Team-DS

Project Report

Predict job success based on Student's credentials.

Machine Learning
Spring 2016

Team Members:

Solmaz Shariat Torbaghan
Himaja Rachakonda
Prithvi Krishna Gattam

Advisor:
Dr. Kush R. Varshney

1. Introduction

In this project we were interested to see what factors can predict success after graduating from college. To address this question we look at a dataset of engineers in India that includes information about their credentials, high school grades, college attended, BS GPA, English proficiency, AMCAT score (a standardized test in India to measure students skills), personal skills (derived from a personality test) , city they went to college and so on. To narrow down our question, we look at starting salary.

Our analysis depicts how these variables interact with salary, and essentially predict the salary for new candidates knowing their credentials. Using our models on the training data we are going to predict dependent variables on the test data. Later we look at our solution to extract features with highest effect on predicting salary. This analysis could be used to give recommendation system on the basis of our data to see what factors are most important to affect one's salary.

In order to answer these questions, we used a couple of regression models to predict exact values for our dependent variables (salary). We thought it's more meaningful to divide the salary range into classes; and so we also used classification models to predict which range a new candidate will most likely end up.

2. Methods and Data

We used the dataset from the data challenge: "Understanding the labor market", could be found here: http://ikdd.acm.org/Site/CoDS2016/datachallenge.html
Detailed information of data is provided under the above link but here we give a summary of datasets and how we cleaned up the data.
The data includes two datasets:

- Training data
- Test data

Each row of these datasets carries information about one candidate's credentials. There are 32 features (independent variables) and 5 dependent variables from which we are going to predict 1 (salary). Training data has 4000 rows (data points) and test data has 1000 rows. The 5 dependent variables columns are empty in the test data to be predicted.
For our analysis, we divided our training data to two sets: training and validation datasets.

3. Data Cleaning

This dataset needed some cleanings and modification. Besides some feature representation should be done.

- Some of the features in the dataset are self-reported and they are not the same across subjects. For example, city could be either "Banglore" or "Bangalor" or "Banglor". These discrepancies could be fixed manually.
- In case of categorical data, we used one-hot-encoding.

- In case of missing values, values were imputed by mean and variance.
- Finally data is standardized to have zero median and unit interquartile range.

Here is the complete list of features and brief description on how we modified them:

- Gender: One hot encode it into two columns
- DOB: just retained the year.
- 10board: The most frequent boards are “cbse”, “state”, “icse” and “n/a”. We reduced each to one of (cbse, state, n/a, icse) and then one-hot-encoded them.
- 12board: The same strategy as 10board.
- collegeID: This is the unique college ID, excluded from our analysis.
- Degree: all the degrees are one of 'B.Tech/B.E.' 'MCA' 'M.Tech./M.E.' 'M.Sc. (Tech.) We used one-hot-encoding for this feature as well.
- I've added some custom columns using the method
- Specialization: We used addCustomSpecialisationColumns. What it does is that it looks at all the unique words in this feature across all columns. Then selects only those that appear 10 times or more. For each row one hot encoded the words presence by setting the corresponding column to one or zero.
- CollegeCityId: same is collegeID
- CollegeState: one-hot-encoding, the original data is pretty clean.
- For the following features if the value in the row was -1 replaced it with a zero, else retained the value: Domain, Computerprogramming, ElectronicsAndSemicon, ComputerScience, MechanicalEng, ElectricalEng,TelecomEng, CivilEng
- The rest of the features are retained as in the original datasets.

Two new feature were generated using the existing features:

- Gradage: candidates age when graduated from college (graduaduation year-Birth year)
- 12age: candidate’s age when finished 12 year of standard schooling (12graduation – birth year)

Overall we came up with 84 features after cleaning and feature representation.

4. Baseline Algorithm

We have implemented the basic ridge regression (with regularization = 1) algorithm to determine the salaries of the subjects. We have implemented this considering only the numeric data from the training set. This resulted in using 25 out of all 32 features.

We divided the training set (of 4000 examples) into 2 parts for training and test (of 3000 and 1000 examples respectively) to evaluate baseline performance.

We calculated the R-Square, Mean Absolute Error (MAE), Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE) on this baseline, to evaluate the regression statistics. The following are some values that we observed –

MAPE	35.20%
r-Square	0.15
MSE	24531350415.6
MAE	102625.46

The baseline evaluation measures indicate that the Mean Squared Error and Mean Absolute error is very high due to the samples in the higher salary ranges. This indicates that the differences in the salary prediction in the lower ranges of salaries are given equal weightage to the differences in the higher ranges. The r-square metric also resulted in a very low coefficient of determination indicating a poor fit of the data to this regression. R-square, MSE, MAE measure are affected heavily by the outliers in the data which is not expected for this problem. The percentage error resulted in a decent output amongst all showing a 35% error.

5. Evaluation Techniques

We tried to look at the distribution of the salaries to make a decision on the evaluation criteria after observing the baseline results. The distribution seemed to be fitting a log-normal distribution i.e. log of the response variables follows a normal distribution, which is a heavy-tailed distribution (Fig.1).

We observed that using r-square/MSE/MAE is not an appropriate method for this problem, as we are more concerned about the differences in the lower ranges of salaries when compared to the differences in higher range of salaries. These methods would penalize high and low incomes the same, which was not what we wanted. i.e. 100$ of difference when salaries are on the range of thousands is not the same as 100$ of difference in salaries when they are in the range of millions. So we had to come up with a method that distinguished these factors.

Some of the approaches we considered for the evaluation criteria are as follows -

- One approach to evaluate our model is to perform a MSE on log difference of predicted and actual values.
- Mean Absolute Percentage Error to evaluate the predictions. This is chosen to accommodate the scale of different salary ranges across people. The model performance is compared between Lasso, Ridge and Median of Means Estimator.
- For our classification model we used Mean-Absolute-Error to evaluate our predictions.

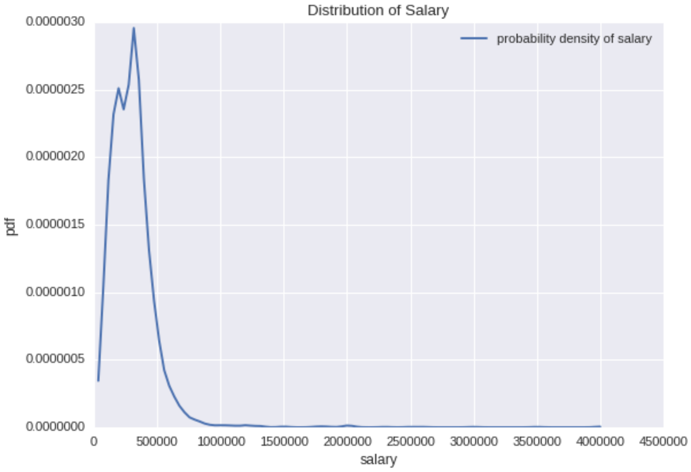


Fig.1: Distribution of salary in training data.

6. Models

a. Regression with Heavy Tailed Distributions:

We have aproached this problem by observing various forms of loss functions of the least squares linear regression –

$$l\big((x, y), w\big) \coloneqq \frac{1}{2}(x^T w - y)^2$$

Many standard methods for estimation and statistical learning are designed for optimal behavior in expectation, yet they may be suboptimal for high-probability guarantees. For instance, the population mean of a random variable can be estimated by the empirical mean, which is minimax optimal with respect to the expected squared error. However, the deviations of this estimator from the true mean may be large with constant probability. In many practical applications, distributions are heavy-tailed and thus are not sub-Gaussian. Thus, standard techniques such as empirical averages may be inappropriate, despite of their optimality guarantees under restrictive assumptions.

Our estimation technique is based on Median of Means Estimator. The basic idea is to repeat an estimate several times by splitting the sample into several groups, and then selecting a single estimator out of the resulting list of candidates with an appropriate criterion (Fig.2 and Fig.3).

Preliminaries:

Assume an example space Z, and a distribution D over the space. Further assume a space of predictors or estimators X. We consider learning or estimation algorithms that accept as input an i.i.d. sample of size n drawn from D and a confidence parameter δ ∈ (0, 1), and return an estimator (or predictor) $(\hat{\beta}) \in$ X. For a (pseudo) metric ρ on X, let $B_\rho \in (\beta_0, r) \coloneqq \{\ \beta\ \in X :$ $\rho(\beta_0, \beta) \leq r)$ denote the ball of radius r around w_0

We assume a loss function $l\colon Z\ x\ R^+$ that assigns a non-negative number to a pair of an example from Z and a predictor from X, and consider the task of finding a predictor that has a small loss in expectation over the distribution of data points, based on an input sample of n examples drawn independently from D. The expected loss of a predictor w on the distribution is denoted $L(\beta) = E_{Z\sim D} l(Z, \beta)$. Let $L_* \coloneqq inf_\beta L(\beta)$. Our goal is to find w such that $L(\hat{\beta})$ is close to L_*.

Warm-Up: Robust Distance Approximation

Given the estimation problem, the goal is to estimate an unknown parameter of the distribution, using a random i.i.d. sample from that distribution. Median of Means estimator for Linear Regression in heavy tail distributions shows that if the sample is split into non-overlapping subsamples, and estimators are obtained independently from each subsample, then with high probability, this generates a set of estimators such that some fraction of them are close, under a meaningful metric, to the true, unknown value of the estimated parameter. Importantly, this can be guaranteed in many cases even under under heavy-tailed distributions. Having obtained a set of estimators, a fraction of which are close to the estimated parameter, the goal is now to find a single good estimator based on this set.

This goal is captured by the following general problem, which we term Robust Distance Approximation. A Robust Distance Approximation procedure is given a set of points in a metric space and returns a single point from the space. This single point should satisfy the following condition: If there is an element in the metric space that a certain fraction of the points in the set are close to, then the output point should also be close to the same element.

Warm-Up: Median of Means Estimator

Median of Means estimator first partitions a sample into k equal-size groups, and returns the median of the sample means of each group. It is well known that the median-of-means achieves estimation with exponential concentration.

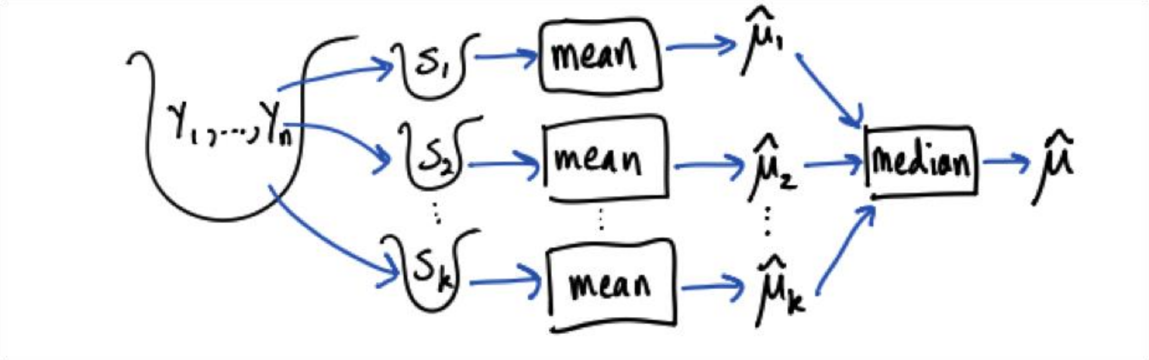
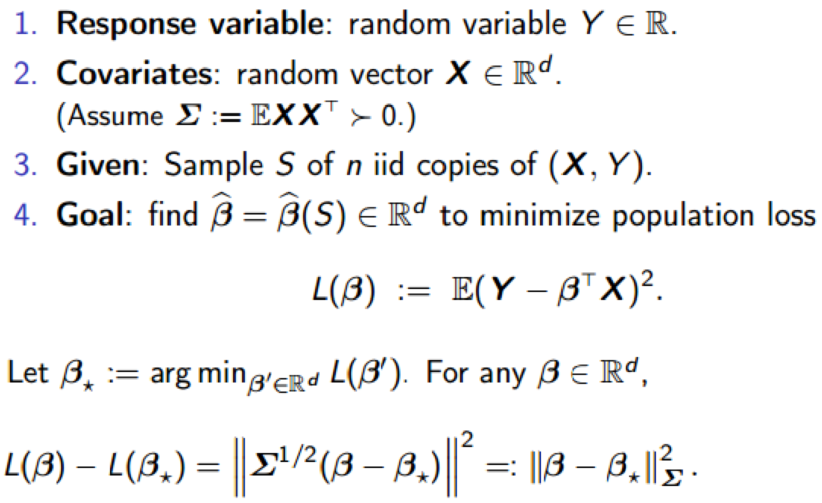


Fig. 2) Sketch of Median of Mean Estimator [Nemirovsky and Yudin, 1983; Alon, Matias, and Szegedy, JCSS 1999].

Linear Regression with Median of Means Estimator

Overview of the Approach:



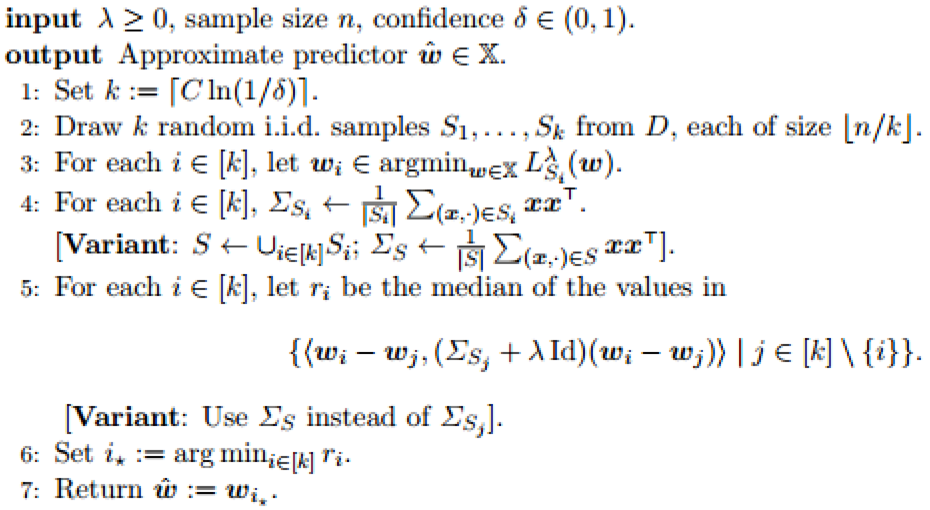
1. **Response variable**: random variable $Y \in \mathbb{R}$.
2. **Covariates**: random vector $\boldsymbol{X} \in \mathbb{R}^d$.
   (Assume $\boldsymbol{\Sigma} := \mathbb{E}\boldsymbol{X}\boldsymbol{X}^\top \succ 0$.)
3. **Given**: Sample S of n iid copies of $(\boldsymbol{X}, Y)$.
4. **Goal**: find $\widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}(S) \in \mathbb{R}^d$ to minimize population loss

$$L(\boldsymbol{\beta}) \; := \; \mathbb{E}(Y - \boldsymbol{\beta}^\top \boldsymbol{X})^2.$$

Let $\boldsymbol{\beta}_\star := \arg\min_{\boldsymbol{\beta}' \in \mathbb{R}^d} L(\boldsymbol{\beta}')$. For any $\boldsymbol{\beta} \in \mathbb{R}^d$,

$$L(\boldsymbol{\beta}) - L(\boldsymbol{\beta}_\star) = \left\| \boldsymbol{\Sigma}^{1/2}(\boldsymbol{\beta} - \boldsymbol{\beta}_\star) \right\|^2 =: \|\boldsymbol{\beta} - \boldsymbol{\beta}_\star\|_{\boldsymbol{\Sigma}}^2 .$$

The proposed algorithm for regression (Algorithm 1) is as follows.
Set $k = C\, log\left(\frac{1}{\delta}\right)$, where C is a universal constant is. First, draw k independent random samples i.i.d. from D, and perform linear regression with λ-regularization on each sample separately to obtain k linear regressors. Then, use the same k samples to generate k estimates of the covariance matrix of the marginal of D on the data space. Finally, use the estimated covariances to select a single regressor from among the k at hand.



```
input $\lambda \geq 0$, sample size $n$, confidence $\delta \in (0,1)$.
output Approximate predictor $\hat{\boldsymbol{w}} \in \mathbb{X}$.
Set $k := \lceil C \ln(1/\delta) \rceil$.
Draw $k$ random i.i.d. samples $S_1, \ldots, S_k$ from $D$, each of size $\lfloor n/k \rfloor$.
For each $i \in [k]$, let $\boldsymbol{w}_i \in \mathrm{argmin}_{\boldsymbol{w} \in \mathbb{X}} L^{\lambda}_{S_i}(\boldsymbol{w})$.
For each $i \in [k]$, $\Sigma_{S_i} \leftarrow \frac{1}{|S_i|} \sum_{(\boldsymbol{x},\cdot) \in S_i} \boldsymbol{x}\boldsymbol{x}^\top$.
  [Variant: $S \leftarrow \cup_{i \in [k]} S_i$; $\Sigma_S \leftarrow \frac{1}{|S|} \sum_{(\boldsymbol{x},\cdot) \in S} \boldsymbol{x}\boldsymbol{x}^\top$].
For each $i \in [k]$, let $r_i$ be the median of the values in
      $\{\langle \boldsymbol{w}_i - \boldsymbol{w}_j, (\Sigma_{S_j} + \lambda \mathrm{Id})(\boldsymbol{w}_i - \boldsymbol{w}_j) \rangle \mid j \in [k] \setminus \{i\}\}$.
  [Variant: Use $\Sigma_S$ instead of $\Sigma_{S_j}$].
Set $i_\star := \arg\min_{i \in [k]} r_i$.
Return $\hat{\boldsymbol{w}} := \boldsymbol{w}_{i_\star}$.
```

b. Classification Problem: Classify Students to Different Salary Ranges

The salary across the dataset was divided into 5 bins (based on the 20th quantlile of the examples) such that each bin has a uniform number of data points. The bins are labeled 1,2,3,4, and 5. Unlike a regular classification setting where there is no ordering amongst the labels, here there is an ordering amongst the bins. The average salary in bin 1 is lower than 2 and so on. Consequently predicting a salary bin of 2 on an actual value of 1 is better than predicting 3 or 4.

Two obvious approaches for handling discrete ordinal labels are -

1. Treating the different rating levels as unrelated classes and learning to predict them as in a multiclass classification setting, and
2. Treating them as a real-valued responses and using a standard regression setting with a loss function such as sum-squared error.

However, neither of these reflects the specific structure of discrete ordinal labels.

Common choices for the loss functions are the logistic loss (as in logistic regression), and the hinge loss (distance from the classification margin) used in Support Vector Machines. We used the *logistic ordinal regression* model, also known as the proportional odds. We used the logistic loss function along with two threshold based approaches, immediate-threshold and all thresholds.

In immediate-threshold we consider, for each labeled example(x, y), only the two thresholds defining the "correct" segment $(\theta_{y-1,}\theta_y)$. This segment contains the correct label y, and penalized violations of these thresholds (eq. 1) where $z = z(x)$ is the predictor output for the example. Here f is an upper bound on the zero one loss, the logistic loss. The immediate-threshold loss is ignorant of whether multiple thresholds are crossed.

Eq.1) $loss(z; y) = f(z - \theta_{y-1}) + f(\theta_y - z)$

We use Mean Absolute Error(MAE) for ordinal regression, which counts the sum of distances between the true and predicted labels.

Immediate-threshold bounds zero-one error, but not (necessarily) mean absolute error. All threshold bounds this absolute error. The all-threshold loss is a sum of all threshold violation penalties. If the binary loss function bounds zero-one error, then the all-threshold loss bounds mean absolute error.

Using the definition of the function $s(l; y)$ as (eq.2) the all threshold loss $loss(z; y)$ is eq.3 where f is the logistic loss function.

Eq. 2) $s(l; y) = \begin{cases} -1, & if\ l < y \\ +1, & if\ l \geq y \end{cases}$

Eq. 3) $loss(z; y) = \sum_{l=1}^{T-1} f(s(l; y)(\theta_l - z))$

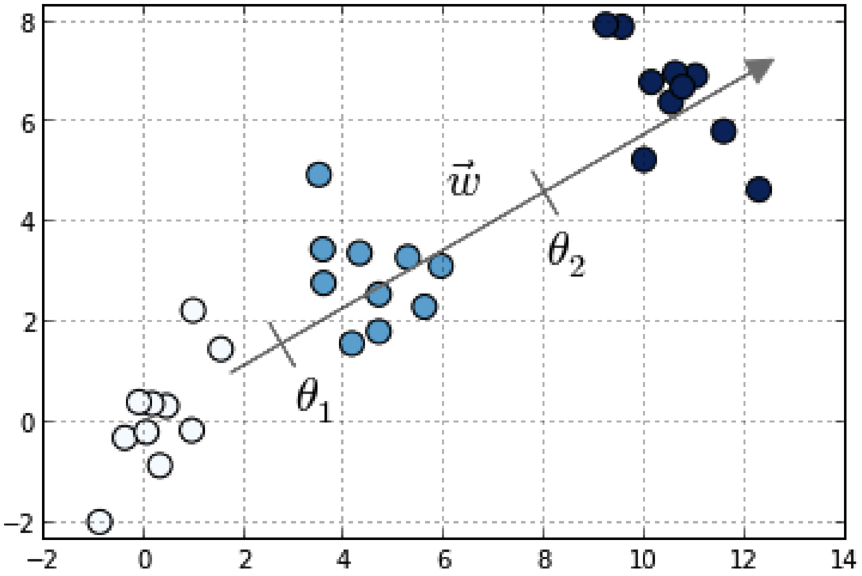


Fig.4) Logistic Ordinal Regression.

7. Results

All the models are implemented over a pipeline of cross-validation of 10 folds. The pipeline included steps of data preprocessing and model evaluation to achieve the best hyper-parameters and scores for the data.

Following Regression models have been implemented to estimate the salary of a student

- Median of Means Estimator for Linear Regression (MOM)
  - With response variable – Salary (following a log – normal distribution)
  - With response variable – log(Salary) (following a normal distribution)
- Ridge Regression
- Lasso Regression

Our experiments with median of means estimator have proven to be successful giving the least error rate (23% test error) when compared to the other Regression models like Lasso and Ridge. Following is the graph showing the train and test errors of the model.

Model	Train MAPE Error	Test MAPE Error
Median of Means	22.12 %	23.16 %
Median of Means(log salary)	28.23 %	30.73 %
Ridge Regression	31.86 %	34.27 %
Lasso Regression	40.23 %	42.83 %

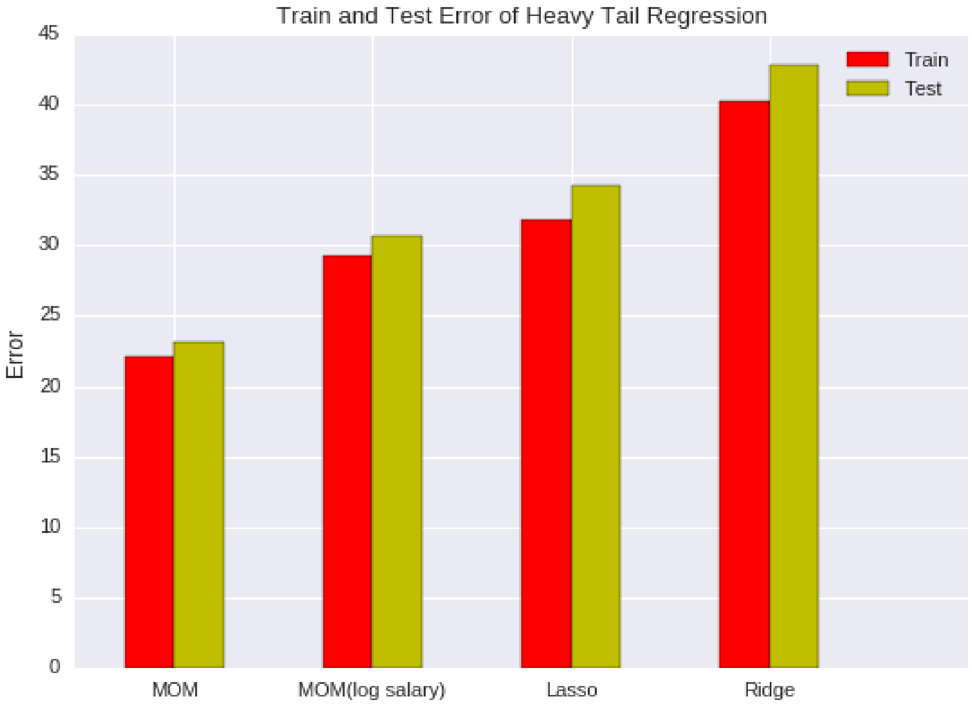


Fig.5 shows that Median of Mean has the lowest error comparing to Lasso and Ridge regressions

Following Classification models were implemented to classify the student into a Salary range –

- Logistic All Threshold
- Logistic Immediate threshold
- Logistic Squared Error

With all three with cross validation we could find a best mean absolute error of 0.97. The Mean absolute error of the ordinal regression algorithms is compared across the three methods.

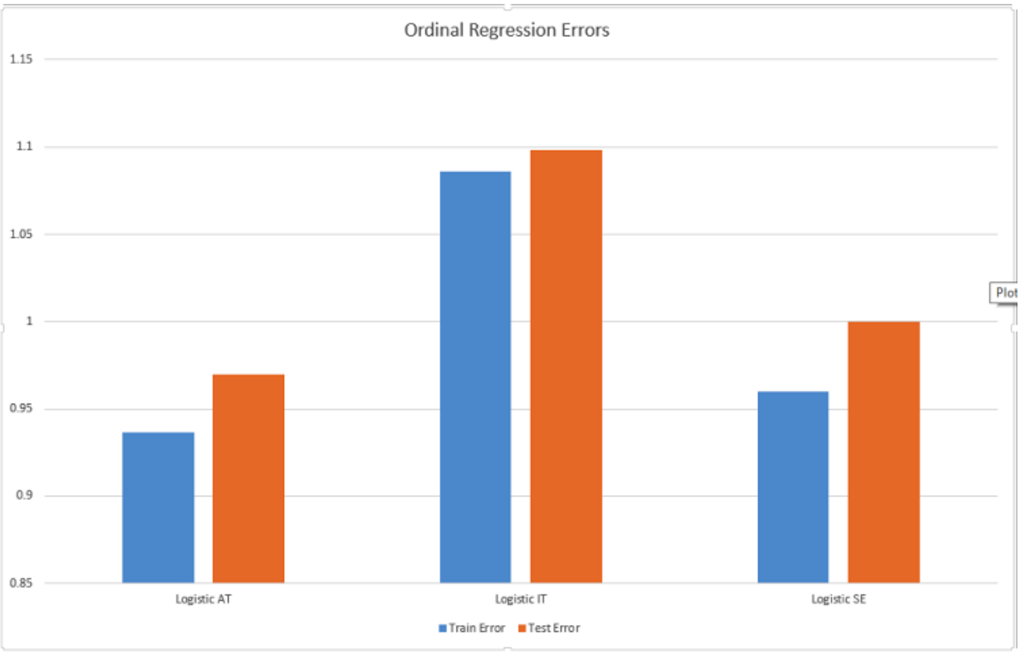


Fig.6. As it is shown Logistic all threshold has lower error comparing to the other two.

8. Analysis of Results

The interesting question in this study is to find features with higher effect on job success. In order to find these features the following has been performed in our best solution of ordinal logistic regression using All-threshold and Median of Means Estimator:

We looked at the coefficients of the solution and put all coefficients greater than threshold $t1$ in tier1 and those less than $t2$ and greater than $t1$ in tier2. Within each tier we then divided them into positive and negative coefficients. We ranked the factors for which we got the highest coefficients to check which factors affect salary prediction the most.

Following is the ranking of top factors from regression problem –

column	weight
Logical	0.001006
10percentage	0.000824
12graduation	0.000685
icse.1	0.000661
icse	0.000649
CollegeTier	0.000597
CollegeCityTier	0.000497
English	0.000491

state.1	0.000455
Quant	0.000427

Interpretation from the Results:

- Logical Scores in the AMCAT tests has the highest ranking showing that the aptitude levels play an important role.
- Followed by 10th percentage results
- ICSE students and English speaking skills follow the list showing the soft skills also are important factors.
- College City and Tier indicate that the background of the student is also an important factor

Following is the list of top negatively ranked coefficients –

Column	Weight
Meghalaya	-0.0001
science	-0.00012
extraversion	-0.00013
Punjab	-0.00013
CivilEngg	-0.00014
biotechnology	-0.00014

Interpretation of Negative coefficients –

- Students from States like Meghalaya, Punjab in the contribute less towards Salary Prediction.
- Branches like Civil and BioTechnology contribute less towards Salary prediction

Following is the top ranking from Classification Problem –

column	weight
12percentage	0.218241
collegeGPA	0.188449
Quant	0.177718
10percentage	0.160634
English	0.130011
computer	0.118656
CivilEngg	0.116987
Karnataka	0.112296

Our Interpretation from the results:

Interpretation of Positive Scores –

- The highest coefficients here are the collegeGPA, Quant, and 12percentage. This indicates that having a higher collegeGPA, quantitative score and 12th standard scores contribute to a higher salary.
- 10th standard scores are also contributing, but not as much as 12th.
- English scores also seem to be important.
- The field in our data which indicates whether or not the specialization contains the word computer, seems to have relevance.
- This indicates that students with a background in computer science are paid more.
- The coefficient of Karnataka also indicates that if the college state is karnataka, the data suggests that the pay is more.

Interpretation of Negative Scores –

Column	Weight
application	-0.133570527
myDOB	-0.14173511
12graduation	-0.167365948
ComputerScience	-0.102521839

- If the word "application" is present in the specialization it seems to have a negative contribution to the salary.
- DOB and 12Graduation both have high negative coefficients. These columns in the dataset are the years in which the event occurs, indicating that there is a trend of older candidates being paid more.
- Computer Science, AMCAT scores are surprisingly having a negative coefficient. This is strange. Maybe this is a property of the dataset. We even tried splitting the data randomly with different splits and still this field is negative.

Tier2 Interpretation –

- All the college states have a coefficient less than 0.5 suggesting that they're not important as karnataka, when it comes to this dataset.
- Presence of the word "information" in the specialization seems to contribute a little more as compared to control/enginnering/technology/instrumentation, which coincides with the earlier observation of computer being present in tier1.

Inference from Both Classification and Regression –

Both the Classification and Regression have given useful insights into the factors affecting the salaries of the students in India. Some of the overlapping results from both the problems indicate that the English speaking skills, Quant skills, College State and IT roles predominantly affect the salary.

9. Conclusion

In this project we looked at a dataset of engineering candidates in India. Each individuals' credentials along with job salary is provided in the dataset. To predict salary with the original features in the files using ridge regression we got a very bad prediction (Mean Percentage Error of: ~35%). After cleaning up the data and feature engineering, we ended up having 84 features.

Salary has a heavy-tail distribution, so we had to use other evaluation techniques than MSE/R-square, such as Mean Absolute Percentage Error for regression and Mean absolute error for classification.

First we used regression to predict salary for each individual. Out of the 4 regression models used here, Median of Means Estimate seemed to have a better prediction. Later we used logistic ordinal regression and classified salaries to different ranges. Across the three different models we used here, logistic all threshold has the best prediction for salary.

Finally, we looked at our solution to give recommendation as what features are best predictive of higher salary. We observed that English skills, Quant skills, College State and IT skills predominantly affect the salary.

10. References

- http://jmlr.org/proceedings/papers/v32/hsu14.pdf
- Fabian Pedregosa-Izquierdo. Feature extraction and supervised learning on fMRI : from practice to theory. Medical Imaging. Université Pierre et Marie Curie - Paris VI, 2015. English.
- J. D. M. Rennie and N. Srebro, “Loss Functions for Preference Levels: Regression with Discrete Ordered Labels,” in Proceedings of the IJCAI Multidisciplinary Workshop on Advances in Preference Handling, 2005.
- http://www.cs.columbia.edu/~djhsu/papers/heavytails-slides.pdf