Predicting Employee Attrition For Augmenting Institutional Yield

Abhinav Sharma
abhinav18002@iiitd.ac.in

Atul Verma
atul18027@iiitd.ac,in

Kshitiz Singh
kshitiz18154@iiitd.ac.in

Abstract

The developing interests in machine learning principles among business pioneers and leaders request that analysts investigate its utilization inside business associations. The loss of talented employees is detrimental to both the morale and the institutional yield. This project studies the trends in employee attrition and predicts the same using machine learning principles. The data-set that has been used for the project is a synthetic one created by IBM Watson, wherein each employee has been described using 34 features. The project employs the use of classification techniques like Logistic Regression, Gaussian Naive Bayes, SVM, Decision Trees etc along with evaluation metrics like accuracy, precision etc. to compare the performance. Specialized models such as XGBoost, Gradient Boosting were also employed. The models were applied to data-set in three settings viz Baseline, Up-sampled, Down-sampled. Logistic Regression (80.01) and Random Forest (91.75) achieved the best precision in the baseline condition. For accuracy also Random Forest and Logistic Regression performed the best. Finally the most important features were figured out with the help of Random Forest.

Keywords - employee attrition, predictive analysis, institutional yield, feature selection, machine learning models.

Github repo - Link to Repository

1. Introduction

Employee Attrition can be defined as the natural process in which the employees of an organization or an institution leave the workforce and are not immediately replaced. The reason for attrition can range from personal reasons such as low salaries to hostile work environments. Employee attrition can be categorized into two categories, viz Voluntary Attrition and Involuntary Attrition. Voluntary attrition implies that the employee leaves an organization due to personal reasons. On the other hand, involuntary attrition occurs when an employee is removed from the organization due to low productivity or other reasons. Loss of employees via attrition has severe impacts on the yield of an organization. Finding eligible candidates to replace the ones that have left is a daunting task. This not only leads to higher costs but also induces a relatively inexperienced workforce in the organization. Continuous employee loss also disrupts the work chain and leads to delayed deadlines and lower customer satisfaction. Higher employee attrition diminishes the brand value of an organization.
In this project, the team strives to use machine learning principles to predict employee attrition, provide managerial insights to prevent attrition, and finally rule out and present the factors that lead to attrition. The project employs the use of models like Logistic Regression, Naive Bayes Classifier, Decision Trees, Random Forests, SVM and Multi Layer Perceptron on an IBM Watson generated synthetic data set.

2. Literature Survey

The paper [1] starts off with describing what is employee attrition, and why it is a major issue faced by institutions across the globe. The paper aimed at predicting Voluntary Employee Attrition within a company using a K-Nearest Neighbours Algorithm, and compare its performance with other models, including Naive Bayes, Logistic Regression and NLP. The authors performed data preprocessing by converting categorical feature values into numerical ones, like converting salary values, that were either ”low”, ”medium” and ”high” to 0,1 and 2. A 70-30 train-test split was created on the data set, and the various model”s performances were evaluated using metrics like Area-Under-Curve, Accuracy and F1 Score. The results of this research showed the superiority of the KNN classifier in terms of accuracy and predictive effectiveness, by means of the ROC curve.

The second paper [2] talked about how classification algorithms often perform unreliably on data sets with large sizes. These data-sets are also often prone to class imbalances, redundant features or noise. The paper applied di-

mensionality reduction by PCA on the Lung-Cancer data set, which was followed by a SMOTE re-sampling to balance the different class distributions. This was followed by applying a Naive-Bayes Classifier on the modified data set, the performance of which was evaluated across four metrics: Overall accuracy, False Positive Rate, Precision and Recall.The results obtained showed that the least misclassifications occurred when PCA was applied followed by applying SMOTE re-sampling twice. Applying SMOTE twice balanced the distributions of the two minority classes, thus giving the best results.

3. Data-Set

The data set is a fictional data set created by IBM data scientists. It has 1470 instances and 34 features (27 numerical and 7 categorical) describing each employee. The target variable - "Attrition," is imbalanced. We have 83% of employees who have not left the company and 17% who have left the company. If one variable is highly correlated to another variable, it will lead to skewed or misleading results.
The correlation matrix showed that the attributes "JobLevel" and "MonthlyIncome" have a high correlation (0.95), and features such as "YearsAtCompany", " YearsInCurrentRole", "YearsSinceLastPromotion" and "YearsWithCurrentManager" have a correlation close to 0.7.
It was observed that younger employees and females have a higher attrition rate. The Sales Department has the highest attrition rate than any other department in the organization. Employees with lower monthly income have a high attrition rate, which is quite obvious. An employee that works extra hours tends to have a higher attrition rate. The job role with the least attrition is of a Research Director, and the one with maximum attrition is of a Sales Representative.
Any formal institution would have all its employees well documented, and further, in real-life scenarios, the class imbalance is highly prevalent. Thus applying machine learning principles on this data set will yield results closer to the practical world.

3.1. Data Preparation and Preprocessing

3.1.1 Changing Datatype

There are some string variables in the data set ("Attrition", "Overtime", "Gender") which were binarized. Categorical attributes such as "BusinessTravel", "Department", "MartialStatus", "EducationField" and "JobRole" were transformed using one-hot encoding. A new attribute "HolisticSatisfaction" has been added which represents the sum of values of the attributes "EnvironmentSatisfaction, JobSatisfaction", "JobInvolvement" and "RelationshipSatisfaction". Employees with less "HolisticSatisfaction" tend to leave the organisation.

3.1.2 Feature Selection

There are some attributes that are meaningless to the attrition prediction. "EmployeeCount", "Over18" and "StandardHours" have the same value for all the employees. Also, "EmployeeNumber" according to its definition represents the employee"s ID. As these attributes don"t provide any information for prediction. So, these attributes were dropped from the data set. Further feature selection was conducted later in different models.

3.1.3 Removal of Outliers

To increase the accuracy of the models, we removed the outliers of the numeric attributes "MonthlyIncome", "TotalWorkingYears and "YearsAtCompany" by calculating Z-Score. The numeric variables which are actually representing some categories were not taken into consideration.

3.1.4 Feature Normalization/Scaling

The goal is to change the values of numeric attributes in the data set to use a common scale. The numeric attributes, "Age", "DailyRate", "DistanceFromHome", "HourlyRate", "MonthlyRate", "MonthlyIncome", "NumCompaniesWorked", "PercentSalaryHike", "TotalWorkingYears", "TrainingTimesLastYear", "YearsAtCompany", "YearsWithCurrManager", "YearsInCurrentRole", and "YearsSinceLastPromotion" are skewed. So, these skewed features were normalized.

4. Methodology

4.1. Classification Techniques

The data-set is highly imbalanced, therefore, for all classification algorithms three separate instances of data-set are prepared viz baseline(without any change), up-sampling (the minority class using SMOTE) and down-sampling (the majority class using sklearn resample). Then the classification methods are applied on all the three instances. The classification techniques that have been used within the domain of the project are Logistic Regression, Gaussian Naive Bayes, Decision Tree Classifier, Random Forest Classifier, Perceptron, Multi-Layer Perceptron and SVM/SVC.
For Decision Trees and Random Forests initially a grid search is done to get the optimal depth and then the further investigation was carried on. For MLP, initially a search is done to get the optimal number of hidden layers and then a search to find the optimal number of hidden units in those layers. Finally, the MLP was tested with all the variations of the activation functions viz. identity, logistic, sigmoid,

tanh. For the Support Vector Classifier(SVM/SVC) the results with all the kernels were explored.
Since the dataset is highly imbalanced, algorithms such as XGBoostClassifier and Gradient Boosting(which are used in these scenarios) with employing a grid search initially to get the optimal depths for both.
In the end, KMeans clustering (unsupervised learning algorithm) was also employed for they are used in anomaly detection and class imbalance this severe can act somewhat equivalent to anomaly detection.

4.2. Evaluation

The evaluation metrics that have been used in this project are accuracy, precision, recall and f1 score. Accuracy is the ratio of correctly defined samples to the total number of samples.

$$Accuracy = (TP + TN)/(TP + FP + FN + TN)$$

Here TP represents True Positive (Correctly predicted positive class), FP represents False Positive (Incorrectly predicted positive class), FN represents False Negative (Incorrectly predicted negative class) and TN represents True Negative (Correctly predicted negative class). Precision is the ratio of true positives to the sum of true of positives and false negatives.

$$Precision = (TP)/(TP + FP)$$

The only difference between precision and recall mathematically is that recall takes into consideration false negatives instead of false positives.

$$Recall = (TP)/(TP + FN)$$

Intuitively precision refers to percentage of the results which are relevant whereas recall refers to the total relevant results correctly classified by the algorithm. F1 Score is the harmonic mean of precision and recall. F1 score reflects a value that is combination of the effects of both precision and recall.

$$F1Score = 2 * Precision * Recall/(Precision + Recall)$$

Since the data set is highly imbalanced with the negative class having 84% of the total instances, accuracy is not a good measure for model comparison as any naive classifier would also yield 84% accuracy. Precision, however, can be a metric that can be used to compare the models as it only takes into consideration the positive class and we are least interested in true negatives. Recall can also provide an insight of how well is the minority class detected. In simpler terms, precision defines how correctly is the minor class being detected and recall states how much is the minor class being detected. For the generation of the statistics, inbuilt methods from Sklearn are used. The scores provided for all the metrics are obtained after performing kfold cross validation to eliminate the chance of a favourable or an unfavourable split. After this ROC curve and PR(Precision Recall) Curves were drawn for all the models. The ROC curve plots True Positive Rate (Y axis) and False Positive Rate (X axis) tells us about the measure of separability of a model. Precision Recall Curve plots Precision (Y axis) and Recall (X axis) and is more useful in practice for needle-in-haystack type problems as it focuses more on positive class detection rather than negative class. The Precision-Recall curve displays the fact that whether a class is being detected(recall) and if it is being detected what is the degree of correctness(precision) of that detection. The more the area under curve (AUC) for both the curves, the better is the performance of the model.

5. Results and Analysis

Model	Accuracy	Precision	Recall
Logistic Regression	89.38	80.91	44.41
Gaussian-NB	41.52	19.86	89.22
Decision Tree	82.03	41.96	33.95
Random Forest	86.62	91.75	17.15
Perceptron	79.67	59.03	53.87
MLP	88.87	78.11	42.23
SVM	89.31	82.27	41.96
Gradient Boosting	86.26	65.86	32.86
XGBoost	87.42	72.43	34.59
KMeans	51.47	16.99	48.38

Table 1. Evaluation Scores for baseline models

The model that yielded the highest precision (91.75) is Random Forest in baseline setting. After trying various depths by the application of Grid Search CV this precision was achieved. The best metrics were achieved at depth 7 and after that all the metrics flattened out. At the lower depths the random forest had a significantly low value for the metrics. Random Forest showed a promising accuracy of 86.62. On up-sampling the minority class, there was a drop in the precision and a hike in recall whereas the accuracy remained almost same for random forest. On down-sampling the majority class, the fall in precision was even greater and the hike in recall also increased. Accuracy for all the models with the exception of Gaussian Naive Bayes remained around 80% in the baseline setting. Logistic Regression also showed satisfactory precision in baseline setting (79.01). However, for precision and recall trends similar to random forest were observed on up-sampling the minority and down-sampling the majority in case of Logistic

Model	Accuracy	Precision	Recall
Logistic Regression	74.54	35.64	74.97
Gaussian-NB	37.75	19.04	90.56
Decision Tree	62.62	24.68	65.98
Random Forest	74.11	34.79	73.07
Perceptron	60.14	36.47	74.21
MLP	74.39	35.01	72.47
SVM	76.21	37.44	75.04
Gradient Boosing	73.09	33.6	71.47
XGBoost	73.46	33.54	44.87
KMeans	50.48	17.27	52.05

Table 2. Evaluation Scores after down-sampling data-set using Sklearn resample

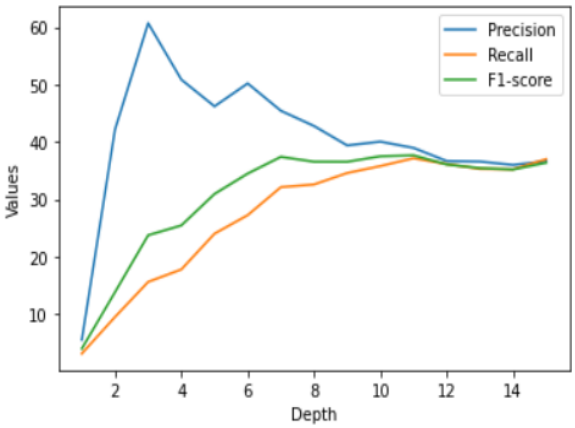

Figure 1. Max Depth vs Metrics for Decision Trees

Regression.Gaussian-NB performed the worst in terms of accuracy (41.52) among all the models revealing its lack in understanding the data set. The precision reported by GNB is also unsatisfactory describing its inability to correctly predict even after good detection (high recall). No clear trend can be ruled out for precision and recall on the up-sampling and down-sampling case with GNB.

The decision tree had a significantly low precision and recall as compared to random forest in all settings depicting its inability to handle the minority class in comparison to random forest.The search for optimal depth was conducted on decision tree too, with optimal depth being 8 at which all the three metrics viz. precision, recall and f1 score had a decent value. Before this the precision values were high (indicating that minority class was detected correctly) but the recall was quite less(indicating the minority class wasn't detected much though). Increasing the depth brought down

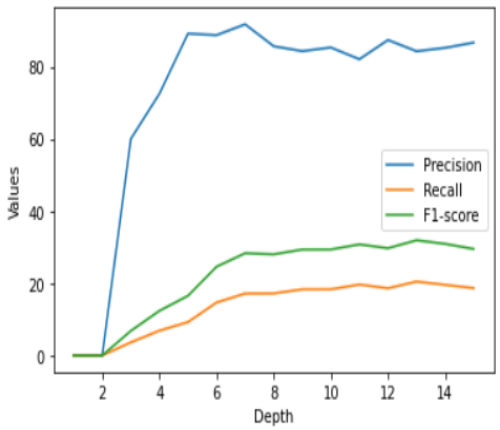

Figure 2. Max Depth vs Metrics for Random Forests

the precision and increased recall with f1 score being almost the same.

In down-sampled and up-sampled settings Perceptron shows a severe drop in accuracy and precision and an increase in recall. For MLP initially a search was conducted to get the values of metric vs the number of hidden layers (keeping hidden units in each layer as 100). This search yielded that the best results were generated with one layer. Further a second search was conducted which explored the number of hidden units in that layer and the optimal value was achieved at around 25 units.

Then all the activation functions were tried out and almost all them yielded similar outputs with sigmoid performing slightly better. In down-sampled and up-sampled settings MLP shows a severe drop in accuracy and precision and an increase in recall. This trend accelerated further in the down-sampling setting.

Model	Accuracy	Precision	Recall
Logistic Regression	77.74	39.22	73.26
GaussianNB	46.25	20.7	85.14
Decision Tree	77.75	32.77	35.99
Random Forest	86.62	69.98	30.81
Perceptron	75.4	43.9	68.23
MLP	79.63	41.61	69.23
SVM	77.53	38.69	72.34
Gradient Boosting	86.76	63.91	39.52
XGBoost	87.27	67.28	38.4
KMeans	53.3	18.45	52.27

Table 3. Evaluation Scores after up-sampling data set using SMOTE

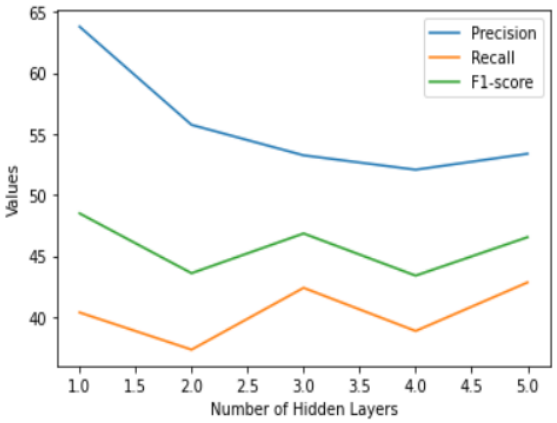

Figure 3. Number Hidden Layers in MLP vs Metrics

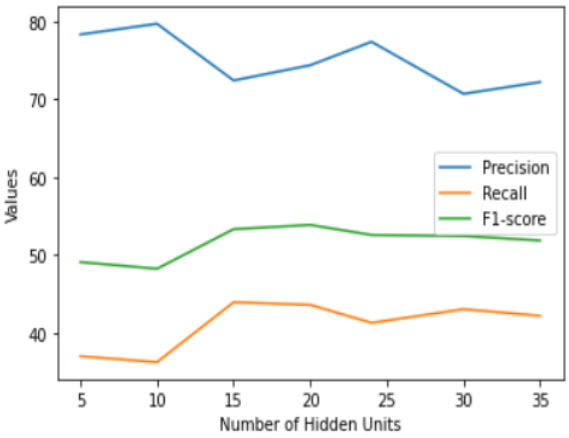

Figure 4. Number of Hidden Units in 1 layer of MLP vs Metrics

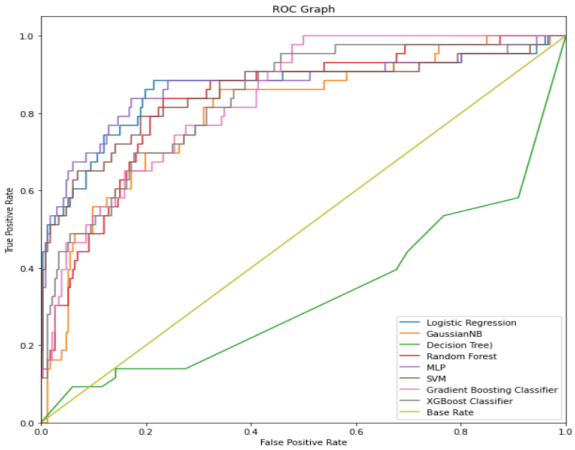

Figure 5. ROC Curve for models

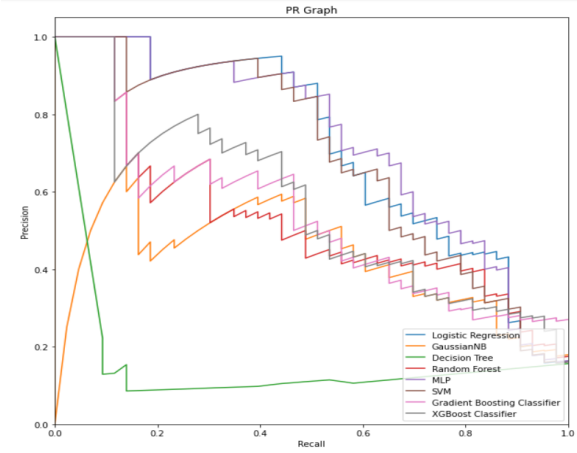

Figure 6. PR Curve for models

For SVM/SVC all the kernels viz. linear, poly, rbf were tried with linear reporting the best values. The other kernels reported a good accuracy but zero precision and zero recall in the baseline setting. This shows that the minority class was not being detected by the SVM and it showed similar results as of a naive model. However, after up-sampling there was a noticeable increase in both the precision (25.46) and recall (63.37). After down-sampling there was an increase in precision and recall but the hike was very small. XGBoost and Gradient Boosting both showed a decent accuracy value (¿85.00) in baseline settings with impressive precision and recall. However like all the other models they too made a dip in accuracy and precision while moving to upsampled and downsampled setting. Recall showed the opposite trend with it being least in baseline setting and increasing as we move from upsampled to downsampled. For these methods also grid search was employed to get the best set of hyperparameters.

Unsupervised learning methods like KMeans has been used but the results haven't been satisfactory and they were somewhere in comparison to GNB. For it initially the elbow criterion was applied to find the number of clusters which turned out to be 2. In all the settings viz baseline, upsampled, downsampled the metrics almost remain the same.

The results were visualized with the help of the ROC and PR curve. Both showed that Logistic Regression, Random Forest, MLP, SVC have a better value for AUC than other classifiers. Among the different settings viz baseline, up-sampled and down-sampled the accuracy and precision were best reported in the baseline setting (with some exceptions). Up-sampling marked a downfall in both accuracy and precision but increased the recall. The trends are steeper in case of down-sampling. A fall in accuracy was observed when the majority class was down-sampled which

shows that it triggers information loss. Finally random forest was employed to get the most important features which were MonthlyIncome, HolisticSatisfaction, Age, Overtime.

6. Conclusion

As seen from accuracy values of baseline condition we can conclude that accuracy indeed is not a good metric in case of imbalanced data. Up-sampling the minority class by generating synthetic samples lead to better detection of the minority class but the degree of trust in that detection falls down. The trend is even steeper when the majority class is down-sampled. The higher precision and lower recall values in baseline condition concludes that the minority class is not detected much but whenever detected the degree of trust is high. On contrary, the higher recall and lower precision value concludes that even though the minority class is being detected but the degree of trust in that detection is very low. The models specialized for imbalanced class scenarios showed promising outcomes in all the metrics. Finally, it can be concluded that Random Forest, MLP, XGBoost, Gradient Boosting, Logistic Regression are the best models among the above mentioned. Unsupervised Clustering Algorithms failed at proper prediction. The important features extracted after application of random forest coincide with the real world which conclude that the model has real world uses despite being trained on a synthetic data-set.

References

[1] Rahul Yedida, Rahul Reddy, Rakshit Vahi, Rahul Jana, Abhilash GV, Deepti Kulkarni. "Employee Attrition Prediction". 02 November 2018. 1

[2] Mehdi Naseriparsa, Mohammad Mansour Riahi Kashani "Combination of PCA with SMOTE Resampling to Boost the Prediction Rate in Lung Cancer Dataset" 1