Job Salary Prediction

Archit Khosla

1 THE DATASET

My task for this assignment is **Job Salary Prediction**. This is a competition on Kaggle. The data I choose is a set of job ads published in the UK. Below is the description of the data provided by the competition. (https://www.kaggle.com/c/job-salary-prediction/data)

Id - A unique identifier for each job ad
Title - A freetext field supplied to us by the job advertiser as the Title of the job ad. Normally this is a summary of the job title or role.
FullDescription - The full text of the job ad as provided by the job advertiser. Where you see ***s, we have stripped values from the description in order to ensure that no salary information appears within the descriptions. There may be some collateral damage here where we have also removed other numerics.
LocationRaw - The freetext location as provided by the job advertiser.
LocationNormalized - Adzuna's normalised location from within our own location tree, interpreted by us based on the raw location. Our normaliser is not perfect!
ContractType - full_time or part_time, interpreted by Adzuna from description or a specific additional field we received from the advertiser.
ContractTime - permanent or contract, interpreted by Adzuna from description or a specific additional field we received from the advertiser.
Company - the name of the employer as supplied to us by the job advertiser.
Category - which of 30 standard job categories this ad fits into, inferred in a very messy way based on the source the ad came from. We know there is a lot of noise and error in this field.
SalaryRaw - the freetext salary field we received in the job advert from the advertiser.
SalaryNormalised - the annualised salary interpreted by Adzuna from the raw salary. Note that this is always a single value based on the midpoint of any range found in the raw salary. This is the value we are trying to predict.
SourceName - the name of the website or advertiser from whom we received the job advert.

The dataset contains 244768 rows. 75% of the data was chosen at random and was used for training while the rest was kept for testing. There is nowhere mentioned if the salaries are in US dollar or pounds so it is referred as units per annum. There is data for about 2732 places and there are 135,436 unique Jobs titles present. The main categories that the jobs fall under are given below with their average salaries next to them

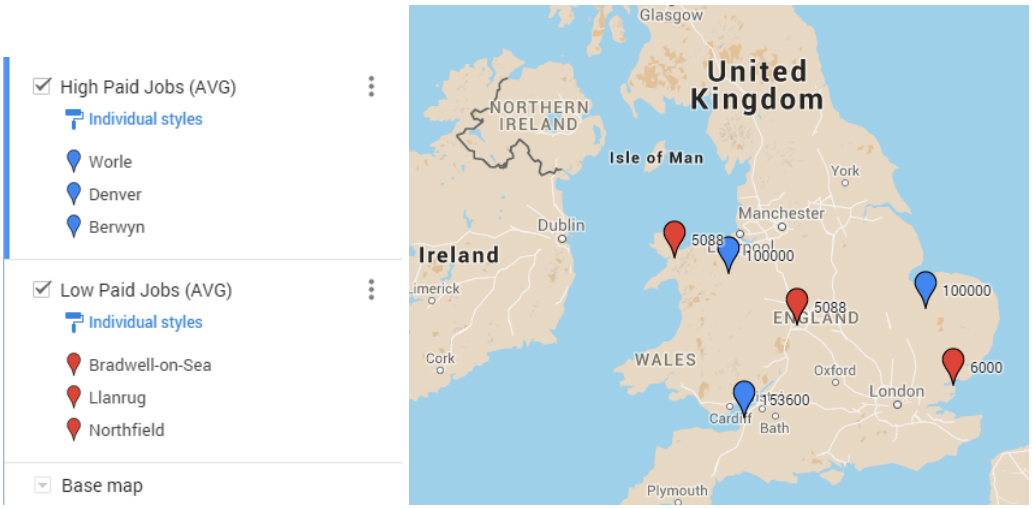
Accounting & Finance Jobs	38622.456172
Admin Jobs	20916.362130
Charity & Voluntary Jobs	28200.204936
Consultancy Jobs	36374.208544
Creative & Design Jobs	32585.487409
Customer Services Jobs	19795.438648
Domestic help & Cleaning Jobs	17492.815789

Energy, Oil & Gas Jobs	**45384.110103** (Max)
Engineering Jobs	35608.058499
Graduate Jobs	28677.438703
HR & Recruitment Jobs	32386.298070
Healthcare & Nursing Jobs	32203.188169
Hospitality & Catering Jobs	23552.462798
IT Jobs	44081.780990
Legal Jobs	42350.550210
Logistics & Warehouse Jobs	25711.287983
Maintenance Jobs	17533.320433
Manufacturing Jobs	25653.276129
Other/General Jobs	34361.716029
PR, Advertising & Marketing Jobs	35521.313285
Part time Jobs	**10514.285714** (Min)
Property Jobs	32167.056647
Retail Jobs	32851.304998
Sales Jobs	30685.809207
Scientific & QA Jobs	33950.363683
Social work Jobs	32324.809030
Teaching Jobs	27240.637182
Trade & Construction Jobs	36050.166566
Travel Jobs	23913.173143

The average salary in UK is 31,554.441 units. The maximum salary is in 'Worle' and ‘Denver’ with the value being 153,600 and 100,000 respectively. However, the minimum salary being in 'Northfield' and 'Llanrug' which is just 5,088 units. One surprising finding was that the average salary in ‘Channel Islands’ is about 71,500 which is about 40,000 above average; however, the rest of the places are about 10,000 above or below average. Channel Islands is not one of LocationNormalised feature in the data. There is another file given which gives a tree like structure of the places in UK. Something like Counrty~State~city. The comparision of Channel Islands is with other states. Although, Worle, Denver etc are at city level.

Another major finding was that the maximum standard deviation among the salaries based on Job titles was found in 'Reconciliations Accountant' with the deviation being 85,920. The maximum salary for this job title is 199,860 and minimum is 27,840. Although, the person who earns the most in UK has a salary of 200,000 and works in an Accounting & Finance Job.

This dataset also tells if people work on contract or are permanent. The initial instinct is that the people who work on contract would be paid more because they do not have any job security and in the USA the freelancers get paid higher than permanent workers. Surprisingly, there is not much difference in the income of the people on contracts and permanent employee in UK. People on contract earn about 36,099.67 where as permanent employee get around 35,327.47 which is just a difference of 700.



Map1*: Cities with highest and lowest average salaries*

Map 1 shows the plot of the cities with highest and lowest average salary. The blue plots are highest salary and the red are lowest average salary at city level. Besides each plot is the average salary mentioned. There is no clear trend based on the geographical locations for the salaries earned. However, London and the South East have a higher income at average.

2 THE PREDICTIVE TASK

Given the data, we can predict the salary of the job. Using this knowledge we can create many scenarios like what will be the salary of the job or what should be the salary of the position (can be used by the company), is my salary enough (the employee can think in this term)

For this task I plan to use the absolute error and mean square error. Using error metric like precision and recall do not make sense as I am not predicting binary values or predicting classes. I predict an absolute value and the best way to find the error is by finding how much I missed the actual value by.

I am using the baseline model as predicting the average for all jobs. Here average is the average of the salaries in the whole training data. I am considering this as the baseline because there this is a model which would give the average prediction rate. Anything below this is a useless model and probably the combination of features used should not be experimented further. Based on the knowledge of jobs and salary provided I focused my model on three features. First, if the employee is permanent or on contract; second, the location of the job; third, the category of the job. I also tried experimenting with the Job titles but they were very diverse and proved to be useless as they just increased the computation time.

Below is the description of all the models I tried and how I made them to work.

Model 0 or baseline:

It just predicts the average salary of all employees in the training data. There is no bias on any feature.

Model 1:

In this model I tried to embed the Location, name of the company and the title of the job. I used 'pandas' for my code. This made my reads from file and computation very fast. Pandas uses 'C' to do do the computations while uses python just as a wrapper. For this model I group the training data by these three features. Then for each test entry I check if any of the three features present in the train data. For all those train entries that have the same location or company or title, I take the average of them and predict this as the salary of the test entry. If it fails to find anything then I predict the global average as in the model 0.

Model 2:

For this model I used the category of the job which is not that broad as shown in the data description and the company name. I used the groupby feature in pandas to create a subset of data where each row represents a unique combination of the category and company with the average salaries for those combinations. Now I checked that for each test item if all three of these category exist together in the grouped data. If it does I predict the training average of that combination as the answer or else I just predict the global average as in model 0. I also started parallelizing code. I used the multiprocessing library by python to run my code on multiple threads.

Model 2.1:

This was just a modification on Model 2 where I also included a third feature, the Location of the job. Everything else remained the same.

Model 3:

I use different regression techniques in this method. The features I choose where the company name, location and the category this job belongs too. As string cannot be passed in regression and converting them into just numbers would not make sense as regression would assume that one element is greater than another. For each of these three features I used the LabelBinarizer provided by sklearn to convert them into binary. Now each row of training data was about 28,000 features long. This was because there are about 25 categories, hence 25 features. About 10,000 locations hence another 10,000 features and so on. One thing to notice was that it will be impossible to store a matrix of size 28,000 x 1.8 Million. I used the sparse matrix to save space. I used the Coordinate style sparse matric which is very efficient with speed. For rows where a feature was missing I filled it with 0 or in other words it does not exist scenario. Once my features were ready I tried different regression models like Linear Regression, SGDRegressor and RandomForestRegressor. Each gave a different result and the best was chosen. SGDRegressor performed better than all. RandomForestRegressor performed better than Linear Regression.

Model 4:

Machine learning task were not giving good results and took a lot of time; therefore I started to experiment without Machine learning algorithms. In this model I find the average for each category of jobs and if the test category is same I just predict this average. Test data contains the same categories as train data.

In this model I also found out that the salaries given in the data are not normalized properly. There are values like 5000,5001,5008,5098 and so on. I rounded them up to last 2 and 3 digits. For example I converted 5001 to 5000, 5055 to 5100 and 50101 to 50100, 50155 to 50200. I applied this rounding model to all the models I tried before and best results were noted.

3 THE LITERATURE

I found this dataset on Kaggle. (https://www.kaggle.com/c/job-salary-prediction) This competition is presented by Adzuna. This data set is very similar to census data. There is census data available for each country. There are other competitions on Kaggle (https://www.kaggle.com/c/us-census-challenge/leaderboard) as well as other places like KDD (https://archive.ics.uci.edu/ml/datasets/Census-Income+(KDD)). However, most of these competitions or dataset requires to predict whether a person's income is above or below a certain threshold which is much simpler than the problem I am trying to solve, i.e. predict the exact salary of the person.

Most of the competitions that require just a binary answer i.e. if the person earns above or below a certain threshold (50,000 for USA) the best or recommended algorithm is randomized forests. Neural networks also perform well on this kind of data. However, the problem in the dataset I choose is that we do not predict binary. If we use similar algorithms we would end up with more than 10,000 labels which would become very costly and inefficient at times. Hence, the best algorithm is to create your own model or use regression. Regression does not predict a label but values that could be multiplied with features to give an approximate value close to the true label. The benchmark provided in the competition uses global average value as shown in Model 0. They also show the use of random Forest where the features are extracted from the description using bag of words. This model is useless as doing bag of words produces redundant data that is already provided in other columns and using bag of words induces some noise. Here noise is refers to words that are not important and may come once or twice only in the whole dataset.

The state of the art algorithm is 'Deep Learning' also known as Deep Belief Network. This algorithm is a graphical model which tries to extract a deep hierarchical representation of data. It works on the principle of CNN and gives amazing accuracy but is very slow to train.

My method is very similar to what other methods are trying to implement. They are using the same features (Location, Company, Contract type, Category) in some combination to get a good result. Based on the benchmark provided and my approximate score on the leaderboard I feel that I my results are consistent to what others are doing.

4 THE RESULTS

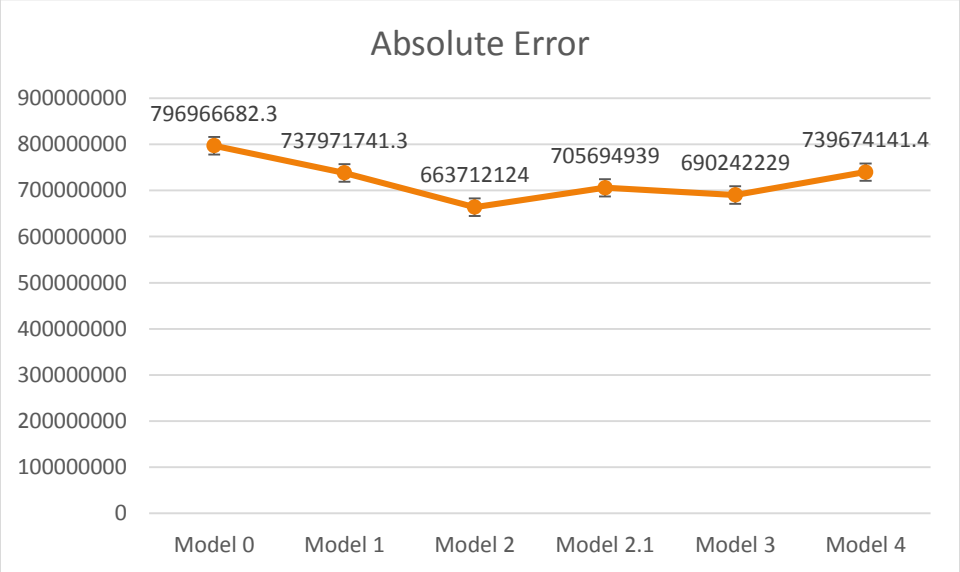
Not all models performed well. However, each model was better than the baseline. It is because baseline does not take into consideration the diversity of the salaries across the features and just predicts an average value. Table 1 sows the Errors each model got.

	Absolute Error	Mean Absolute Error	Mean Square Error
Model 0	796966682.3	4341.344633	321230055.7
Model 1	737971741.3	4019.979416	261936172.6
Model 2	663712124	3615.462392	230256255.3
Model 2.1	705694939	3844.156856	251742591.6
Model 3	690242229	3759.980766	237854119.1
Model 4	739674141.4	4029.25296	269918358.5

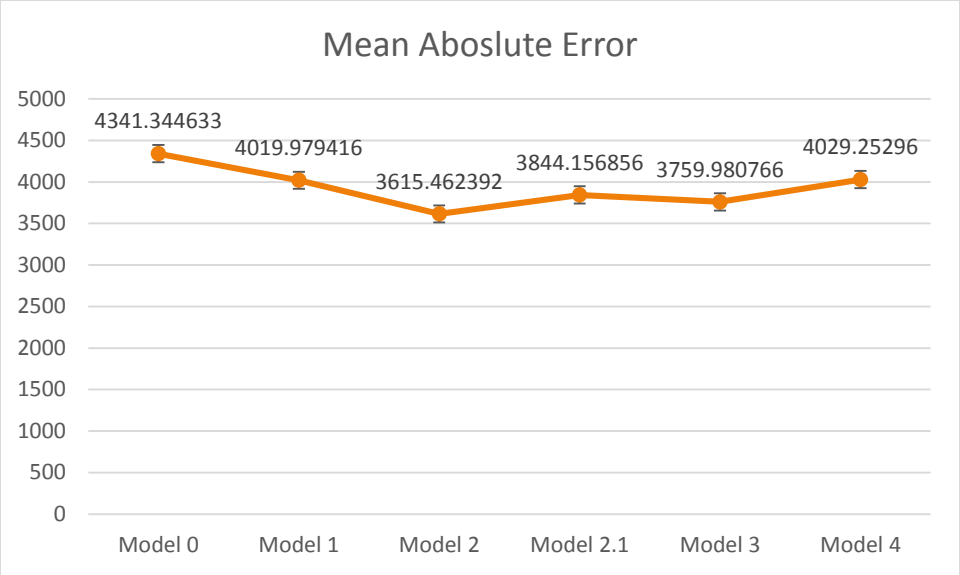
Table 1: *Results for each model*

Model 2 uses only Category of the Job and Company name to create a model. All the models are basically trying to cluster based on different models and in slightly different ways. Model 2 is very simple and does not complicate features. It uses the column that contain the maximum number jobs per feature. There are only 20-25 Categories so the average found is over a larger scale. Model 2 uses Company Name and Category of the job. This gives a good result. Over complicating reduces the cluster size and increases the number of clusters formed and hence increase the error. This can be verified by model 2.1 where adding another feature increases the error.

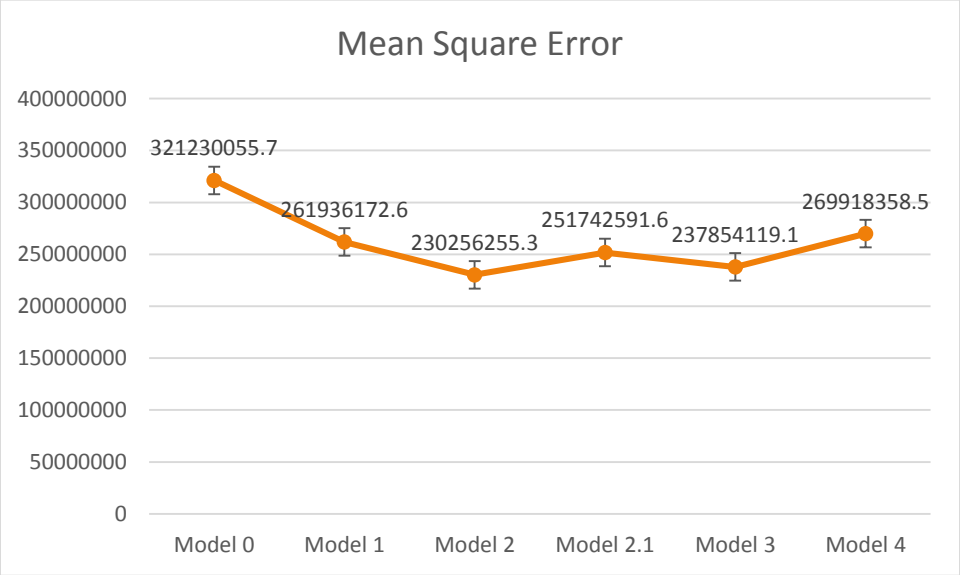
Regression i.e. model 3 performed similar to the best model but the time to create features is insane and not feasible to try many options in the given time and resources. I could have reduced the data but then the diversity of the data drops significantly and there is a possibility to have a subset where some categories will give biased average due to the less number of jobs and the unique findings for the data set i.e. stats like max salary, min salary etc. would make less sense and would be less realistic.



Graph 1: Change in Absolute Error



Graph 2: Change in Mean Absolute Error



Graph 3: Change in Mean Square Error

Based on the results it seems to be like that the Company, Location and the Type of job help determine the salary. However, the contract type does not effect when the salary is taken in consideration per annum.

In the Kaggle competition they use Mean Absolute error to determine the leader. I have also calculated that. According to my score I should be second on the leaderboard. However, claiming that is not completely valid. I have create my own test data as mentioned which is 25% of the training data only which I separated at the starting. This competition does not allow submissions now nor does it give out the true labels of the test data; therefore I cannot compare my error rate with the leader board.