Chasing the Bag: Factors That Influence Offered Higher Starting Salaries

Berk Aksoy, Vryan Feliciano,
JC Roa, Jade Vasquez
CSS 100
Professor Mohammad Samavat

Abstract

Our research uses a dataset of applicants who applied to a school in India. We want to investigate what factors may influence a postgrad program or company to offer a higher starting salary. We define a high starting salary as the top quartile of the IQR of all offered starting salaries in the dataset, which was 300,000 rupees. After data cleaning, we used one-hot encoding to preprocess the data in order to include categorical variables in our machine learning algorithms. We then used the train-test-split method to create subsets of the data to train and test our models. We used linear regression tables to present salary and its correlation to multiple varying factors. Our research also used logistic regression to help identify which factors contribute to higher salary, and RandomForest to evaluate the individual influence of features. Our model found that the most important feature in determining who was offered a high salary was the employability test percentage. This research is important because we can find the best way to achieve a higher starting salary.

Introduction

The Campus Recruitment dataset was uploaded by Ben Roshan for practice using Python and R at a business school in Bangalore, India. The author of this dataset is Dr. Dhimant Ganatara, and it consists of information about students at a university. There are 215 subjects, with 15 columns of features. Each applicant has factors related to their competitiveness: their school performance during high school, undergrad, and post-grad; whether or not they have work experience, etc, and these were some of the features in the dataset. Whether the applicant was placed or not, and the offered starting salary were also important features.

Our project aims to find the best feature that will predict whether or not a candidate was offered a high salary when placed. Since the dataset was uploaded from a university in India, we are interpreting salary in rupees, the national currency of India. We defined a high salary as the upper quartile of the interquartile range for the salary column, 300,000 rupees.

Due to the scope of the project, the dataset was cleaned of applicants who did not get placed into the program; these applicants did not receive a salary, and thus will not be needed to do the analysis. Aside from this, normal data cleaning considerations regarding missing values and outliers were done. Visualization techniques were used to preemptively identify relationships between categorical features for further analysis.

Methods (JC)

Data Cleaning

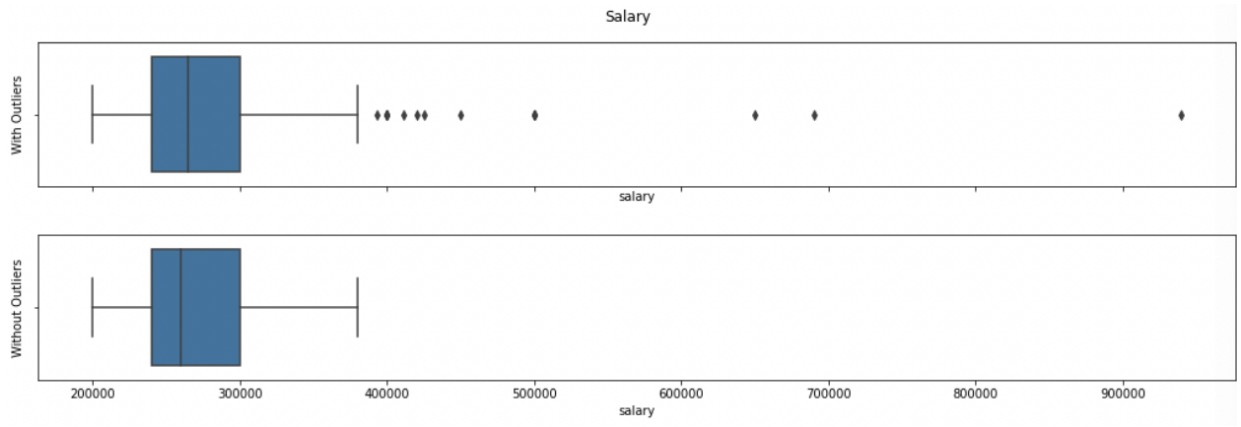
We began the data cleaning process by checking if the dataset had any null values and found that the only null values we found were within the “salary” variable. We then verified that instances with a status of “not placed” also did not have any salary values. Given the context of

the dataset, it would make sense that the salary variable would have the only missing values because it was only filled in if an applicant had been placed. Since our project explores what features affect being offered a higher salary, we removed the instance where applicants were not given a placement from the dataset. We then removed the "status" column from the dataset because it no longer had a purpose for our analysis; all instances have the same entry for this feature.

Features such as "sl_no", "ssc_b" and "hsc_b" were removed from the dataset because they provided no additional information regarding the applicant; they either served to give a serial number identification or noted what board of education they belonged to for their secondary schooling. There is no need to organize data instances by their serial numbers, nor interpret the board of education an applicant belonged to because they are irrelevant to getting accepted into a business school, especially when compared to other features such as specializations or degrees during their schooling.

Data Pre-processing

We created two boxplots of the salary data to observe how spread out the values were. One boxplot contained outliers and the other did not. We were able to identify outliers by using the IQR of the dataset. After visualizing the data with and without outliers, we decided to use the dataset that did not contain outliers so that we could avoid the data being skewed.



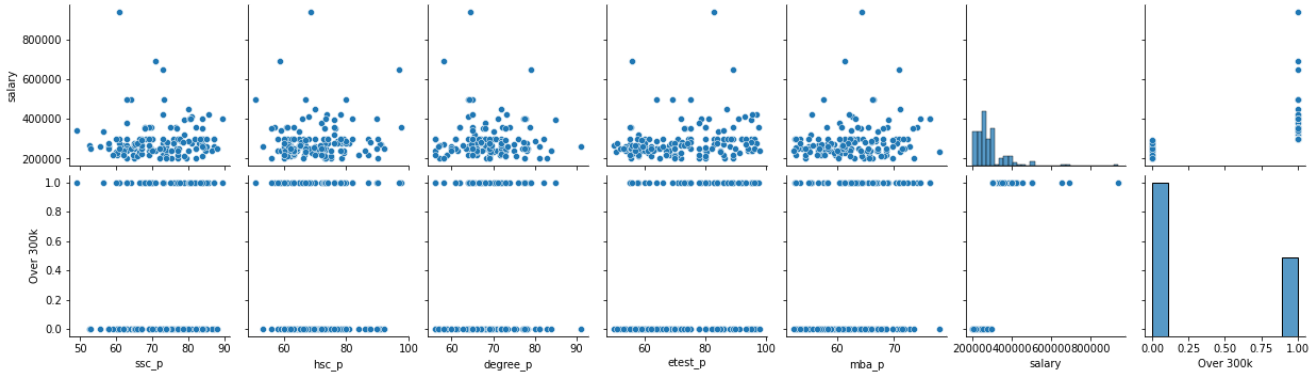
The process of defining what would be considered a “higher salary” was also performed through using IQR; we used Q3 (300,000 rupees) as the boundary that determines a “high salary”. The IQR of the dataset before and after removing any outliers remained similar with both having a Q3 of 300,000.

Many variables in the original dataset were nominal and in order to run statistical tests, these nominal variables had to be manipulated in order to be usable. Thus, we used label-encoding for variables that could be seen as binary such as “gender”, “workex”, and “specialization”; gender was either male or female, work experience was either yes or no, and specialization was either marketing and finance or marketing and human resources. In these cases, the 1-case would be male, yes, and marketing and finance. We also used one-hot-encoding to manipulate other nominal variables such as “hcs_s” and “degree_t”. In order to do this, we had to dummy code variables for each unique entry for the variables. We then removed the original columns and added the dummy coded variables as columns. Regarding our representation of high salary, we created a binary variable called “Over 300k” to be able to reflect our 300,000 rupee definition in our machine learning models.

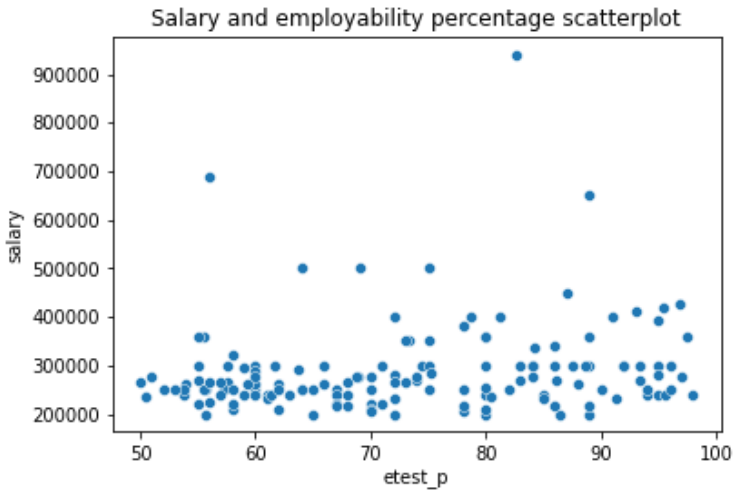
Results

Data Visualization

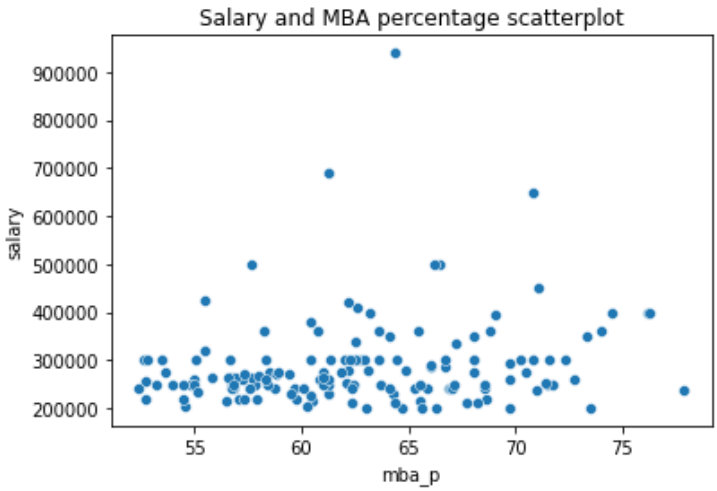
Visualizations were used to allow for a better understanding of the spread of the features within the dataset and pinpoint relationships of interest. We first visualized a pairplot to find interesting correlations, shown in the figure below.



Since we are mainly looking at which variables correlate with salary, we created additional boxplots and scatterplots to compare this with the other features. A scatterplot of salary and employability test percentage showed a slight positive correlation between these two features, meaning that those who scored higher on the employability test were offered higher starting salaries if they were placed.

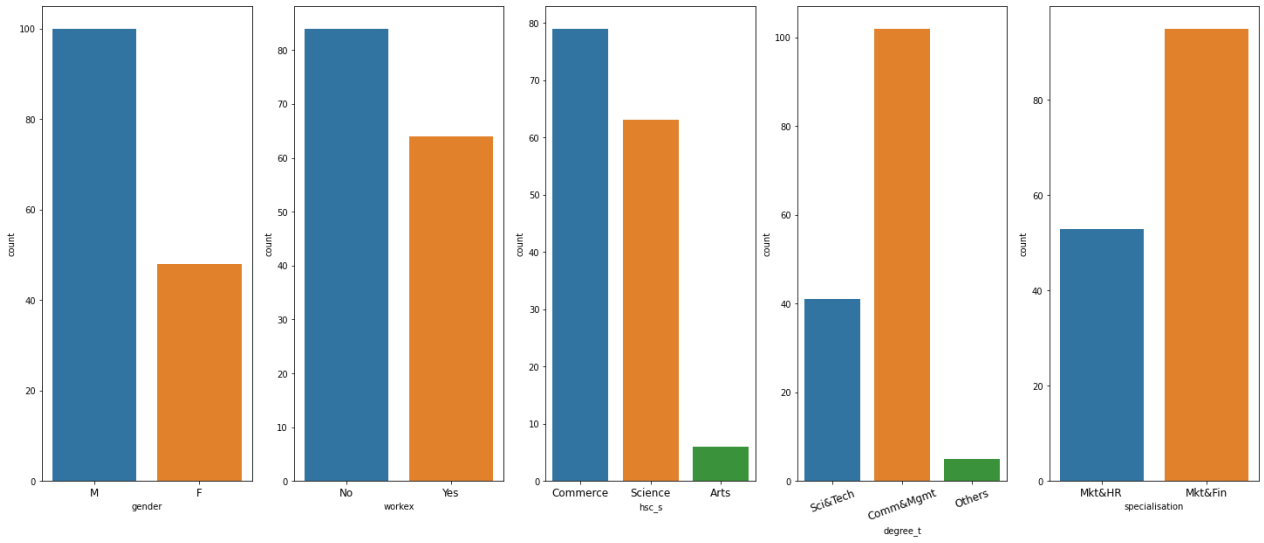


There was also a positive correlation between salary and MBA percentage, as shown in the scatterplot below.



This visualization tells us that the applicants who scored higher on this exam were offered higher salaries.

We also created several boxplots to compare salaries within different categorical data, such as salary based on undergraduate degree type. The boxplots with categorical features against salary did not show clear results. We did not see a clear difference in higher salaries based on specialization and degree type. After creating a bar graph with count of each of these features, we concluded that this is because there is an uneven distribution of these types.



For example, there are nearly twice as many applicants with a degree in commerce & management than science & technology. This makes it hard to analyze whether these factors influence salary offered.

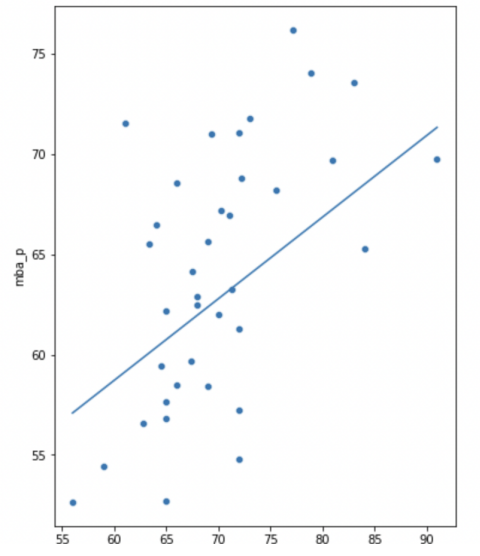
We also used line plots to provide a second view between some of the categorical and quantitative data. Line plots made it much easier to visualize the spread of income amongst applicants using different factors such as gender and work experience. We also continued to incorporate line plots in the visualization of mba_p to salary to realize that there was no direct correlation between mba standing and salary.

Machine Learning Models

The goal of our machine learning models was to find the features that best predicted whether or not a hired applicant would get offered a high salary, to which we defined as over 300,000 rupees. As such, we had to adjust the features used in our machine learning models. The features “salary” and “Over 300k” were removed from the X parameter, because “salary” is the actual target data and “Over 300k” was the target classifier we were using. Because “Over 300k” is our target classifier, it was assigned to y only. When the data was split into training and tests, an 80:20 split was used.

Linear regression tables were used in the case of very evident signs of a predictive model for us. Unfortunately, our returned R^2 values were quite low but were able to show best fit lines in accordance to multiple different factors. The downside to using linear regression for some of our test data was that categorical variables could not be representative of their correlation. There is a vast amount of data regarding mba_p, degree_p, salary, etc. that makes it difficult to get the best predicted linear regression. Nevertheless, the training and testing results signified that there was no indication of error

```
Training R^2= 0.202931266589516
Testing R^2= 0.2753267689479655
y =  0.4063250836602163 *x +  34.33406026631471
```



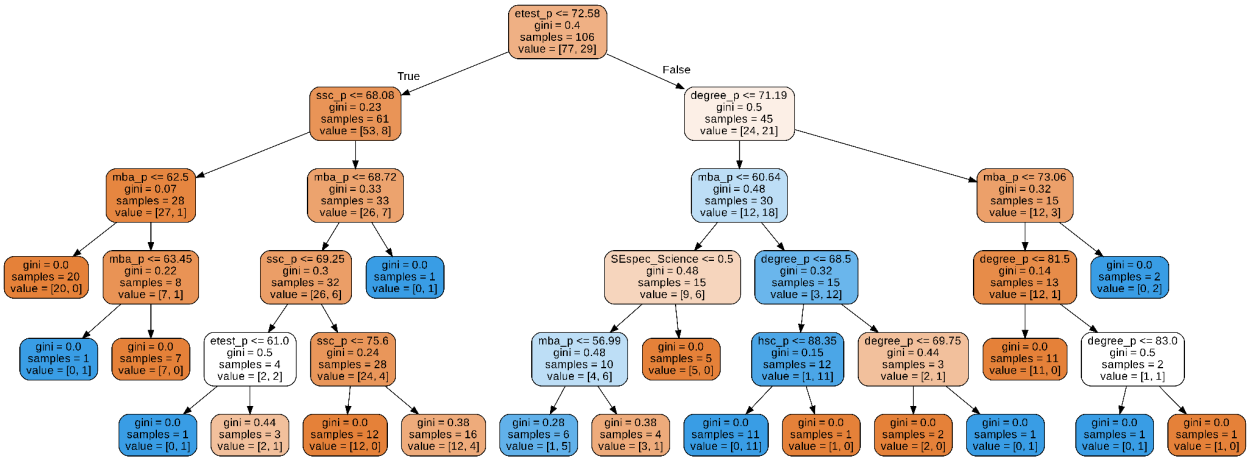
such as the example below; as long as testing R^2 was greater than training R^2.

A logistic regression classifier was used to evaluate how well our model classified a data instance to either have or not have a high salary. The model's accuracy was 70.3%. To evaluate the model's performance, a confusion matrix and a subsequent classification report was generated. The confusion matrix interpretation is as follows: 18 instances were correctly classified as not having over 300k, 3 were wrongly classified as having 300k, 5 were wrongly classified as not having over 300k, while only 1 was correctly classified as having a salary over 300k. Looking at the classification report, the logistic regression classifier's precision and recall scores are 52% and 51% respectively. This means that, from the 70% accuracy initially generated, the model correctly predicts having a high salary 52% of the time and correctly classifies 51% of the data.

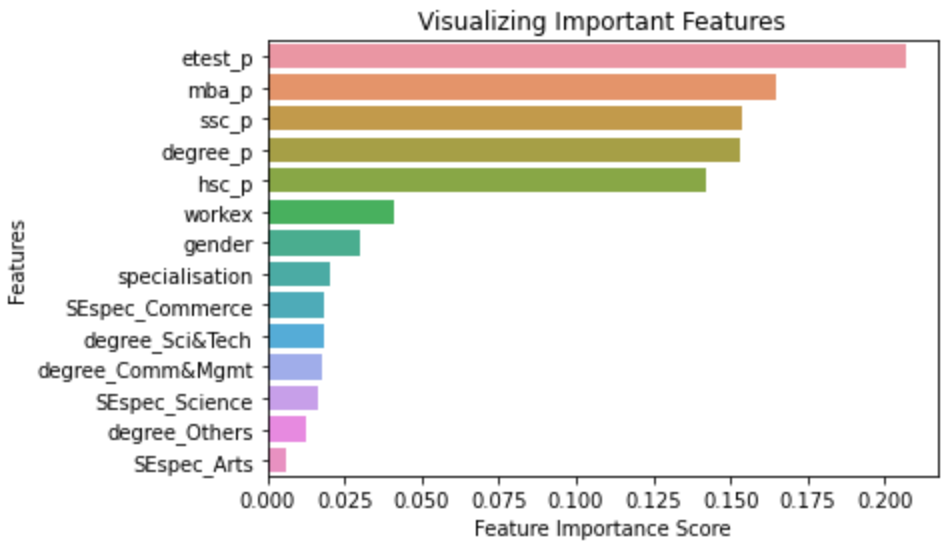
```
Confusion Matrix:
 [[18  3]
 [ 5  1]]

Classification Report:
              precision    recall  f1-score   support

           0       0.78      0.86      0.82        21
           1       0.25      0.17      0.20         6

    accuracy                           0.70        27
   macro avg       0.52      0.51      0.51        27
weighted avg       0.66      0.70      0.68        27
```

A Decision Tree classifier was run to see if our model would perform differently under another classification model, without any set parameters. The Decision Tree classifier's accuracy was 66%, less than the Logistic Regression Classifier's accuracy. A precision, recall, and F1 score were also calculated for the Decision Tree: respectively, 28.5%, 33%, and 30.8%. From the initial 66% accuracy, the Decision Tree correctly predicts having a high salary 28.5% of the time,

and correctly classifies 33% of the data. According to the F1 Score, our Decision Tree classified correctly 30.8% of the time. Next, the Decision Tree was visualized to see how the classification occurred. Six sets of branches were produced with five splits based on ssc_p and degree_p, mba_p, then finally either etest_p or hsc_p. The most basic split in the DecisionTree in determining whether or not an applicant would have had a high salary occurred under degree_p compared to ssc_p, where at least 71.19% of data was evaluated and classified into a branch.



A RandomForest classification model with default n_estimators = 100 was run to see if a more complicated Decision Tree classifier would produce a better model. The RandomForest's accuracy was 62.9%, similar to the Decision Tree's accuracy score. After running the RandomForest classifier, we evaluated the classifier using pd.feature_importances_ to see which features best predicted having a high salary. From the subsequent horizontal barplots, it seems that the employability percentage (etest_p) calculated by the business school had the highest bearing on the salary offered to an accepted applicant, followed by their performance in their post-grad programs (mba_p) and performance early on in high school (ssc_p).



Discussion

From our analysis, we found that the most effective way to predict whether someone will be offered a salary greater than 300,000 rupees is by the employability test percentage according to the RandomForest classification model. Additionally, an interesting trend occurred: performance during school mattered more compared to the specific degree and specialization an applicant held. This trend may be due to what features apply to who: the features describing percentiles (etest_p, mba_p, etc.) are features of all applicants; in contrast, the specializations and degrees of an applicant that were one-hot encoded are only specific to an applicant, and do not have influence on the models akin to the percentiles. Regardless, the results are descriptive: if one is looking to be offered a higher starting salary, it most certainly pays to both have a postgraduate degree and to have higher academic performance.

As descriptive as this is, however, it is prudent to mention that the RandomForest classification model itself did similarly to the Decision Tree classification model, even on subsequent reruns after factory resetting the Jupyter Notebook. Both did worse than the Logistic Regression. To explain, the similar accuracy scores between Decision Trees and the

RandomForest model suggest that the Decision Tree model was sufficient enough as a classification model. This may be a reflection of the RandomForest model overfitting our dataset, especially in terms of complexity and sample size. In comparison with Logistic Regression, the Logistic Regression's classification algorithm is less complicated than either Decision Trees or RandomForest, because only one classification occurs.

The overall performance of the models was underwhelming after a more in-depth look. Initially, for both the Logistic Regression and the Decision Tree classifiers, favorable accuracies were produced. However, when that accuracy was broken down by precision, recall, and F1 score, the accuracies became less favorable as the aforementioned measurements of performance produced low percentages.

Future projects may further refine the features used in the RandomForest model to eliminate complexity in the data by either combining or removing them, or simply increase the sample size. The Decision Tree may potentially have had a different accuracy score depending on the criterion used to measure the quality of a split; by default, sklearn uses gini, the other option being entropy. Beyond the two aforementioned classification models, it may not be efficient to try more complicated classification models as seen by the performance of the ones used in this project; the culprit here may be the dataset, and so more complicated models may not contribute much.

A limitation from our research is that our sample data size was small. Our dataset initially had N = 215 applicants; after data cleaning, the sample size reduced to N = 148. A larger sample size would help to generalize our results, but more importantly help determine whether the trends we found in our analysis actually exist, or only occurred because of the sample size. Another limitation is that our dataset consisted of candidates from only one country, whose preferences in

who to offer higher starting salaries may differ culturally compared to other countries. By including participants from multiple countries, we can better find the best predictor for a high starting salary because, then, the results would be better generalizable.

Contributions

Vryan Feliciano - Machine Learning Models, Discussion, Abstract

Jade Vasquez - Introduction, Visualizations, Abstract

JC Roa - Data Cleaning, Data Pre-processing, Abstract

Berk Aksoy - Introduction, Linear Regression ML, Abstract

Link(s):

- Dataset on Kaggle
- Github