to accurately answer that question based on credentials. Companies and hiring managers tend to use this unknown to lowball incoming applicants and end up profiting off of that. Keeping this in mind, our mission for this project was to figure out what type of model would accurately predict the expected salary for an incoming engineering graduate based on their academic performance and personality characteristics. We were able to find a sizeable dataset that provided the information for us to train our models. Based off of the explanatory characteristics we used - which include GPA, Degree, Test Results, and work habits(i.e. Personality traits) - we hypothesized that the KNN method would be the most effective in predicting the expected salary for any given engineering graduate because it is nonlinear. After trial and error with various classification and regression models we were able to conclude that solely using classification models, the expected salary can accurately be predicted based on these factors alone using a classification model over regression.

0.2 Methodology

0.2.1 Dataset

The dataset that we used to implement our hypothesis is: train.csv. This dataset contains 33 explanatory variables surrounding a student's academic background.

0.2.2 Data Description - Names are Bolded

ID: A unique ID to identify a candidate, **Salary**: Annual CTC offered to the candidate , **DOB**: Date of birth, **10percentage and 12percentage**: Overall grades obtained in 10th and 12th grade exams (high school), **10board and 12board**: the school board the candidate used in that relative grade, **12graduation**: Year of graduation, **CollegeID**: Unique ID to identify the university the student attended, **CollegeTier**: Colleges with an average AMCAT score above a threshold are tagged as 1 and others as 2, **Degree**: Degree obtained/pursued by the candidate, **Specialization**: Specialization pursued by the candidate, **CollegeGPA**: Aggregate GPA at graduation, **CollegeCityID**, **CollegeCityTier**: Population based Tier, **CollegeState**: College State, **GraduationYear (Bachelor's), English**: Scores in AMCAT English section, **Logical**: Score in AMCAT Logical ability section, **Quant**: Score in AMCAT's Quantitative ability section, **Domain**: Scores in AMCAT's domain module, **ComputerProgramming**: Score in AMCAT's Computer programming section, **ElectronicsAndSemicon**: Score in AMCAT's Electronics Semiconductor Engineering section, **ComputerScience**: Score in AMCAT's Computer Science section, **MechanicalEngg**: Score in AMCAT's Mechanical Engineering section, **ElectricalEngg**: Score in AMCAT's Electrical Engineering section, **TelecomEngg**: Score in AMCAT's Telecommunication Engineering section, **CivilEngg**: Score in AMCAT's Civil Engineering section, **Conscientiousness, Agreeableness, Extraversion, Nueroticism, Openesstoexperience**: Scores in sections of the AMCAT personality test

0.2.3 Preprocessing the data

We started our tests by preprocessing the data - which included correlation and colinearity testing to identify what models would best fit our dataset. The results of those tests are shown below; it suggests that due to

a lack of direct correlation amongst the columns in our set, a non-linear model would better suit our set. Keeping this in mind, we chose to use non-linear methods KNN, Ridge Regression, LASSO, and Elastic Net to test our hypothesis.

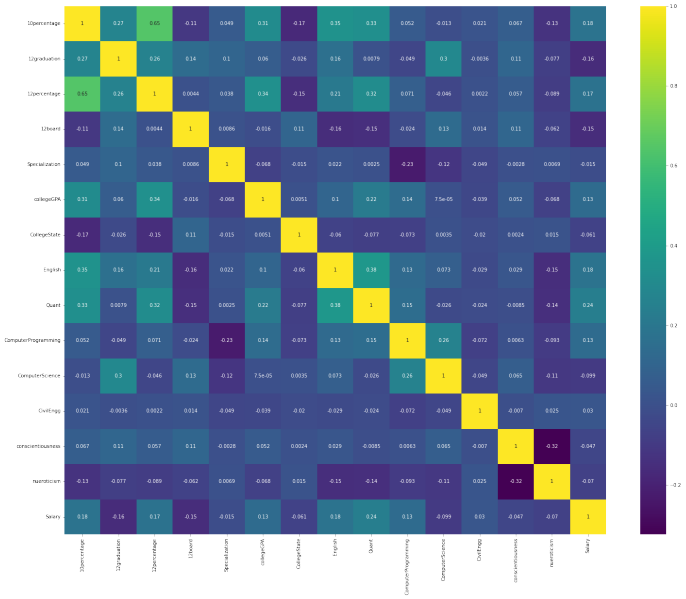
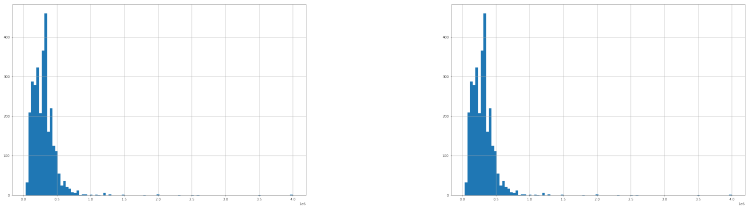


Figure 1: heapmap for correlation

We performed a square root transformation to remove the relationship between variability and mean, and logarithmic transformation to remove the relationship between variability and mean.



(a) Salaries before Transformation

(b) Salaries after sqrt Transformation

Figure 2: salary before and after transformation

0.3 Implementation Details

In order to remove insignificant features, we performed a recursive feature elimination. Using the RFE feature selection algorithm, we dropped 19 columns based on ranks of and left with more or most relevant features in predicting the target variable. Since we intend to output our data in continuous form and we don't want it to be binary, we didn't switch the type of the variable Y train (salary). With that background, we decided to use multiple regression algorithms instead of classification algorithms and compare them.

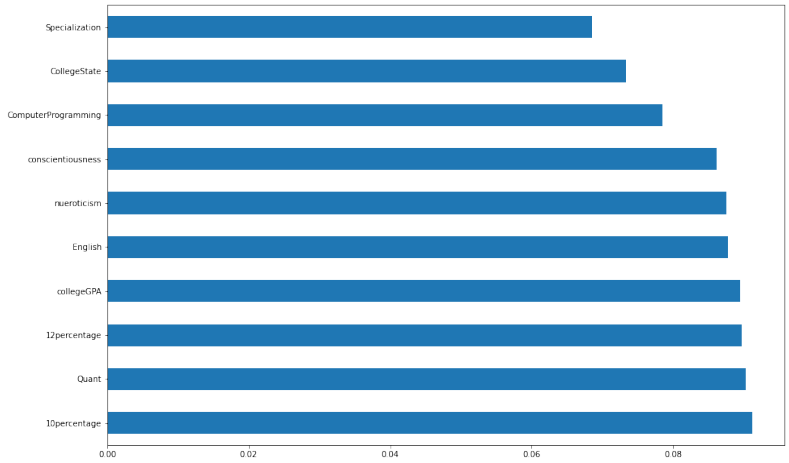
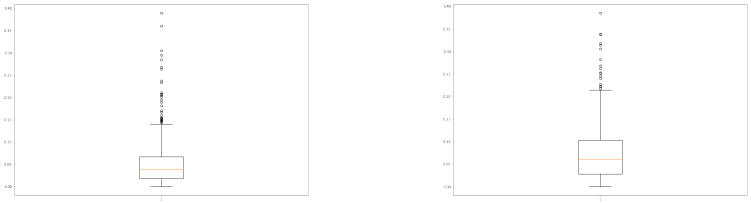


Figure 3: Bar graph of 10 features with largest importance

Since we intend to output our data in continuous form and we don't want it to be binary, we didn't switch the type of the variable Y train (salary). With that background, we decided to use multiple regression algorithms instead of classification algorithms and compare them.. For conducting the regression algorithm modeling, we transformed our X and Y training data to improve the data quality before testing different models. The first classification algorithm we implemented was KNN regression. We found the best K value when we witness an increasing number of errors after that K and at that point we know we haven't exceeded the number for K too far. We ultimately found K = 3 to be the best count of nearest neighbors. Furthermore, when we used linear regression as a regression algorithm, we fit the training data in a linear model and then calculated the error in the prediction in all data points and the coefficient of determination besides the mean squared error and also plot a boxplot illustrating the outliers and the of errors in a visible way.

Since we observed some multicollinearity in our dataset and we wanted to penalize non-related features, we decided on the implementation of the regression algorithm, LASSO Regression. In order to extend the process, we fit the LASSO regression model by using X and Y training data and then use the model to make prediction by comparing the LASSO predictor values to our testing Y, and suggested the error through a boxplot. The most similar regression algorithm to LASSO we could work on is Ridge. We approximately did the similar process and computed the mean squared errors and compared the errors to LASSO even though we guessed the errors of these 2 methods have the potential to be very similar. As a combination of LASSO and Ridge regression algorithms, we used another regression algorithm called Elastic Net Regression which required a different type of modeling. In order to fit our transformed training data, we defined a model evaluation method. We used Elastic Net Regression to observe whether it's the critique on LASSO or not. Also, we sketched a boxplot for the error indication purposes.



(a) KNN (b) Ridge Regression

Figure 4: Boxplot for prediction errors

0.4 Results and Interpretation

0.4.1 Linear Regression

Our linear regression resulted in a relatively low MSE of 0.004632 and model score of 0.211246.

Coefficients for Linear Regression							
Variable	10percent	12grad	12percent	12board	Spec.	collegeGPA	CollegeState
Coefficient	0.008077	-0.012507	0.009506	-0.007829	0.002843	0.003617	-0.002407
Variable	English	Quant	ComProgram	CS	CivilEngg	consc.	nueroticism
Coefficient	0.010433	0.015022	0.008279	-0.006064	0.005686	-0.002189	-0.003406

0.4.2 Ridge Regression

Our Ridge Regression resulted in a relatively low MSE of 0.004630 and model score of 0.242705. This indicates that Ridge Regression was a better predictor than Linear Regression.

0.4.3 Lasso Regression

Our LASSO testing resulted in a relatively low MSE of 0.004630 and model score of 0.242675 - nearly identical to that of Ridge Regression. Apart from this, the R-squared value for our training set is 26.89.

Coefficients for Ridge Regression							
Variable	10percent	12grad	12percent	12board	Spec.	collegeGPA	CollegeState
Coefficient	0.008069	-0.012355	0.009417	-0.007800	0.002785	0.003636	-0.002414
Variable	English	Quant	ComProgram	CS	CivilEngg	consc.	nueroticism
Coefficient	0.010373	0.014934	0.008204	-0.006043	0.005621	-0.002177	-0.003387

Coefficients for Lasso Regression							
Variable	10percent	12grad	12percent	12board	Spec.	collegeGPA	CollegeState
Coefficient	0.008071	-0.012443	0.009473	-0.007799	0.002758	0.003572	-0.002363
Variable	English	Quant	ComProgram	CS	CivilEngg	consc.	nueroticism
Coefficient	0.010396	0.015027	0.008207	-0.006017	0.005614	-0.002109	-0.003325

0.4.4 Elastic Net

Our Elastic Net testing resulted in a relatively low MSE of 0.005877 and a mean MAE of 0.062 (0.002). Since the Errors are higher on Elastic Net, we will not choose to use this as our final model.

0.4.5 KNN

Our KNN testing resulted in a very low MSE of 0.0029712 and a model score of 0.355275. The results of our KNN test clearly indicate that it was the most effective algorithm due to the higher model score and much lower MSE, thus it is our most accurate model.

0.4.6 Conclusion

After testing and analyzing the multiple models we chose, our hypothesis was indeed true! We concluded that the KNN algorithm would be the most accurate because it has better scores and lower MSE. The predicted salary values (Y) were much closer to the actual values. Furthermore, since our dataset includes a variety of columns with different data types, a linear model would not fit well - and KNN being a non-linear model

makes the most sense. We learned that even though we were right about KNN being the most accurate model, it still did not show to be an incredibly strong model with a score of 0.355. This is because salary isn't directly tied to academic accolades and personality traits and indeed relies heavily on external factors such as presentation skills, work habits, response levels, and more. That being said, our model helps our situations, as soon to be college graduates, by providing a rough estimate of what salary a student should anticipate based on their credentials and will help students in the future when determining their valuation to a company for a professional engineering role.