2017

Prediction of Employee Attrition in Companies

APPLIED MACHINE LEARNING

SIDDHARTH NAIR

Introduction

Today, in the age of "Big Data", companies around the world are collecting as much data as possible about their customers and performing all kinds of analysis on them using concepts of Machine Learning. These companies, for a long time, have also had large internal datasets which contain records of their employees. These datasets contain a wide variety of detailed information about their employees, including their pay scale, their employment level, working hours as well as details from their internal survey. This data is extremely valuable, to judge employees general sense of happiness, satisfaction with the company and the work they do.

The rest of this paper is organized as follows. Background, Related Works are covered in the next section. The Methodology section outlines the collection of the dataset, preparation, baseline performance, error analysis and the remaining methodology for machine learning. This is followed by an analysis of my Results and culminates with a discussion of my findings in the Conclusion.

Background and Motivation

Attrition has always been expensive for companies. If there was a robust way to predict employee dissatisfaction, companies could predict employees who would or are planning to leave. This would be very valuable for companies, as it could inform companies hiring policies, and help them plan for the future as to deciding, how many employees to hire, the vacancies that need to be filled in, project continuity and knowledge transfer tasks and how much should their financial outlay be for new hires depending on the number of people to replace. Hence, this prediction of employees who are possibly looking to leave the company, could be extremely valuable.

The work outlined in this paper looks at this problem. I am trying to address this research question by trying to determine the factors, which could be relevant to determining what causes an employee to be disgruntled with their work environment and eventually leave.

Related Works

Msachine learning is being increasingly used in companies to deal with the problem of employee attrition. In the work by Punnoose et al. [1] they looked at employee turnover in an organization by examining the dataset from HR Information Systems(HRSIS) of a global retailer company. They have treated the problem as an attrition problem, and have shown the superior performance of Extreme Gradient Boosting (XGBoost) technique as compared to other standard classifier techniques. In the work by Hong et al., they conducted a comparison of employer turnover by applying the Logistic Regression(logit) and Probability Regression(probit) prediction models. In the work by Sikaroudi et al. [2] they looked at turnover pattern among employees by comparing various knowledge based systems, that require minimum parameters for tuning, and hence are easier to implement. The factors they were trying to evaluate was the importance of user friendliness and time consumption, in evaluating such systems to address the attrition problem. Their results showed that SVM, PNN and KNN are sensitive to parameters. In contrast, Naive Bayes was the most user friendly model that had a good performance in classification. In the studies by Swider et al. [3]

and Cotton et al. [4], they showed how characteristics like age, tenure, pay, overall job satisfaction, employee's perceptions of fairness, working conditions, supervision, advancement, burnout etc. were some of the strongest predictors of employee turnover. This was very interesting to me, and had a direct influence on my project, as I had many of the factors that were listed as strong predictors. Also, these readings helped me a lot with my understanding of the dataset as well as during error analysis, while trying to look for possible combinations of predictors.

Methodology

1.Data Collection:

My data, the Human Resource Analytics dataset was sourced from the Kaggle website. This was a relatively new dataset in Kaggle, and I thought this would be an interesting problem to address.

- Fields in the dataset include:
- Last evaluation
- Employee Satisfaction Levels
- Number of projects
- Average monthly hours
- Time spent at the company
- Whether they have had a work accident
- Whether they have had a promotion in the last 5 years
- Department
- Salary
- Whether the employee has left

2.Data Preparation:

I divided the dataset into three sets: *development set,* for data exploration; the *train set;* and the *test set* as per the recommended 20:70:10 ratios.

To eliminate any possible skewness in the dataset, I started off by randomizing the dataset completely, before splitting it into the above three sections. To randomize it, I generated a column of random numbers, and then sorted my dataset, according to that column of random numbers.

After randomizing it, I divided the dataset into 2999 samples for the *development data (dev)*, 10500 samples for the *train/cross-validation set (cv)* and finally 1500 samples for the *test set.*

I then examined my dataset, to see the kind of values, I was dealing with. Since the ranges of my various attributes were so diverse, I standardized all my numeric attributes, so that the differing range of values in the attribute columns, won't affect the weight that is given to those features.

3.Baseline Performance:

Next, I performed a baseline performance analysis in LightSide, by using my *cross-validation(CV)* set, and testing various learning algorithms on it.

Before I started my experiment, after examining my dataset, I felt Logistic Regression would be the right algorithm for my problem because my class values were binary (0-Stay, 1-Left).

So using the default set of features I ran Logistic Regression with L2 Regularization on the CV set and I got the following results:

Metric	Value
Accuracy	0.7841
Kappa	0.2915

Next, I tested using Naïve Bayes. I was slightly reluctant using Naïve Bayes since Naïve Bayes proceeds with this inherent assumption, that the various features of the algorithm, are independent of each other. But I felt, this was not the case in my dataset, as I felt features like, *number of projects*, *average monthly hours* and *time spent at the company* were related, and couldn't be categorized as independent of each other.

After running Naïve Bayes on the CV, I got the following results:

Metric	Value
Accuracy	0.8366
Kappa	0.5594

So, the Accuracy as well as Kappa had improved significantly. But I was still hesitant to use this, because of the above mentioned reasons.

Next, I tried SVM, and obtained the following results:

Metric	Value
Accuracy	0.7747
Kappa	0.235

Its performance was worse than Logistic Regression.

Algorithm	Accuracy	Kappa
Logistic Regression	0.7841	0.2915
Naïve Bayes	0.8366	0.5594
SVM	0.7747	0.235

Keeping the above results in mind, I decided to choose Logistic Regression, since although Naïve Bayes performed better, I had my doubts regarding the feature independence. To get some reference I looked up the paper[5] regarding the difference between Generative Models (Naïve Bayes) and Discriminative Models(Logistic Regression) by Andrew Ng and Michael Jordan where they concluded that when the training size reaches infinity, the discriminative model(logistic regression) performs better than the

generative model(naïve bayes), although the generative model finds the solution comparatively faster. Also, since Naïve Bayes performance was high to begin with, I felt choosing Logistic Regression over Naïve Bayes, would give me some room to work with and learn about improving my model's performance. This also aligned with my original thinking of using Logistic Regression as the class value I was trying to predict was binary.

4.Error Analysis:

For this project, I did three rounds of error analysis on LightSide. Most of the features in the dataset are numeric features with a few nominal features.

So for the error analysis, I followed the methodology for non-text features.

1st Round of Error Analysis:

I loaded the standardized CV dataset, and chose the standardized column features, and extracted it. Using these extracted features, I build the model using Logistic Regression with an L2 regularization over 10-fold cross validation. I got the following performance metrics:

I got the following performance metrics:

Metric	Value
Accuracy	0.7839
Kappa	0.2909

Next, I used this trained model and tested it on my *dev* set and received the following performance metrics:

Metric	Value
Accuracy	0.7846
Kappa	0.2956

The confusion matrix output was:

Actual\Predicted	0	1
0	2129	139
1	507	224

On exploring the results from testing on the dev set, I chose the lower left cell, since it had the higher number of misclassified instances. After setting the frequency, horizontal absolute difference and the feature weight, and sorting it on decreasing order of the feature weights, I chose the value

of Normalized Satisfaction Value, since it showed the ideal combination of high absolute difference, feature weight and frequency.

I exported the prediction values based on this feature, and copied it to excel and compared against the other features, looking for some patterns or co-relations with other features. On analyzing the normalized satisfaction values, I found that the correctly classified values had values below -0.32709502. While the wrongly classified values had ranges up to +1.255694772. There were at least 195 instances out of the 507 misclassified instances, that showed this. So I decided to address this issue, by creating a new feature which would draw weight away from the sentiment values which were much higher than -0.327. I did this by introducing a new feature, which contained values which were a fraction of time the values which were greater than -0.327, thus reducing their overall weights, while it remained unchanged for the others.

After adding this new column feature, and extracting and building the model again on my CV set, I found a "Highly Significant Improvement" on comparing my original model and new model. My new model performance metrics were:

Metric	Value
Accuracy	0.7966
Kappa	0.3493

This is the result of my 1st round of error analysis on my dataset.

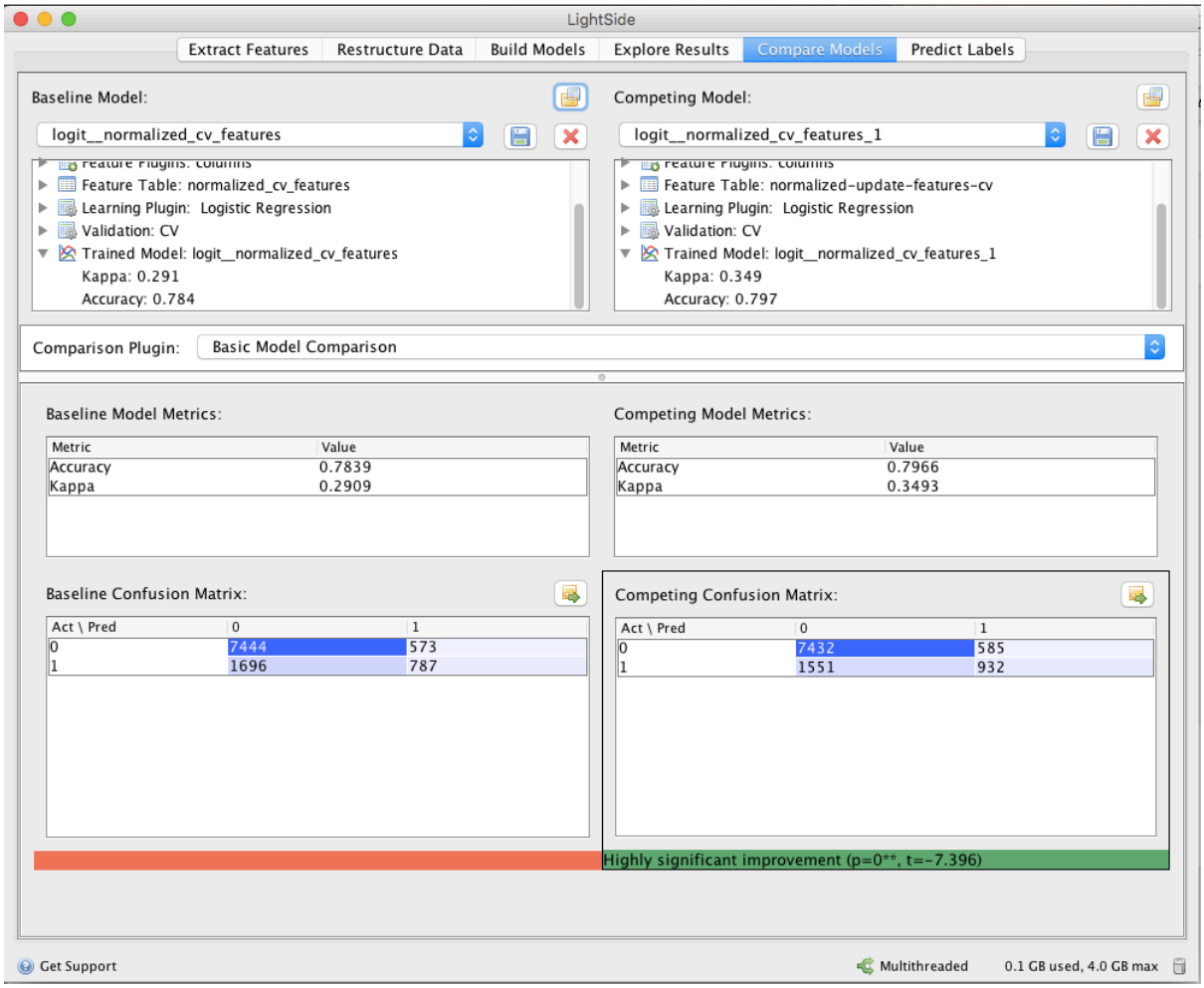

FIGURE 1. RESULT FROM 1ST ROUND OF ERROR ANALYSIS

2nd Round of Error Analysis:

I loaded the standardized CV dataset from my last iteration and build the model for which I got the following performance metrics:

Metric	Value
Accuracy	0.7966
Kappa	0.3493

Next, I used this trained model and tested it on my dev set and received the following performance metrics:

Metric	Value
Accuracy	0.7693
Kappa	0.3411

Confusion Matrix:

Actual\Predicted	0	1
0	1976	292
1	400	331

On exploring the results from testing on the dev set I obtained from last week's analysis, I chose the lower left cell, since it had the higher number of misclassified instances. After setting the frequency, horizontal absolute difference and the feature weight, and sorting it on decreasing order of the feature weights, I chose the value of Normalized Number of Projects, since it showed the ideal combination of high absolute difference, feature weight and frequency.

After exporting the prediction values based on this feature I analyzed the results.

On analyzing, the results between correctly predicted and wrongly predicted data, I found that for the wrongly predicted results, there were a number of cases where the number of projects were high but the number of monthly hours worked by the employee was less. I looked at this relation, because I thought these two could be a particular indication of an employee's dissatisfaction with their work environment, and a high project count, but low hours spend, could indicate they are not interested. This was contrasting with the correctly predicted results, were for employees who had more than 5 projects, showed significant monthly time investment. To resolve this issue, I added a new feature, that would reduce the weight the model, would give for employees who showed a low value for this combination of 2 features. I did this by introducing a new feature, which contained values which were a product of both the number of projects and the average number of hours per month, and I set a baseline at 1250 (5 projects x 250 hours per month), and for those employees who fell below this baseline, I set their value to be half of this product, while for others, it remained the same.

After adding this new column feature, and extracting and building the model again on my CV set, I found a "Highly Significant Improvement" on comparing my original model and new model. My new model performance metrics were:

Metric	Value
Accuracy	0.8616
Kappa	0.5863

This is the result of my 2nd round of error analysis on my dataset.

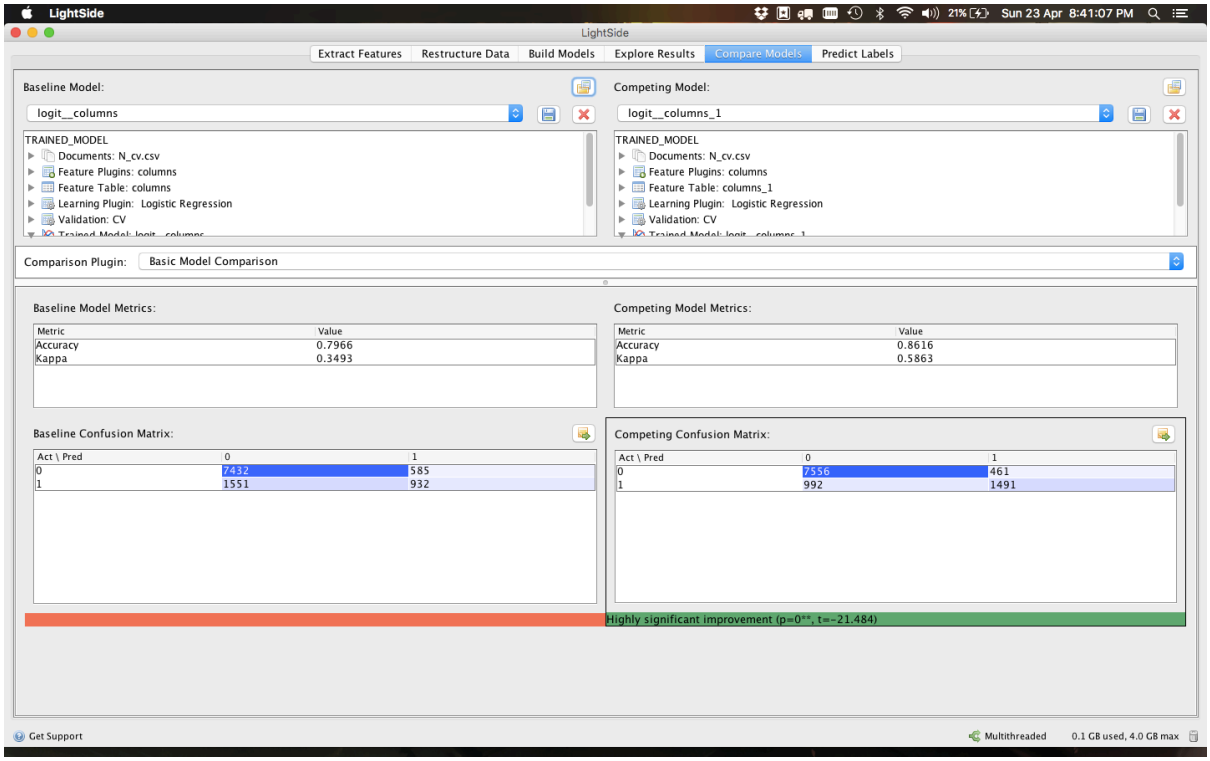

FIGURE 2. RESULT FROM 2ND ROUND OF ERROR ANALYSIS

3rd Round of Error Analysis:

Lastly, I used the trained model from my last iteration and tested it on my *dev* set and received the following performance metrics:

Metric	Value
Accuracy	0.8443
Kappa	0.5702

Confusion Matrix:

Actual\Predicted	0	1
0	2053	215
1	252	479

On exploring the results from testing on the dev set, I chose the lower left cell, since it had the higher number of misclassified instances. After setting the frequency, horizontal absolute difference and the feature weight, and sorting it on decreasing order of the feature weights, I chose the value

of Normalized Satisfaction Level, since it showed the ideal combination of high absolute difference, feature weight and frequency.

On analyzing the results between correctly predicted and wrongly predicted data, I found that for the wrongly predicted results, there were a number of cases where the satisfaction levels were either very low or very high and the number of average monthly hours were very high. I looked at this relation, because I thought these two could be a particular indication of an employee's dissatisfaction with their work environment, and high amount of work, as this could indicate an employee's frustration with the job. For the satisfaction level range, I chose to correct values for employees which satisfaction levels were below 0.37 or greater than 0.5 and who had to work for more than 155 hours per month. To resolve this issue, I added a new feature, that would reduce the weight the model, would give for employees who showed a low value for this combination of 2 features. I did this by introducing a new feature, which contained values which were a product of both the above two features in a particular combination, that had a baseline set.

After adding this new column feature, and extracting and building the model again on my CV set, I found a "Highly Significant Improvement" on comparing my original model and new model. My new model performance metrics were:

Metric	Value
Accuracy	0.8673
Kappa	0.6135

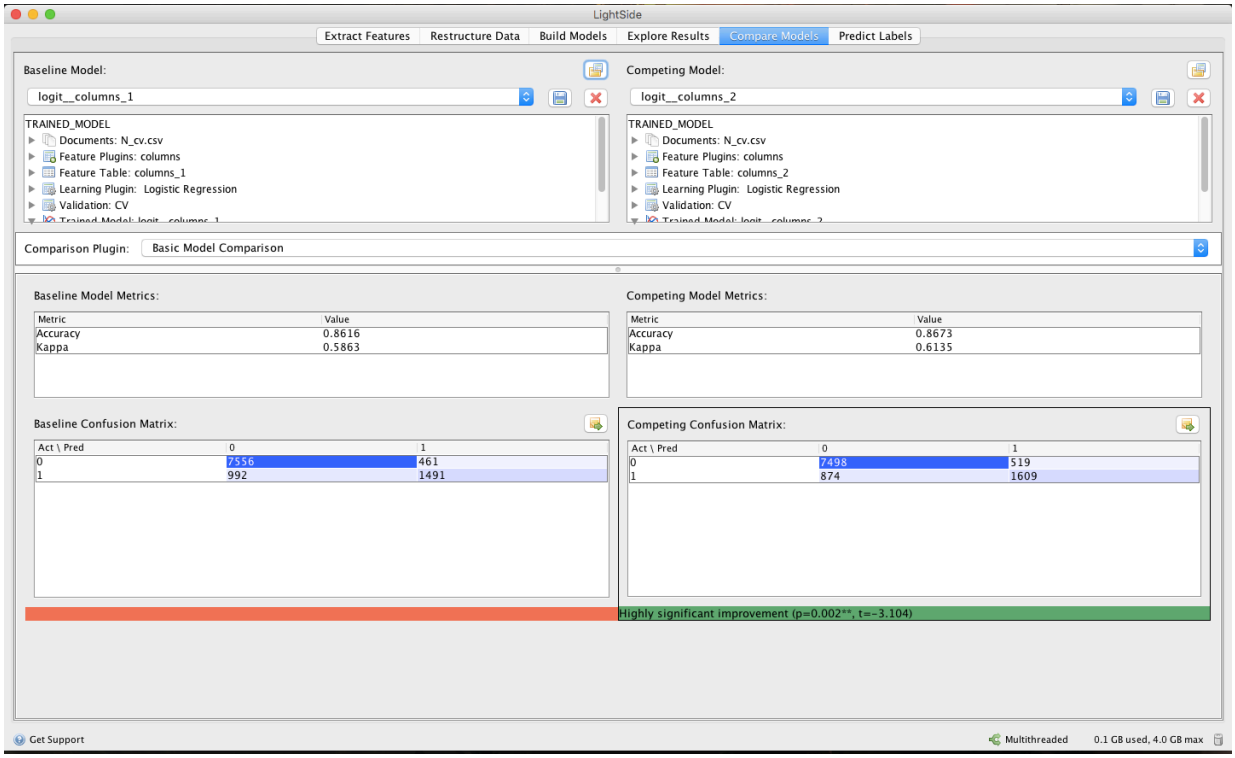

FIGURE 3. RESULT FROM 3RD ROUND OF ERROR ANALYSIS

As can be seen from these results, through error analysis, I was able to significantly improve my performance, and now it is better than the baseline performance score I had obtained using Naïve Bayes, but which I chose not to pursue.

5.Parameter Tuning:

Next, I chose to tune the parameters of the Logistic Regression algorithm parameters to find the optimal values for that model such that the algorithm gives the best performance.

For this, I used the Weka Explorer tool.

I loaded my *CV* set into the tool, and then chose CVParameterSelection as the classifier. Since my algorithm of choice was Logistic Regression, I didn't have too many parameters to choose to tune. I chose to tune the number of Iterations. For the settings, I chose 1 to 100 in steps of 10.

From the results of running this, I found the optimal number of Iterations = 10

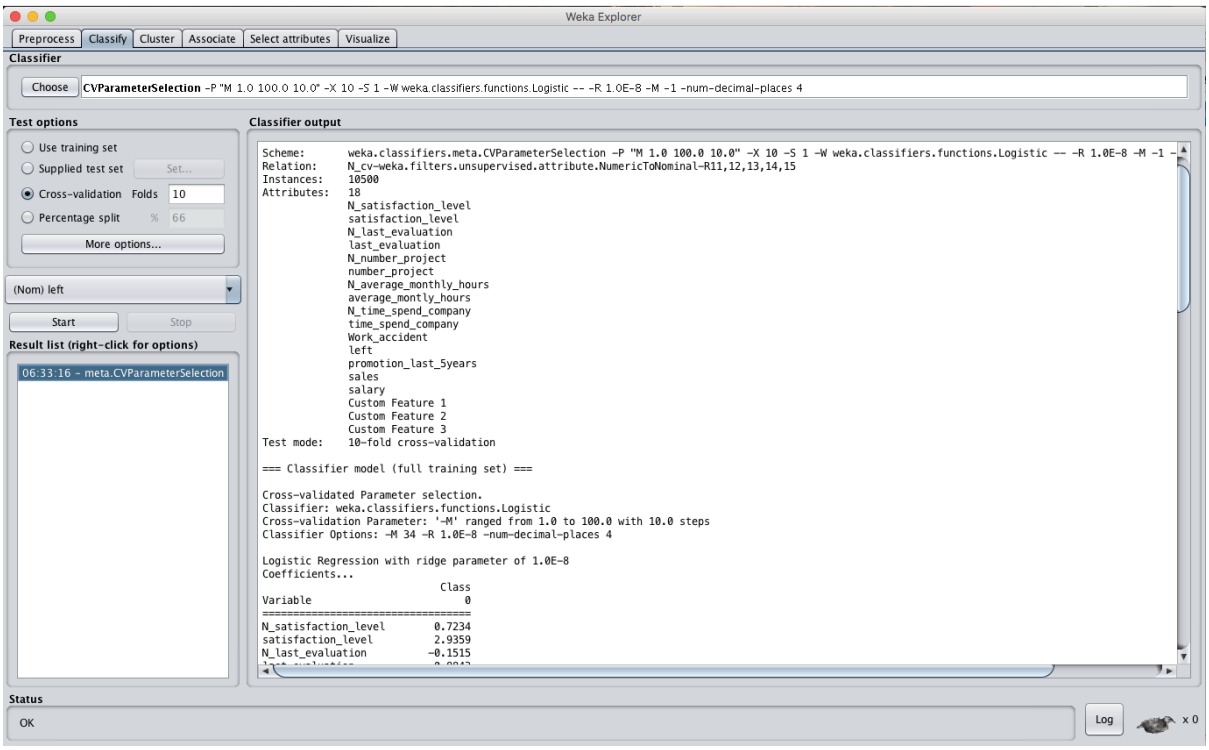

FIGURE 4. RESULT FROM CV PARAMETER TUNING

6. Feature Selection:

I also did a round of feature selection, to check if this would make any difference to the performance of my model. I was initially reluctant to do this, since the number of features that were originally present in my dataset were limited (only 9 features). I got extra features, after performing my three rounds of error analysis (new total = 12 features). For this I checked the Feature Selection parameter in LightSide, before training my model over the *CV* set, including my new features.

Number of Features	Accuracy	Kappa
12	0.8673	0.6135
10	0.8656	0.6088
8	0.8557	0.5803
5	0.8259	0.4309

From the above results, it can be seen, that the performance of the model, actually decreased when lesser number of features were chosen. Hence, going to show that all the 12 features are important.

Results

For the final result, I combined by *dev* and *cv* set, over which I build my model to evaluate on the test set, which I had set aside at the beginning. I took the test data set and then I added the new features that I had developed through my error analysis. I then extracted the features from my improved combined training set (*dev* and *cv*), build my model over the new features and then tested my tuned improved model on the test set data.

I got the following performance metrics:

Metric	Value
Accuracy	0.8527
Kappa	0.5846

The confusion matrix results were as follows:

Actual\Predicted	0	1
0	1044	99
1	122	235

Next, I trained a new model, without my new features, just keeping the original data set features, and evaluated its performance on the test set. This would be an estimate of the original model's performance on the test set, without any improvements or modifications.

From the result obtained, I then compared the two models on LightSide, and found that my new model showed "Highly Significant Improvement" as compared to the original model.

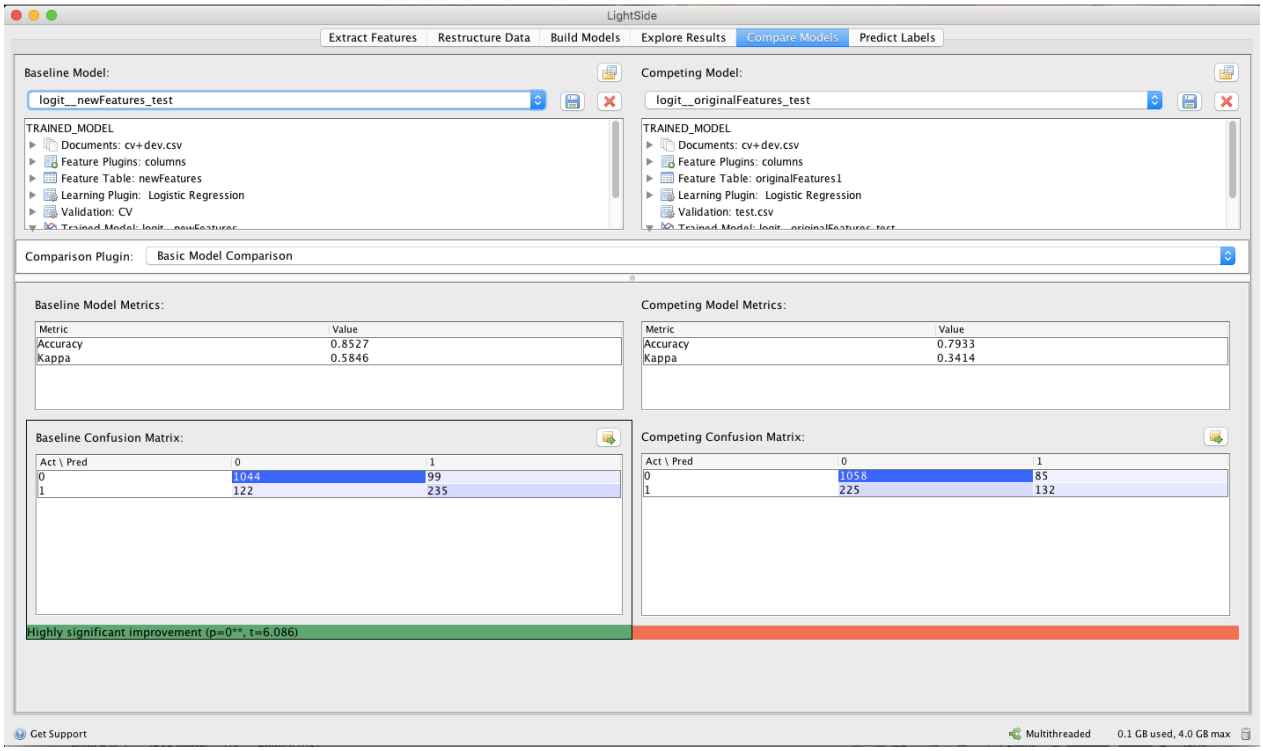

FIGURE 5. COMPARISON BETWEEN ORIGINAL MODEL AND NEW MODEL ON TEST SET

Conclusion

For this project, I compared various models and tried to make a prediction of factors that could lead to employees leaving a company. I used Logistic Regression with L2 Regularization as the algorithm on which to build my model. From my process of error analysis, I found that the following factors played an important role in the improvement in the performance of my model:

- The satisfaction levels of employees: This can be a direct indicator of an employee's contentment at their current work environment.

- The number of projects, the employees were working on: This is an indirect indicator, although from my results, I noticed that employees who were working on less number of projects, showed more tendency to leave the company.

- The average number of hours per month, they were working: This again was an indirect indicator, but when clubbed with the factors mentioned above, a high number of average work hours with low satisfaction levels, could show that the employee is being or feeling overworked. Similarly, a low average number of monthly hours, and low number of projects, indicated that the employees were probably getting disenchanted with their current work environment, and were looking for a change.

Acknowledgement

I would like to thank Professor Carolyn Rose, for her guidance and teaching throughout this semester. I learned a lot during this semester. Initially, I struggled with some of the topics. But as we moved forward, I understood more and I remembered the professor's analogy to a Spiral method of learning. I feel very happy with the progress I have made this semester. This course has piqued my interest in the subject of Machine Learning and I will continue to study and hopefully work in this field.

I would also like to thank the Teaching Assistant's in this course for their help both with the assignments and the project. It was good to get a different perspective, and explore new ideas, after discussions with them.

References

[1]Punnoose, R., & Ajit, P. (2016). Prediction of Employee Turnover in Organizations using Machine Learning Algorithms. *(IJARAI) International Journal of Advanced Research in Artificial Intelligence*,9(5)

[2]Sikaroudi, A., Ghousi, R., & Sikaroudi, (2015). A data mining approach to employee turnover prediction (case study: Arak automotive parts manufacturing). *Journals of Industrial and Systems Engineering,*8(4), 106-121.

[3]B. W. Swider, & R. D. Zimmerman, (2010). Born to burnout: A meta-analytic path model of personality, job burnout, and work outcomes, *Journal of Vocational Behavior*, 76(3), 487-506.

[4]J. L. Cotton & J. M. Tuttle, (1986). Employee turnover: A meta-analysis and review with implications for research, *Academy of management Review*, 11(1), 55-70.

[5]Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, *2*, 841-848.