In [ ]:

```
#import relevant packages
from IPython.display import Image
```

Fictional Context For My Data Science Project as a Springboard Data Science Fellow for my Capstone 2 Project as of August 8th, 2022 (2022-08-08)

Student Placement For An Institution of Higher Education

1) Problem Formulation

An organization for higher education has requested that data scientist takes a look at a sample of their student information to for two main purposes:

- A model that can predict whether a student will find employment or not based on features provided the organization
- Some preliminary work on model that can predict salary to see if a model can be considered for further contract work

The goal is to provide students more information regarding their career trajectory so they can make more informed choices. While the data set we can used is locked (as the organization could only get an initial sign off from so many students to use their information for the model building stage), we are free to explore any model we wish in terms of developing the model.

2) Data Source

The data source for our project can be found here as provided by the project sponser. We are fortunate that the data is relatively clean but does require some data wrangling to get into a state that we can begin to use for our data science project.

3) Data Wrangling

For the full workthrough, please check the following notebook here

We were given the following 14 features for 215 observations to develop our models on:

- sl_no: Is the serial number for an observation (student going forward), which is effectively an index key
- gender: Provides the gender for a student (M for Male, and F for Female)
- ssc_p: The percentile Rank of the student in 10th grade
- ssc_b: The Board of Education associated with the student's 10th grade class (Central or Other)
- hsc_p: The percentile rank of the student in 12th grade
- hsc_b: The Board of Education associated with the student's 12 grade class Central or Other)
- hsc_s: The specialization of the student in 12th grade (Commerce, Science, or Other)
- degree_p: The percentile rank of the student at completion of their undergradute degree
- degree_t: The undergraduate degree that the student acquired which are Communications & Management (Comm&Mgmt), Science & Technology (Sci&Tech), or Other
- workex: Whether the student has work experience or not
- etest_p: The percentile rank of thes student on their employability test
- specialization: The specialization of the student which are Marketing & Finance (Mkt&Fin) or Marketing & Human Resources (Mkt&HR)
- mba_p: The percentile ranking of the student once they've completed their MBA
- status: Whether a student has been placed (aka outcome) which are Placed or Not Placed
- salary: The salary of a student upon placement

"Status" will be the target feature for our first model to predict whether a student is employed or not. "Salary" will be the target feature for our second model to predict earnings.

The main data "wrangling" and "cleaning" procedures we needed to utilize are listed below:

1. Some categorical features that had only two possible outcomes (such as gender) needed to be converted to a binary variable that could be used by various models. For example, we converted the "gender" feature to a "female" feature which is "0" if the student is male, and "1" if the student is female. This procedure was repeated for other features including (but not limited to) "ssc_b", "hsc_b", "status", and "workex." In addition, feature names were adjusted for readability (e.g., "workex" was converted to "work_experience"
2. For features that were categorical features with more than two possible outcomes (such as "hsc_c"), we create binary features to capture the different outcomes while leaving one category out to act as the base category. In the case of "hsc_c" which had three possible outcomes (i.e., "Commerce", "Arts", and "Science"), we leave the "Arts" category out as that becomes the base category and have binary features for "Commerce" and "Science"
3. For the "salary" feature, if the student was unemployed, the salary was listed as "NaN" which correlated to a lack of employment. For the purposes of our modeling, the "NaN" outcome was replaced with zeroes.
4. We checked any features that had an associated range (e.g., "hsc_p") to see if there were any outliers we'd need to consider. Outliers were determined using the IQR test of being either less than the 25% percentile value minus 1.5x the IQR or greater than the 75% percentile value plus 1.5 the IQR

Once we completed all our procedures, we wrapped up data wrangling and moved to Exploratory Data Analysis

4) Exploratory Data Analysis

For the full-workthrough, please check the following notebook here

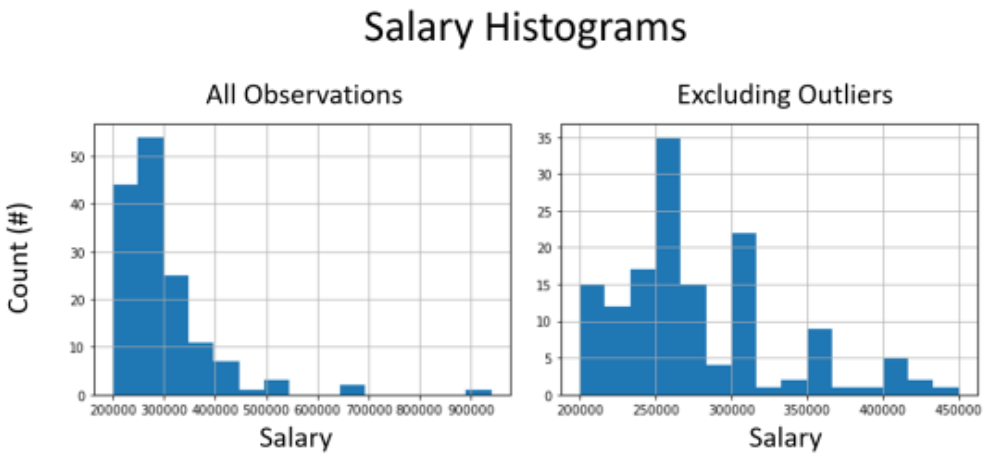
Once our data was wrangled and put into a proper form, we begin exploring our data to get a better understanding of our data and considerations for modeling.

- One of the first things we noticed was that certain binary features (e.g., "female", "hsc_Central", etc.) didn't have a balanced 50/50 split between observations. While it's not necessarily important, it is something to

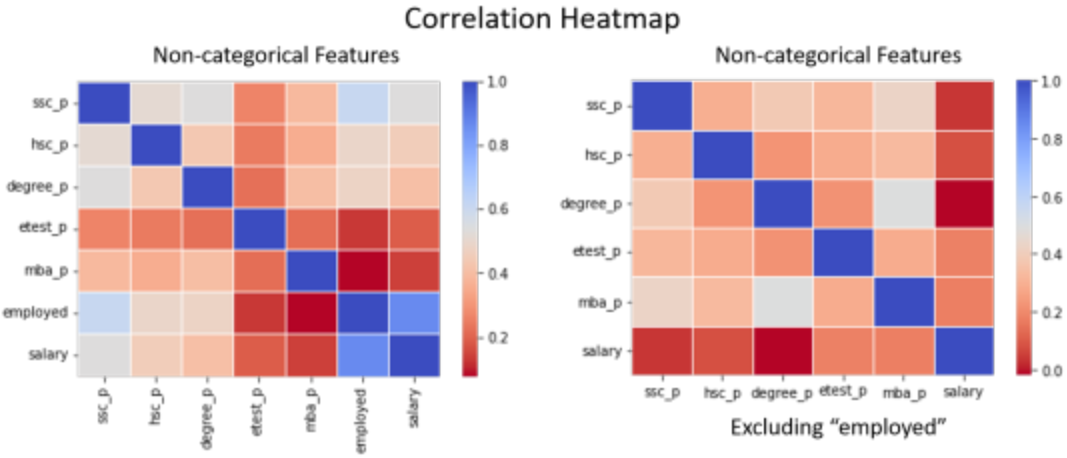
keep in mind to check as once we begin splitting the sample for training and testing data sets, we may get some splits that are very sparsely populated in terms of representation of certain types of students.

- In addition, any feature with an associated range, we checked via histogram to get an overall sense of the distribution. Nearly all our features (with the exception of "salary") had a relatively normal distribution shape with "etest_p" being the only other variable that appeared to be more uniform distribution.

- Salary appeared to be heavily skewed to the right (excluding the zero observation) regardless of whether we included the most extreme outliers our not as represented below. This may require us to consider the distribution of salary when we are developing models (as well as the scale of the values)

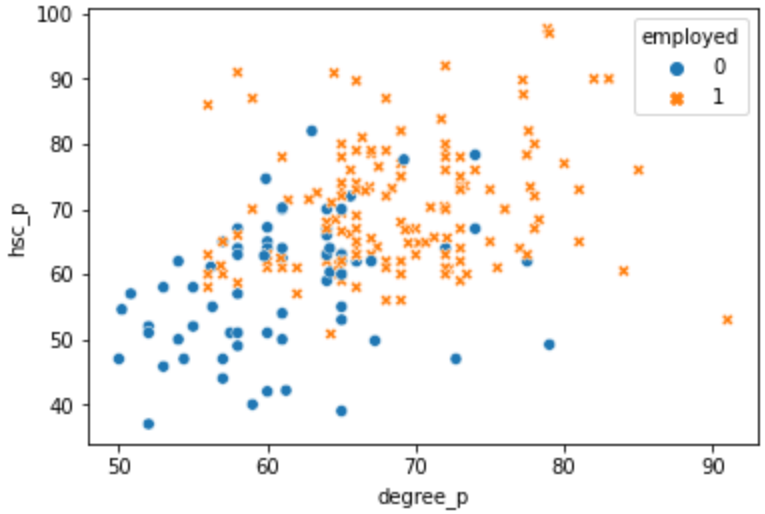


- Now that we have a better understanding of all our features on their own, we begin looking at our features in relation to our target features as potential predictors. An easy place to begin here is with correlation heatmaps get a better understanding of magnitude and direction of relationships (below).



Exploring correlations with "employed"

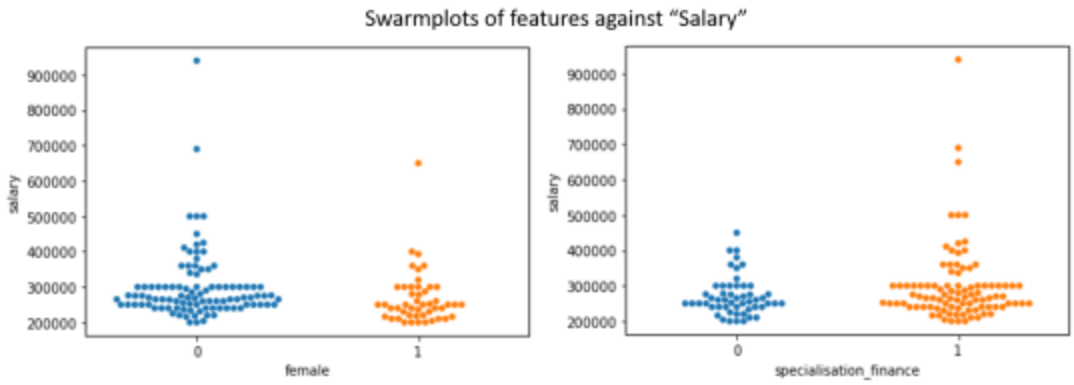
- We first explored features that may be correlated with "employed" as the first model we need to build is associated with predicting whether a student will be employed. In exploring the data, we find that both the "degree_p" feature and the "hsc_c" feature appear to be correlated with the employed feature. As such we would expect these feature to help predict employability



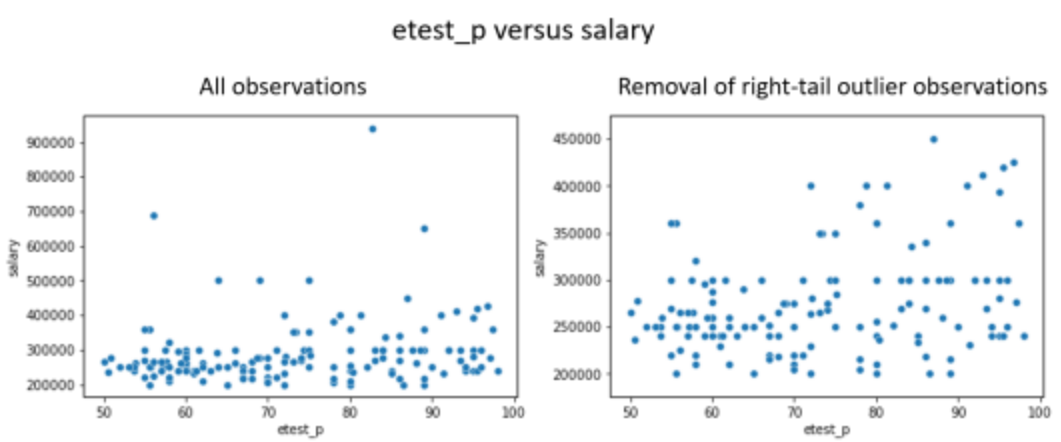
- In addition, we ran some t-tests to check if other categorical features had some predictive power. For example comparing students with a finance specialisation versus students with a HR specialisation gave us a p-value of <0.001 in terms of whether a student was employed or not. This would imply that a student's specialization does have a statistically significant effect on predicting employment status.

Exploring correlations with "salary"

- Since salary is a continuous feature, we want to check how some categorical features relate. An easy way to visualize this is with a swarmplot for some features like "female" and "specialisation_finance." One of the main trends we can see is that there are some outliers that drag the swarms upward for one category or another and that there is a massive congregation of observatiosn in both categories around the 200k to 400k band of salaries



- In addition we can create some straightforward scatterplots of continuous features against "salary." One example would be for the feature "etest_p" which we fine



Notice the y-scale is different for each graph as the right-hand graph removes the extreme high value outliers

- We find some correlation for feature like etest_p, but not the strongest correlation.

Now that we've had a chance to explore our data and get a better sense of its structure and the relationships our features have with each other. The next step is to prepare our data for model usage

5) Preprocessing

For the full-workthrough, please check the following notebook here

In the preprocessing step our two main objectives are to:

1. Remove any features that are not helpful in making predictions
2. Transform our data to be more appropriate for model usage.

In this step, features like "sl_no" were removed because the in essence act as an additional index that provide no real information. In addition, for categorical features with n different categories, at least one category had to be remove (e.g., Commerce, Science, and Arts) to make sure there were no co-linearity issues (i.e., drop Arts). One of the nice things of categorical binary features (0 or 1) is that they are naturally scaled in a nice manner.

For the other continuous input features in our dataset, a Min Max Scaler approach was utilized to convert them to a range between 0 and 1 which puts them in line with our binary features

6) Modeling

For the full-workthrough, please check the following notebook here

Now that our data is fully prepared for modeling, we can begin developing our prediction models. First, we'll work on predicting employment

Modeling Employment

Logistic Regression

Since we are working with a binary classification problem with predicting modeling, a good place to start is logistic regression. However, to do this properly, we also need to split our data set into a training set and a testing set. For our modeling, create a testing set that is 25% of the original data set. In addition, using Grid Search, we determine that the "C" parameter for our model should be "1." The end result is an accuracy measure of 88.89%

Random Forest Classifier

Another good model to use for a problem like this is a Random Forest classifier. Using the same training/testing split and Grid Search, we find the best "max depth" to be 10 and the optimal "number of estimators" to be 103. Once we complete training the model, we get an accuracy measure of 81.48%

Gradient Boosting Classifier

One last model to consider is a Gradient Boosting Classifier. Using the same training/testing split as before and using Grid Search, we conclude the optimal hyper parameters are a learning rate of "1" and "n_estimators" of 54. When we train the model and then test its accuracy, we get an accuracy of 79.63% which is quite low and therefore we won't further consider the Gradient Boosting Classifier.

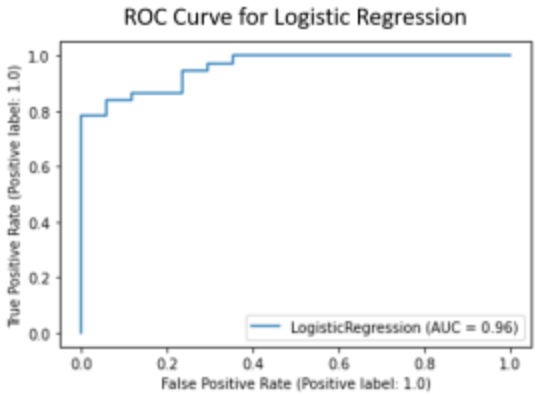
Precision Consideration

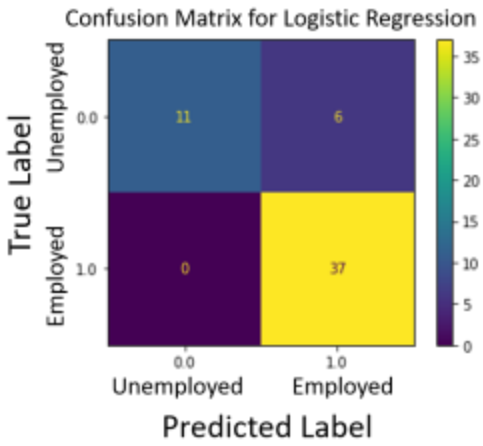
In a situation like this, we need to consider the gravity of False Negatives vs. False Positives. In general, it's probably far worse to have False Positive (predicting someone will get employement when in fact they will not get employment) as opposed to a False Negative (predicting someone will be unemployment but does end up getting a job). As such, the precision of a model will be something to consider as this is an indication of how often a model gets False Positives. For our Random Forest Classifier, find it has a precision score of 82.93% while our Logistic Regression Model has a precision score of 86.05%.

Due to the Logistic Regression Model having a higher accuracy score and higher precision score, we conclude that it is the best model for this problem and therefore recommend the Logisitic Regression Model for implementation.

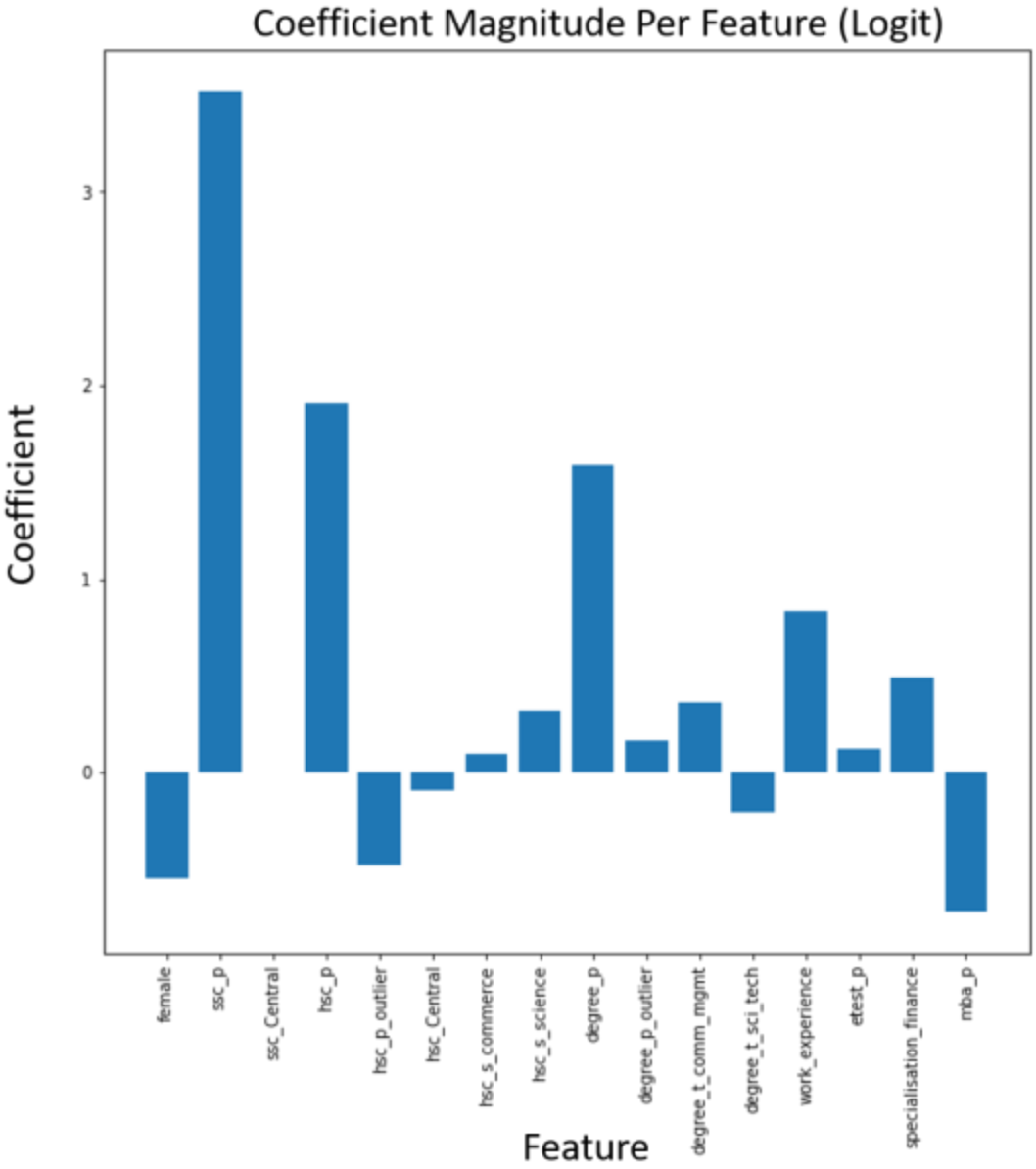
Logistic Model Evaluation

To gain some insights about our model, we can look a few aspects of it. The first two aspects we'll consider are the confusion matrix and the ROC curve of the Logistic Regression Model





The ROC curve gives us confidence that our model is performing fairly well. One consideration it does show us is that we could chance our threshold to label an onservation "Positive" (meaning predicting employement) to increase our rate of getting True Positives with the tradeoff of getting more False Positives. However, given the context of this problem where a False Positive has more costly connotations than a False Negative, we'll keep our model as is. If we check the Confusion Matrix, we can see that our model avoided all False Negatives but did end up with *some* False Positives. This may mean we can consider whether we are better off having some predictions as False Negatives if it means we can decrease the number of False Positives. Otherwise, the model performed quite well at accurately predicting student employment outcomes. One last thing we can check is the magnitudes of our coefficients to get a sense of how pertient any given input feature was in predicting employment.



When looking at the magnitudes, a few things stand out as either observations or things that could potentially warrant further evaluation:

- Being female *decreases* the probability someone will be predicted as employed which may be cause for concern at the organization. It may need to consider whether females are being given the support they need or if there are systemic factors disadvantaging females that need to rectified.
- Someone's "ssc_p", "hsc_p", "degree_p" (different testing results), and having "work_experience" appear to have the highest effect on positively predicting someone will have employment. If these are features that are determine *before* a student enters the institution, it may call into question how effective the institution is at providing value to students in terms of gaining employment. The institution may need to consider how it can support students who come in with "pre-determined" features to get them up to speed. As a recommendation, the institution would want to see pre-determined features *decrease* in terms of magnitude while seeing features determined while at the institution *increase*.
- Oddly enough, "mba_p", has a *negative* magnitude with gaining employement which seems counter-intuitive if the organization is based on business education. That said, it may warrant further investigation regarding the value of getting high marks if employers don't seem to be particularly concerned about it. This may also suggest that activities that take away from studying *but* can contribute to gaining employment (e.g., time spent on informational interviews, networking, internships, etc.) may provide more value in getting employment.

Modeling Salary

It's important to note that this charge from the sponser is simply to do preliminary research to test model accuracy per the scope of our contract. If further work on predicting salary is desired, a new contract for this project will need to be drafted.

Linear Regression

Since "salary" is a continuous variable, a good starting point is linear regression. For this portion, we conduct a new train/test split with 25% of the data being in the testing set. For this Linear Regression, we'll use Lasso regression. With Kfold regression. As a result, we get an accuracy score (or R^2) of 81.56% which would give us some confidence that predicting salary is at least feasible

Random Forest Regressor

Another regression model we can use is the Random Forest Regressor model. Using the same train/test split and Grid Search, we find the best hyper parameters to consider are a "max depth" of 2 and "n_estimators" of 134. Once we test the model, we end up with an accuracy score of 80.42%

KNN Regressor Model

A final regression test we conduct is a KNN Regressor Model. Using the same train/test split and Grid Search, we find the best hyper parameters are "n_neighbors" at 7 and best weighting scheme to be "distance". However, our accuracy score for this model comes out to be -7.28% which would imply that either this model is not well-suited for our data *or* that our data needs further processing to be better evaluated by a model like this.

Salary Regression Conclusion

Based on the accuracy scores of the Lasso Linear Regression and Random Forest Regressor, it would seem possible to develop a model to make predictions on salary and for them to be fairly accurate *but* it appears that some more processing (and potentially feature engineering) would be necessary to get accuracy scores high enough to consider the models "satisfactory."

7) Conclusion

We have successfully developed a Logistic Regression model for predicting Employment for our sponsor. In addition, the model identified some key considerations the organization may need to consider when meeting the needs of their students regarding gain employment!

As a personal note, this was a really fun Capstone project for me and I definitely learned a lot! I'm looking forward to further improving my Data Science Skills!

Cheers! Emre