Use Machine Learning classification algorithms to predict job change in HR Analytics

Tran Nguyen Phuc* Rucha Shende* Nalveer Moocheet*
Ahmed Yasser Mohamed Sabri Eita *

Abstract

The goal of this project is to use Machine Learning(ML) to design a prediction model that will be used by the Human Resource department of a company to predict if an employee will stay or leave the company based on the candidate's personal information. In order to achieve a high performance ML prediction model, We trained and evaluated multiple Machine Learning Classification algorithms such as Support Vector Machine (SVM), Multi-layer Perceptron (MLP) Neural Network, Logistic Regression, Decision Tree, and Random Forest. Moreover, as a secondary goal, we analysed the resulting models to get the features that have higher impact on employee decisions.

1. Introduction

One of the biggest investments for big corporations and even small enterprises is staff recruitment. This includes long periods of paid training which often ends up with the candidate not even joining the company. Analytics has been used in human resources for many years. However, data collection, processing, and analysis have been mostly manual. In this project we use Machine Learning to develop a model that will automate these analytics processes and allow the employers to make fast and accurate data-driven decisions for recruitment.

Using a HR analytics dataset found on kaggle[1] , we start by cleaning and pre-processing the data. We then trained 5 classification algorithms - Logistic Regression, Decision Tree, Random Forest, Support Vector Machine and MLP Neural Network, followed by detailed evaluation, analysis and visualization which not only allows us to achieve our main goal, but also understand the features impacts on employee decisions.

The rest of the report is organized as follows: Section 2 goes through relevant data-set information including some issues such as data imbalance. Section 3 is about the methodology, followed by performance evaluation and visualization in section 4. Section 5 explores feature importance and finally we conclude this report in section 6.

*Concordia University, Montreal, Quebec, Canada

2. Dataset

Our dataset was taken from t[1]. The raw dataset consists of 19158 rows with 13 features (columns) - enrollee_id, City code, city_development_index, gender, relevent_experience, enrolled_university, education_level, major_discipline, experience, company_size, company_type, last_new_job and training_hours. Our prediction targets are '1' which signifies the candidate stayed and '0' for candidates who left the company.

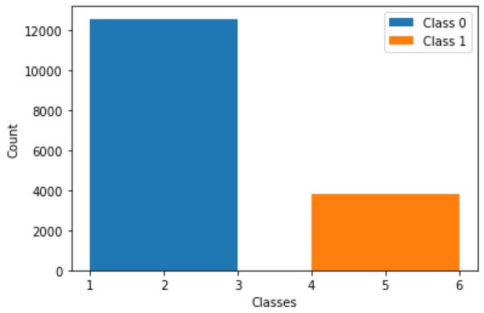

Figure 1: The data imbalance between 2 classes

One of the problems that is often faced when working on a Machine Learning project, is the imbalanced data. There will be no deviations here, this dataset also faced this issue as the target label had 75% '0' (NO) and only 25% '1' (YES), which showed in figure 1. The second issue we confront is the presence of null data in our dataset. An overview of the dataset is given in figure 2. Section 3.1 covers the handling of these issues.

3. Methodology

3.1. Data Pre-processing

Before running the dataset through several algorithms, we have executed a pre-processing phase to clean up the

	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university	education_level	major_discipline	experience	company_size	company_type	last_new_job	training_hours	target
0	8949	city_103	0.920	Male	Has relevent experience	no_enrollment	Graduate	STEM	>20	NaN	NaN	1	36	1.0
1	29725	city_40	0.776	Male	No relevent experience	no_enrollment	Graduate	STEM	15	50-99	Pvt Ltd	>4	47	0.0
2	11561	city_21	0.624	NaN	No relevent experience	Full time course	Graduate	STEM	5	NaN	NaN	never	83	0.0
3	33241	city_115	0.789	NaN	No relevent experience	NaN	Graduate	Business Degree	<1	NaN	Pvt Ltd	never	52	1.0
4	666	city_162	0.767	Male	Has relevent experience	no_enrollment	Masters	STEM	>20	50-99	Funded Startup	4	8	0.0
5	21651	city_176	0.764	NaN	Has relevent experience	Part time course	Graduate	STEM	11	NaN	NaN	1	24	1.0

Figure 2: The overview of the dataset

data which was then split into a 80% training and 20% testing set. Firstly, non-numerical data was converted to numeric data; this will allow fast calculations during the training phase and also make the data visualization easier. Following that, we have removed any row with three or more null values since three estimates in one sample is not considered credible. In addition, we have used the Multivariate imputation by chained equations (MICE)[2] approach to replace missing variables. Finally, we addressed the imbalanced data-set issue by using re-sampling techniques such as down-sampling where class '0' was down sampled to match the number of class '1' and up-sampling in which class '1' was over sampled to match the number of class '0' [3]. Re-Sampling allowed us to get 2 different data-set, up-sampled (25,090 samples) and down-sampled (7,564 samples). Figure 3 depicts the data set following the preparation stage.

3.2. Model Training

For each of the considered ML Classification algorithms - Logistic Regression, Decision Tree, Random Forest, Neural Network MLP Classifier and Support Vector Machine, we used Hyper-parameter Search and Cross-Validation with the aim to get the optimal prediction models for our dataset. Moreover, We conducted all the experiments on both the up-sampling and down-sampling dataset, after which we compared and selected the best configurations and re-sampled dataset. The following are some of the most important experiment outcomes:

(i) Using up-sampling dataset produced better results than down-sampling.

(ii) Logistic Regression had a better performance when using a Cross-Validation of 5 folds with parameters: *[solver: liblinear, C : 0.1, max_iter : 50']*.

(iii) The best performing Neural Network had parameters: *['activation': 'tanh', 'alpha': 0.1, 'solver': 'adam']*. Cross Validation was not effective on the MLP Classifier.

(iv) Random Forest had the best performance with *default parameters* and no Cross-Validation.

(v) Best performing SVM was with no Cross-Validation and using the *'rbf' kernel*. Hyper-Parameter tuning was not effective as it caused over fitting with training accuracy up to 99.9% percent but validation score of less than 50%.

The **SVM** had the best performance out of all the other classification algorithms with an accuracy of 94%. The detailed evaluation and visualization of the model performance are explored in the next section.

4. Performance Evaluation

As stated in the last section, SVM was found to be the best classification algorithm for out dataset. The performance metrics used for evaluation and comparison are accuracy, Receiver operating characteristic (ROC), precision and recall. The details of results for all algorithms on the up-sampling dataset are shown in table 1 and on the down-sampling dataset are shown in table 2.

Furthermore, we plotted the confusion matrix of that best model to show the reliability of its predictions in Figure 5. As it may be seen from the figure, the SVM model displayed only about 5% false predictions; 1.95% False Positive and 3.27% False Negative.We also plot the feature importance to analyse which factor impacts employees decisions.

5. Feature Importance

In each trained model, we have extracted the features importance. For most of the algorithms, *'city_development_index'*, *'training_hours'* and *'relevant_experiences'* were the features with the highest weight. Figure 6 and Figure 7 shows the features importance of the best 2 performing classifiers; SVM and Random Forest respectively.

	city_development_index	gender	relevent_experience	enrolled_university	education_level	major_discipline	experience	company_size	company_type	last_new_job	training_hours
0	0.920	1.0	1	0.0	2.0	1.0	21.0	2.0	1.0	1.0	36
1	0.776	1.0	0	0.0	2.0	1.0	15.0	1.0	0.0	5.0	47
4	0.767	1.0	1	0.0	3.0	1.0	21.0	1.0	1.0	4.0	8
6	0.920	1.0	1	0.0	1.0	1.0	5.0	1.0	1.0	1.0	24
7	0.762	1.0	1	0.0	2.0	1.0	13.0	-1.0	0.0	5.0	18
8	0.920	1.0	1	0.0	2.0	1.0	7.0	1.0	0.0	1.0	46

Figure 3: Dataset after prepossessing

Algorithm	accuracy	ROC	precision	recall
SVM	94.78%	94.79%	96.03%	93.52%
Decision tree	88.16%	88.09%	83.14%	96.01%
Random forest	91.81%	91.76%	87.81%	97.27%
Neural network	73.75%	73.74%	73.41%	73.24%
Logistic regression	69.23%	69.32%	70.20%	65.40%

Table 1: The details of results for the 5 algorithms on up-sampling dataset.

Algorithm	accuracy	ROC	precision	recall
SVM	65.96%	65.90%	69.03%	56.99%
Decision tree	69.53%	69.52%	69.92%	67.78%
Random forest	76.80%	76.77%	79.24%	72.17%
Neural network	69.33%	69.33%	68.66%	70.31%
Logistic regression	68.67%	68.65%	69.54%	65.65%

Table 2: The details of results for the 5 algorithms on down-sampling dataset.

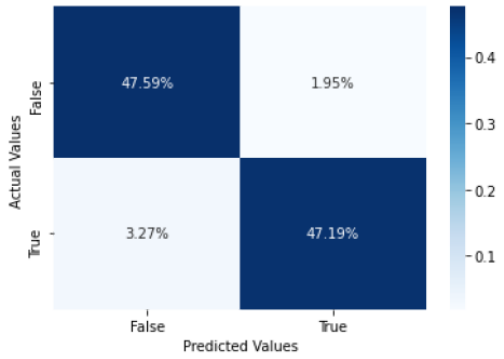

Figure 4: Confusion Matrix of the best performing algorithm (SVM up-sampling with RBF kernel)

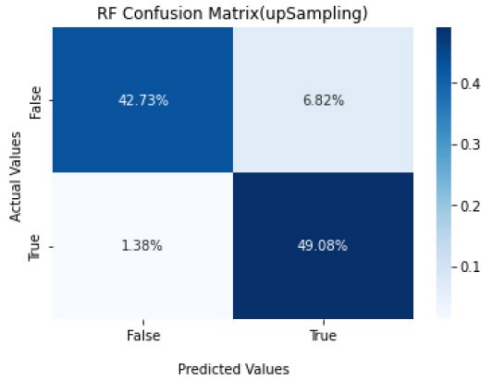

Figure 5: Confusion Matrix of the second best performing algorithm (Random Forest)

6. Conclusion

The aim of this report was to implement a Machine Learning classification algorithm to predict job change in HR analytics. In order to achieve our final prediction model we used several machine learning concepts such as data preprocessing, training of multiple algorithms, resampling, and Hyperparameter tuning. As demonstrated in the report, the best model was the SVM classifier trained with up-sampled data. After further visualization and analysis, we also discovered that the highest impacting features in the prediction was the candidate's city development and relevant experience. Having achieved an accuracy of nearly 95%, precision of 96% and recall of 93%, we can conclude that our model have a high enough performance to meet the goal of this project.

References

[1] Möbius. Hr analytics: Job change of data scientists. 1

[2] Sam Wilson. The mice algorithm. 2

[3] Tara Boyle. Dealing with imbalanced data. 2

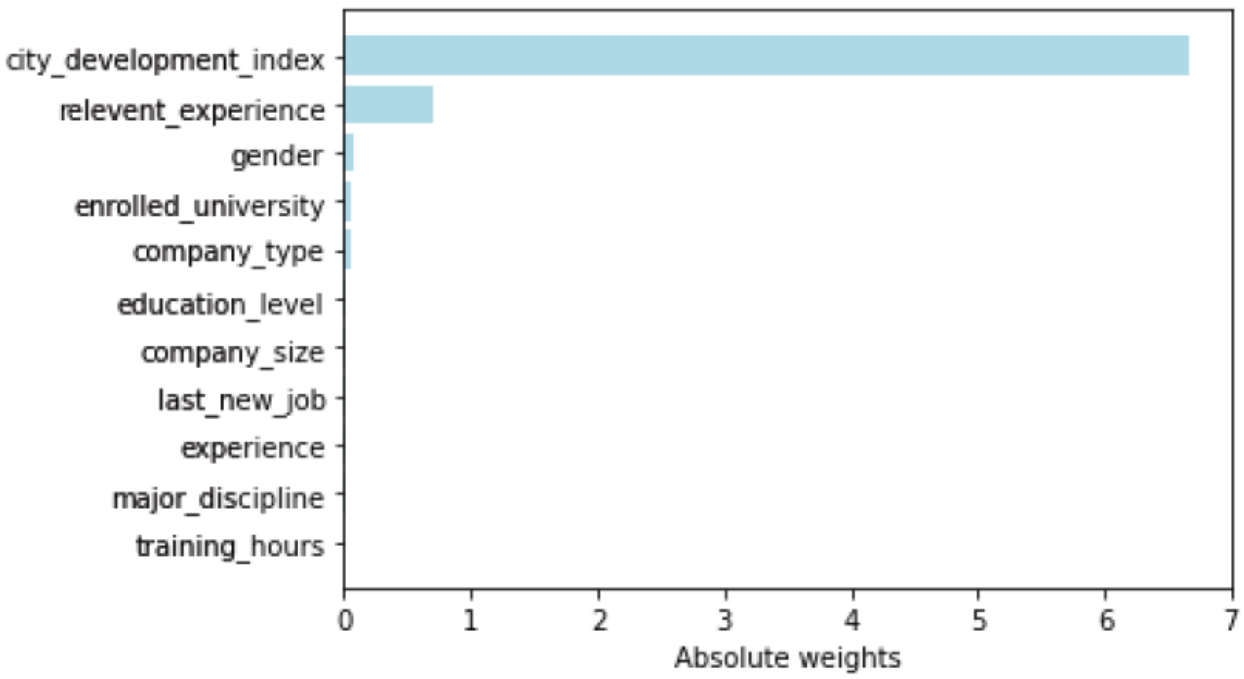

Figure 6: Features importance according to SVM classifier

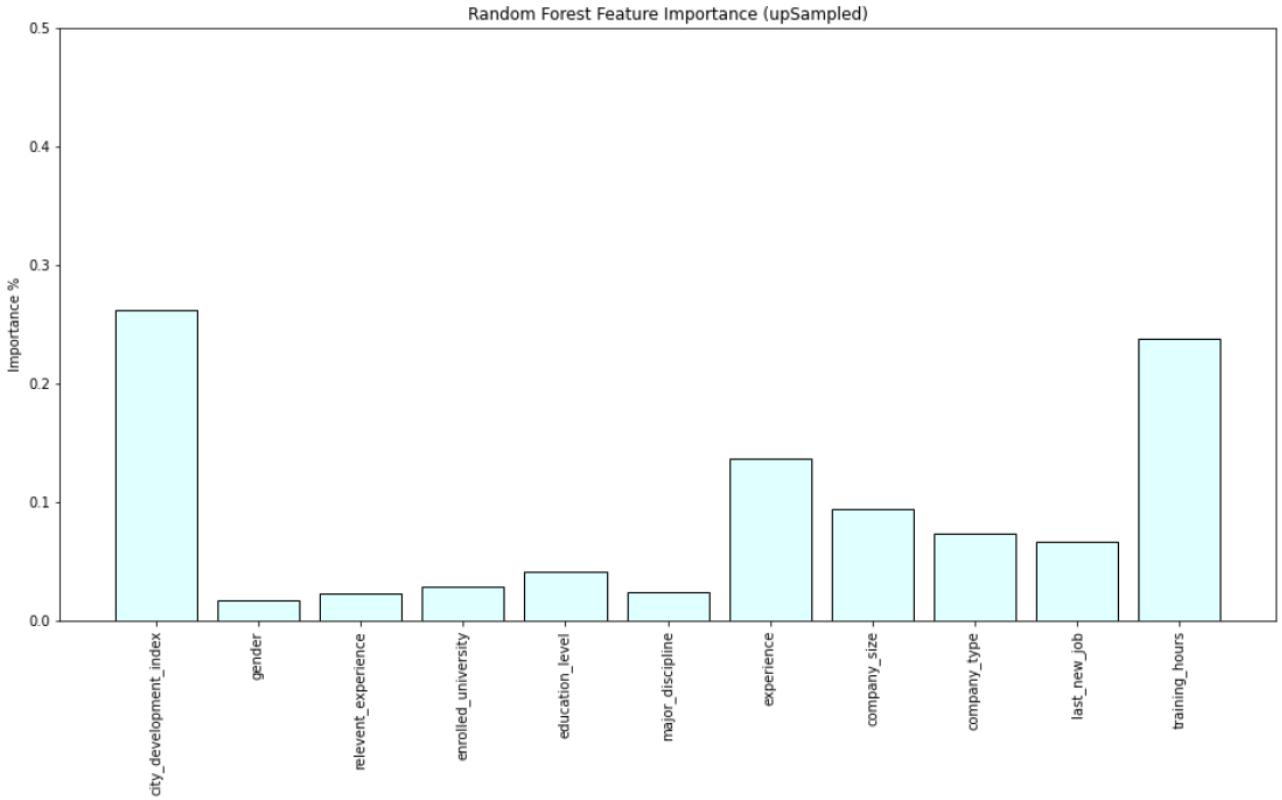

Figure 7: Features importance according to Random Forest classifier