

Machine Learning Engineer Nanodegree

Capstone Project

Topic: Attrition prediction with machine learning approach in human resource management

Ivan Chen
January 4, 2018

I. Definition

Project Overview

In human resource management, HR professionals take the responsibility in designing suitable systems such as work culture, compensation or promotion system to help the company or organization to retain top employees. Attrition in human resources refers to the gradual loss in labor over time.

In general, average attrition rate compared with other companies in same business is acceptable but relatively high attrition may cause trouble for companies. A major problem in high attrition rate is high cost to an organization; for example, human resources team may need to start hiring processes, providing new hire training. Besides, to the organization, high attrition rate may lead to lose knowledge and experience; it may impact daily or business operation. Usually human resources teams take attrition rates as factors into their department budgets to account for potential losses in productivity and the costs with replacing employees.

Machine learning techniques were used to predict employee turnover in prior paper [1]. The dataset used in this project was downloaded from kaggle, (<https://www.kaggle.com/ludobenistant/hr-analytics-1/data>) which was provided by Ludovic benistant for HR Analytic study and practice.

Problem Statement

The attrition rate is determined by dividing the number of employees who left the company or organization during a specific period by the number of employees during the same period. Attrition rate can be computed by monthly, quarterly or annual periods. Consistent rate of attrition is treat as the norm for a company but high attrition rate may induce trouble to a company. To predict and classify high attrition risk of employees are critical for human resources members. With the classified result human resources teams may evaluate the causes of attrition and find solutions.

The attrition prediction problem would be a binary classification problem and the goal of the project is to generate a relatively accurate model to predict attrition or stay of employees based on collected employees data. In the study, a solution was proposed and to be implemented. Benchmark model, evaluation metrics, key feature importance observation will be considered and discussed.

Metrics

The goal of the model is predicting whether an employee will leave or stay in the company, F-beta score is appropriate metric for this kind of binary classification problem. F-beta score is proposed to be the main metric and accuracy as supporting metric.

Accuracy measures how often the classifier makes the correct prediction. It's the ratio of the number of correct predictions to the total number of test data points. That is, accuracy = (True Positives + True Negatives) / Total number of data points.

"F-beta score" is a metric that considers both precision and recall. The formula was described below:

$$F_{\beta} = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

Illustration 1: Metrics: F-score

In the formula of F-beta score, the precision is the ability of the classifier not to label as positive a sample that is negative. For example, in a spam email classification application, it can tell what proportion of messages was classified as spam, actually was spam. That is,

$$\text{Precision} = \text{True Positives} / (\text{True Positives} + \text{False Positives})$$

The recall is the ability of the classifier to find all the positive samples. In a spam email classification application, it can tell what proportion of messages that actually was spam that was classified by us as spam. That is,

$$\text{Recall} = \text{True Positives} / (\text{True Positives} + \text{False Negatives})$$

The F-beta score is a weighted harmonic mean of the precision and recall. User for different interest of application can adjust the weight beta. If the value of beta equals 1 means recall and precision are equally important.

For the attrition prediction problem, I thought precision is more important than recall. If an employee were classified as high risk of attrition candidate, it's better to be a real attrition because HR professionals may take some action based on the prediction result, for example, providing compensation program if the employees are valuable or preventing the employee's improper operations that hurt the company before leaving. So I propose to adjust the beta to smaller than 1 because the model requires high precision.

II. Analysis

Data Exploration

The dataset used in this project was downloaded from kaggle which was provided by Ludovic benistant for HR Analytic study and practice. The dataset consists total 14999 rows with 10 columns. The last one "attrition" column renamed by me from "left" for this study was used as target label of the prediction. Other columns provide related information of the employee such as average monthly working hours, number of projects.

#	Column name	Meaning	Type	Possible value
1	satisfaction_level	Satisfaction Level	numerical	float
2	last_evaluation	Last evaluation	numerical	float
3	number_project	Number of projects	numerical	integer
4	average_monthly_hours	Average monthly hours	numerical	float
5	time_spend_company	Time spent at the company	numerical	float
6	Work_accident	Whether they have had a work accident	numerical	Binary, 1 denotes 'Yes', 0 denotes 'No'.
7	promotion_last_5years	Whether they have had a promotion in the last 5 years	numerical	Binary, 1 denotes 'Yes', 0 denotes 'No'.
8	Departments (original column name: sales)	Departments	categorical	sales, accounting, hr, technical, support, management, IT, product_mgn, marketing, RandD.
9	salary	Salary	categorical	low, medium, high.
10	Attrition (original column name: left)	Whether the employee has left	numerical	Binary, 1 denotes 'Yes', 0 denotes 'No'.

Illustration 2: Feature list

Although the number of columns is not many but they provide important work related information of employees. Besides, the distribution of target variable “attrition” (76.2% is No, “23.8% is Yes) is unbalanced but I think it is not skewed, so the dataset is suitable for this attrition prediction problem.

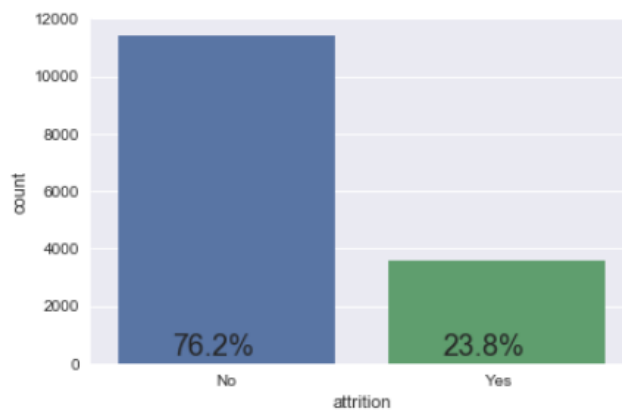
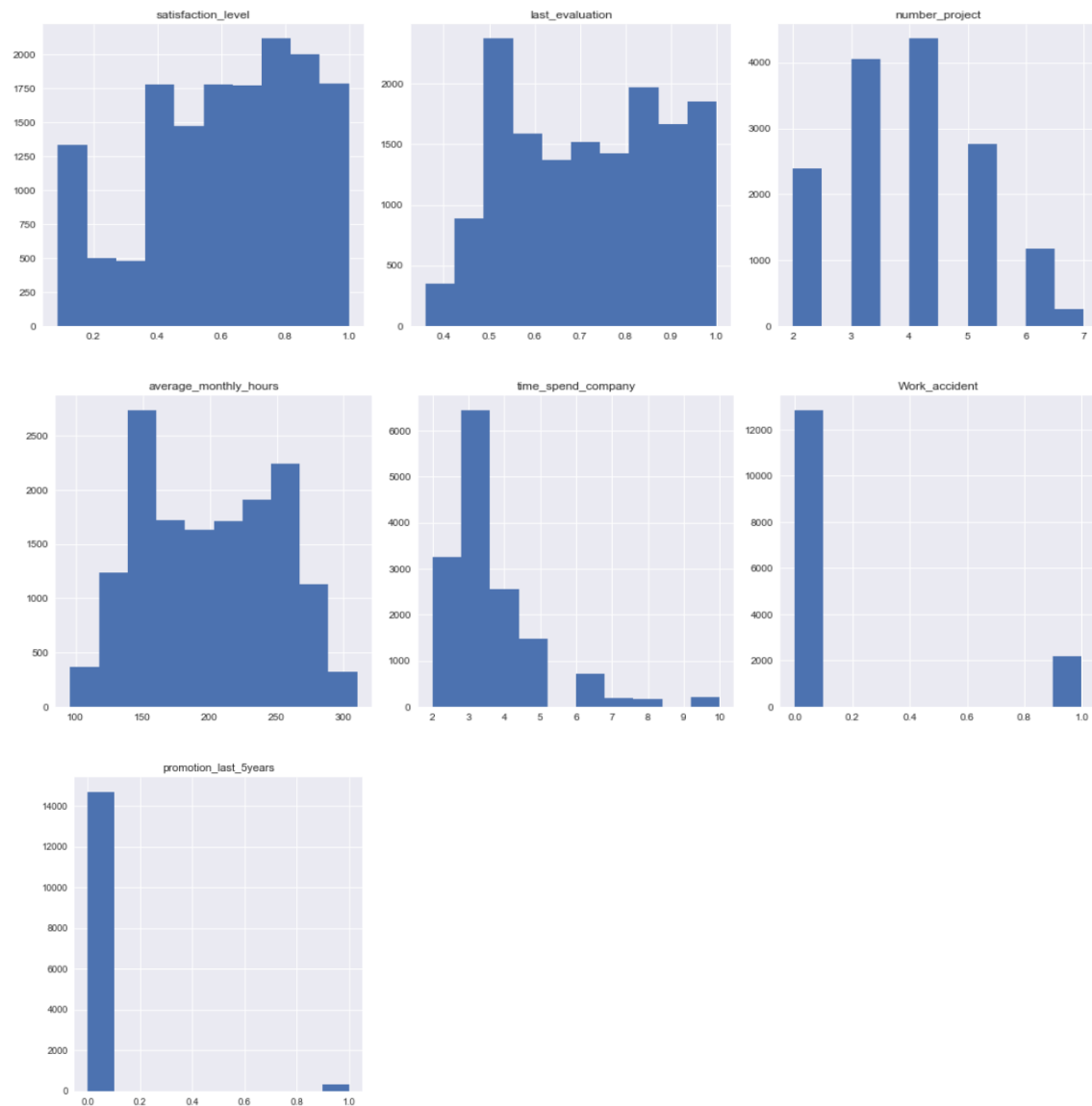


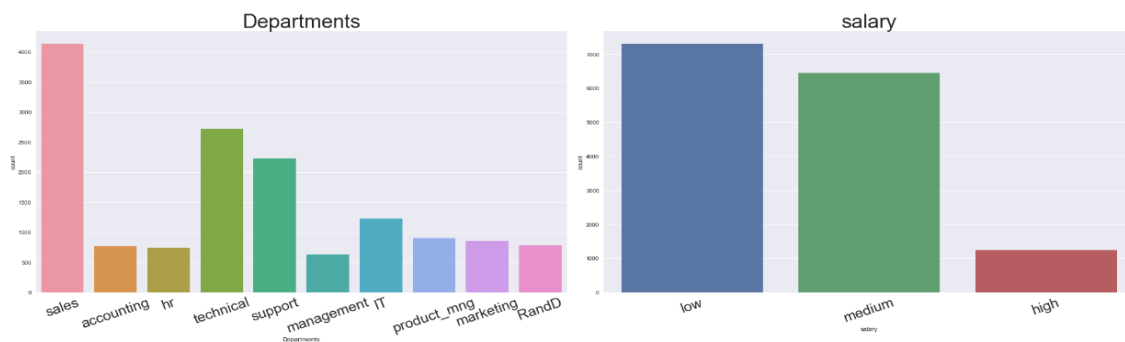
Illustration 3: Attrition in dataset

Exploratory Visualization

There are seven columns are numerical types, the distribution were illustrated as follows. Features “Work_accident” and “promotion_last_5years” are binary value, either 1 (denotes “Yes”) or 0 (denotes “No”).



Two features are categorical, the distribution and possible values are illustrated below.



From the correlation map of features below we may see the feature “average_monthly_hour” has relative high correlation with “number_project”. It denotes that the more projects an employee evolves the higher average monthly working hours he/she takes. The two features may related to employee's working loading also.

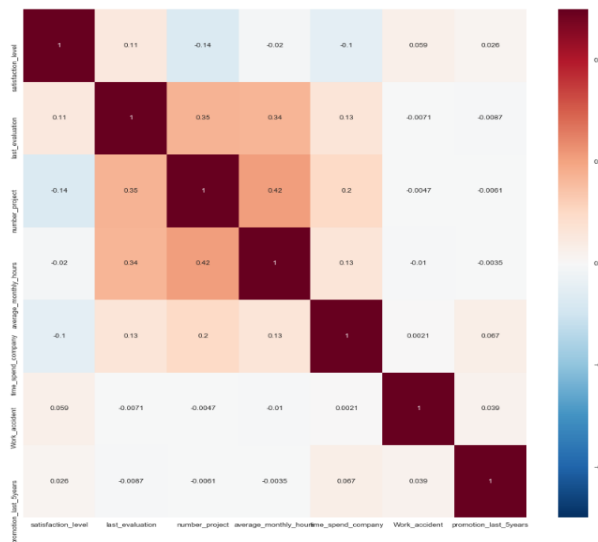


Illustration 4: Features correlation map

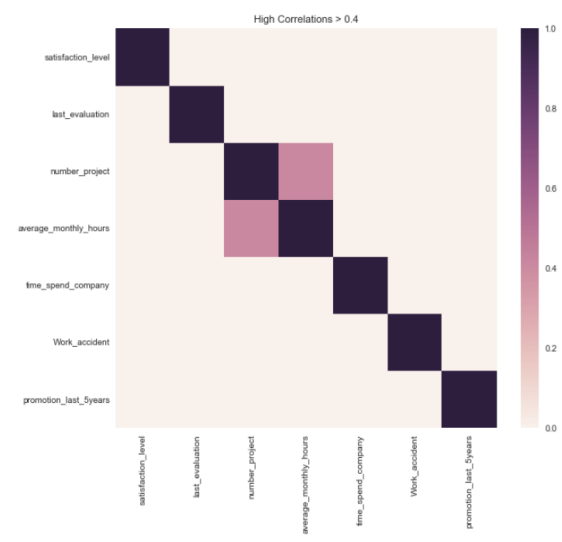


Illustration 5: High correlations > 0.4

Illust

Algorithms and Techniques

The workflow of the solution for this attrition prediction consists the following main steps and described below. For this attrition predicting, three supervised learning models to create a training and prediction pipeline were adopted.

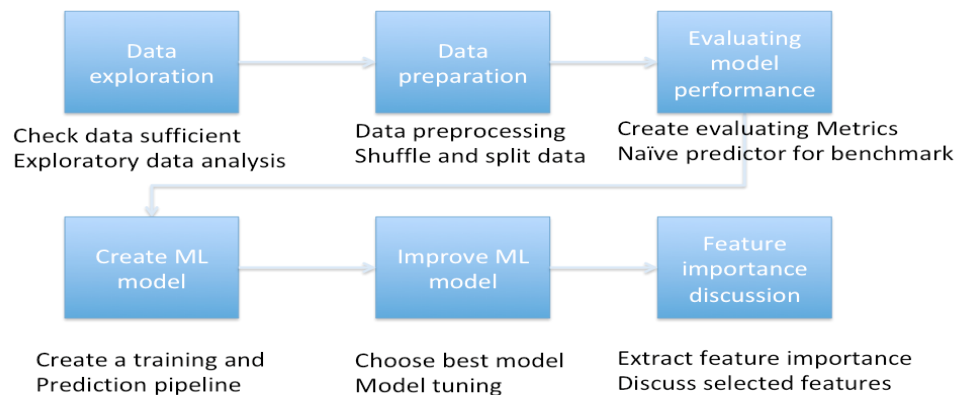


Illustration 6: Workflow of proposed solution

1. Data exploration

First, I will check the dataset whether it is sufficient or suitable for the attrition study or not, then perform exploratory data analysis, such as exploring how many rows in the dataset, what is the attrition ratio in the raw data, which I can judge if there is a large skew of the target label (attrition)? After feature exploration, I may know the types of features, what types are numerical (or binary), what types are categorical they are for later data preprocessing.

2. Data preparation

After data exploration, data preprocessing for different kind of types is needed, such as transferring skewed continuous features, normalization numerical features, using one-hot encoding scheme to convert categorical features to numerical dummy variables. After data was preprocessed properly, then I may shuffle and split data for later training and prediction model creation.

3. Evaluating model performance

In this attrition prediction project, F-beta score metric was chosen to measure the performance of model. Meanwhile, human resources member may care about more precision than recall, a smaller than 1 beta may suitable to get high precision model. Because there is no previous version of model for benchmark, I propose to generate a naive predictor, which always predicts attrition or always non-attrition as base model without any intelligence for comparison.

4. Create machine-learning model

In this attrition prediction, I will adopt three supervised learning models creating a training and prediction pipeline, which are Support Vector Machine (SVM), Decision Trees and AdaBoost ensemble learning. They both have advantages and disadvantages and I think intuitively the AdaBoost ensemble learning or Decision Trees may have better performance in the study, but the measurement metrics may tell us the result finally.

5. Improve machine-learning model

Based on previous evaluation of performance, I may choose the best model and perform model tuning for optimization. If the training time is not a big issue, then I may choose the model with better F-beta score as best model. By using grid search approach with various values of hyper parameters to fine tune the chosen model can get final optimal model.

6. Feature importance discussion

In the final stage, I will explore the feature importance and their relevance with target label (attrition). With the feature importance observation, if the training time is a critical concern, then new model with these top features (say top five) may have a little lower score than the model with completed features but training time may significantly reduced. Besides, with these top important features observation, human resources teams may consider to provide compensation programs to valuable employees or take prevention actions to avoid employees' improper operation to the company before leaving.

Benchmark

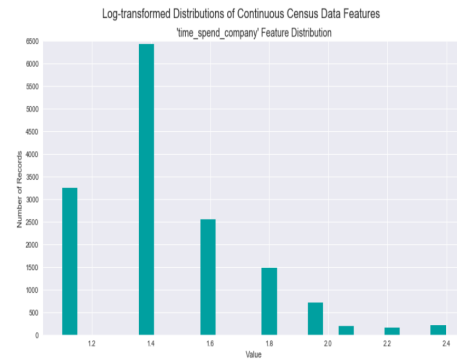
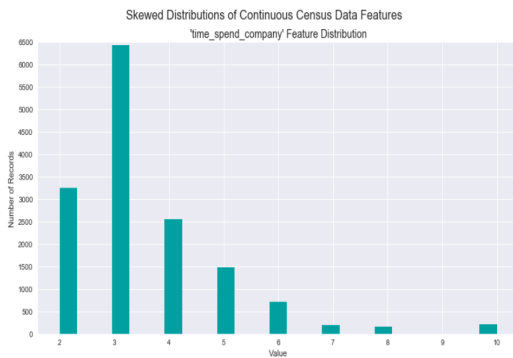
In this project, I propose to generate a naive predictor, which always predicts attrition (value 1) or always non-attrition (value 0) as base model without any intelligence for comparison. F-score measurement can be adopted to measure the result. With this generated cost-effective naive model, we may have base model to compare performance once the proposed solution model is ready.

III. Methodology

Data Preprocessing

After data exploration, data preprocessing for different kind of types is needed, such as transferring skewed continuous features, normalization numerical features, using one-hot encoding scheme to convert categorical features to numerical dummy variables.

In this dataset, the feature "time_spend_company" is a little skewed. A common method to avoid outliers to negatively affect the performance of a learning algorithms is applying logarithmic transformation on the data. We can see the range of distribution was narrowed in the below .



To ensure that each feature is treated equally when applying supervised learners, scaling was applied on all numerical features including 'satisfaction_level', 'last_evaluation', 'number_project', 'average_monthly_hours' and 'time_spend_company'. Two numerical features 'Work_accident', 'promotion_last_5years' are numerical but they are binary values, either 1 or 0 that denotes "Yes" and "No", so they are not needed to apply scaling. Applying a scaling to the data does not change the shape of each feature's distribution. The target variable "attrition" is either 1 or 0, it is not needed to preprocess.

From the data exploration above, we can see there are several features for each record that are non-numeric. Because learning algorithms expect input to be numeric, one-hot encoding scheme was applied to convert those non-numeric features (called categorical variables) to numeric. One-hot encoding creates a "dummy" variable for each possible category of each non-numeric feature. For example, the "salary" feature has three possible entries: low, medium and high, three dummy variables "salary_low", "salary_medium" and "salary_high" were created after the conversion.

After data was preprocessed properly, then I may shuffle and split data for later training and prediction model creation.

Implementation

In this attrition prediction, I have adopted three supervised learning models creating a training and prediction pipeline, which are Support Vector Machine (SVM), Decision Trees and AdaBoost ensemble learning. They both have advantages and disadvantages and I think intuitively the AdaBoost ensemble learning or Decision Trees may have better performance in the study, but the measurement metrics may tell us the result finally.

The three supervised learning models were described briefly as follows:

(1) Support vector machines (SVMs)

Real-world application in industry can be: Faces recognition example using eigenfaces and SVMs, reference: (http://scikit-learn.org/stable/auto_examples/applications/plot_face_recognition.html#sphx-glr-auto-examples-applications-plot-face-recognition-py)

Reference: (<http://scikit-learn.org/stable/modules/svm.html>)

Strengths: . Effective in high dimensional spaces. . Still effective in cases where number of dimensions is greater than the number of samples. . Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient. . Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

Weaknesses:

. If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and . regularization term is crucial.

. SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation (see Scores and probabilities, below).

Since the number of features are small and there may be a clear margin of separation, SVC (Support vector classifier) can be a good candidate to define a clear boundary.

(2) Decision Trees

Real-world application in industry can be: A MIT study examined how decision trees could be used to predict whether an applicant would receive a loan. Reference:

http://courses.media.mit.edu/2008fall/mas622j/Projects/CharlieCocoErnestoMatt/decision_trees/

Reference: <http://scikit-learn.org/stable/modules/tree.html>

Strengths:

- . Simple to understand and to interpret. Trees can be visualized.
- . Requires little data preparation. Other techniques often require data normalization, dummy variables need to be created and blank values to be removed. Note however that this module does not support missing values.
- . The cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the tree.
- . Able to handle both numerical and categorical data. Other techniques are usually specialized in analyzing datasets that have only one type of variable. See algorithms for more information.
- . Able to handle multi-output problems.
- . Uses a white box model. If a given situation is observable in a model, the explanation for the condition is easily explained by boolean logic. By contrast, in a black box model (e.g., in an artificial neural network), results may be more difficult to interpret.
- . Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model.
- . Performs well even if its assumptions are somewhat violated by the true model from which the data were generated.

Weakness:

- . Decision-tree learners can create over-complex trees that do not generalize the data well. This is called overfitting. Mechanisms such as pruning (not currently supported), setting the minimum number of samples required at a leaf node or setting the maximum depth of the tree are necessary to avoid this problem.
- . Decision trees can be unstable because small variations in the data might result in a completely different tree being generated. This problem is mitigated by using decision trees within an ensemble. The problem of learning an optimal decision tree is known to be NP-complete under several aspects of optimality and even for simple concepts. Consequently, practical decision-tree learning algorithms are based on heuristic algorithms such as the greedy algorithm where locally optimal decisions are made at each node. Such algorithms cannot guarantee to return the globally optimal decision tree. This can be mitigated by training multiple trees in an ensemble learner, where the features and samples are randomly sampled with replacement.
- . There are concepts that are hard to learn because decision trees do not express them easily, such as XOR, parity or multiplexer problems.
- . Decision tree learners create biased trees if some classes dominate. It is therefore recommended to balance the dataset prior to fitting with the decision tree.

Since the number of features are small and easy to understand, few data preparation is needed, I think Decision Tree classification is suitable and easily to implement. DTs (Decision Trees) can be a good candidate to classify attrition.

(3) AdaBoost Ensemble learning

Real-world application in industry can be: Basketball player detection. (Reference: https://www.uni-obuda.hu/journal/Markoski_Ivankovic_Ratgeber_Pecce_Glusac_57.pdf)

Strength: (Reference: http://user.ceng.metu.edu.tr/~tcan/ceng734_f1112/Schedule/adaboost.pdf)

- . Very simple to implement
- . Does feature selection resulting in relatively simple classifier . Fairly good generalization

Weakness: (Reference: <http://www.nickgillian.com/wiki/pmwiki.php/GRT/AdaBoost#Advantages>)

AdaBoost can be sensitive to noisy data and outliers. In some problems, however, it can be less susceptible to the overfitting problem than most learning algorithms.

Since the dataset is large and AdaBoost ensemble method is fast, it will be able to perform multiple training iterations to maximize accuracy.

To properly evaluate the performance of each model, a training and predicting pipeline was created that allows effectively train models using various sizes of training data and perform predictions on the testing data. F-beta score function was adopted as major metric of performance and beta value 0.5 was set to the metric function because in the attrition prediction precision is supposed more important than recall.

Refinement

Based on the performance metric above, basically the model with highest F-score can be chosen as best model. Other factor such as training time can be also considered when choosing best model.

Once the best model was decided, the model can be fine-tuned with grid search cross validation (GridSearchCV) approach with important parameters. With this grid search approach, data fitting on a dataset all the possible combinations of parameter values are evaluated and then the best combination is retained. In this case, if the model with decision-tree methodology is the best, "max_depth" of decision-tree which is an important feature to control the size of the trees was used to fine-tune the model.

After model refinement, I will also train it on the same training set with only top important features whose accumulated normalized weighting is larger than 50% which means these most important features contribute more than half of the importance of all features present in the data. We can reduce the feature space and simplify the information required for the model to learn. We may also expect that the training and prediction time is much lower.

IV. Results

Model Evaluation and Validation

In the model evaluation, the number of records was selected with 1%, 10% and 100% of the training data respectively, that is 119, 1199 and 11999 rows. And three models (AdaBoostClassifier, Support Vector Classifier and Decision-tree classifier) were adopted to train on the testing data and predict on the testing data respectively with default settings for each model.

The performance metrics and result for these tree supervised learning models were as follows.

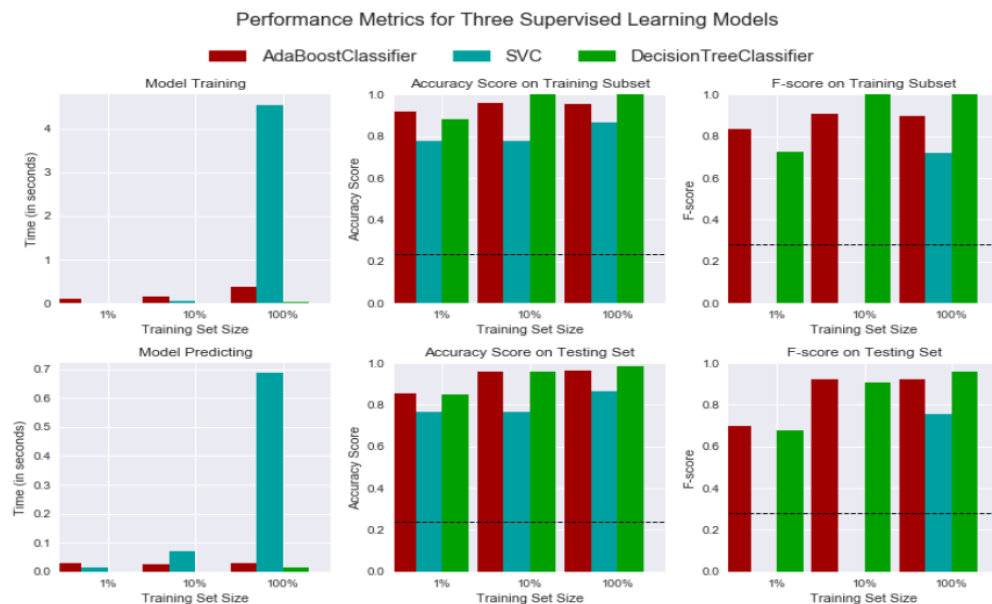


Illustration 7: Performance Metrics for models

(1) 1% of the training data:

Samples_119	Training time	Accuracy	F-score
AdaBoost	0.09	0.85	0.7

SVC	0	0.77	0
Decision-tree	0	0.85	0.68

(2) 10% of the training data:

Samples_1199	Training time	Accuracy	F-score
AdaBoost	0.15	0.96	0.92
SVC	0.05	0.77	0
Decision-tree	0.01	0.96	0.91

(3) 100% of the training data:

Samples_11999	Training time	Accuracy	F-score
AdaBoost	0.39	0.96	0.92
SVC	4.55	0.87	0.76
Decision-tree	0.03	0.98	0.96

Based on the result of F-score on the testing when 100% of the training data is used, both the F-score and accuracy of Decision-tree methodology are higher than other two methods. Besides, from the training time, Decision-tree (0.03 sec) is much faster than SVC (4.55 sec) and a slightly faster than AdaBoost (0.39 sec), considering the prediction and training time, Decision-tree classifier seems be the best model.

Based on the performance metric above, decision-tree classifier is the best model, then the model can be fine-tuned with grid search cross validation (GridSearchCV) approach with important parameters “max_depth”. In this case, “max_depth: 7” of decision-tree is the best parameter to generate the optimized model, and the F-score of optimized model is **0.968** which is better than unoptimized model which is 0.956.

Metric	Unoptimized Model	Optimized Model
F-score	0.956	0.968
Accuracy Score	0.982	0.978

Illustration 8: F score of optimized model

Feature selection for model with reduced data

Top five important features illustrated below are “satisfaction_level”, “average_monthly_hours”, “number_project”, “time_spend_company” and “last_evaluation”. The accumulated normalized weights of top three is larger than 0.5 which means these three most important features contribute more than half of the importance of all features present in the data.

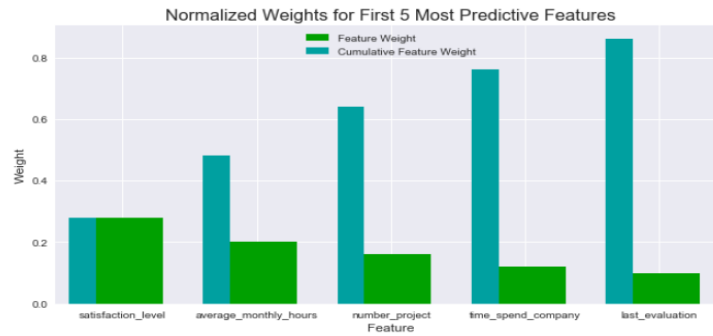


Illustration 9: Normalized weights for top 5 most predictive features

The F-score of model trained on full data and reduced data (top three important features) was 0.968 and 0.878.

Metric	Model trained on full data	Model trained on reduced data
F-score	0.968	0.878
Accuracy Score	0.978	0.937

Illustration 10: F-score of model trained on full data vs. reduced data

Justification

The optimized model's F-score is 0.968 and accuracy is 0.978. These scores are better than the unoptimized model whose F-score is 0.956 and accuracy is. The naive predictor's F-score is 0.28 and accuracy score is 0.238. The optimized model's scores are much better than naive predictor's whose F- score is 0.2809 and accuracy score is 0.2381.

The F-score of model trained on reduced data is 0.878, lower than the full data but I think it is acceptable if the training time is a concern. Besides, we may know these top three important features “satisfaction_level”, “average_monthly_hours”, “number_project” contribute half importance of all features present in the attrition data.

V. Conclusion

Free-Form Visualization

From the F-score metrics, the model with decision tree classifier has great performance of prediction for the attrition dataset. The training time is relatively smaller than other two algorithms. Since the number of features are small and easy to understand, few data preparation is needed, I think Decision Tree classification is suitable for such kind of attrition prediction for human resource management.

From the feature selection for model with reduced data, both F-score and accuracy on the reduced data using only three features are a little lower than those when all features are used. If training time was a factor, I will consider using the reduced data as my training set to make a trade-off between accuracy and training time. From the accumulated normalized weights of features we may know these top three important features

“satisfaction_level”, “average_monthly_hours”, “number_project” contribute half importance of all features present in the attrition data. Human resource professionals may consider to provide compensation program to those valuable employee with high attrition risk if they have low satisfaction level and high average monthly working hours or relatively high number of evolved projects.



Illustration 11: Accumulated normalized weights for top 3 most predictive features

Reflection

The main process used for this project can be summarized using the following steps: 1. Data exploration, 2. Data preparation 3. Evaluating model performance, 4. Create machine-learning model, 5. Improve machine-learning model and 6. Feature importance discussion.

I think in step 5 “improve machine-learning model” is the most interesting part. Intuitively I thought AdaBoost method will have best F-Score but in fact the decision-tree has best performance from the result of step 4. The F-score is high enough (0.956) which is beyond my expectation, but after refining the model in step 5 even higher F-score (0.968) is got with the grid search cross-validation approach. Data may tell us the fact, the truth and it is beyond human's intuition.

Improvement

In this attrition study, the final F-score of best model Decision-tree is high with input dataset from kaggle but I think in real-world the human resource related data maybe more complicated or even non-structured, there should rooms to improve for attrition prediction.

From the top five or top three important features, we may know the key factors are “satisfaction_level” and work loading (“average_monthly_hours” and “number_project”), if we have more detailed information about these features, we may generate more features to generate accuracy models. For example, “satisfaction_level” may be extended to the company, satisfaction to the work, satisfaction to the facility, satisfaction to the salary or compensation; work loading may be extended to such as involved project type, cooperation department, cooperation colleague. With these extended features, human resource team may consider to provide compensation program or improvement to those valuable employees to resolve their concerns. If human resource team can collect the attrition data before and after deploy these compensation program of improvement, then attrition prediction model can be expanded to relate to compensation and improvement program.

Basically the attrition prediction is a supervised binary classification problem, if the number of column is large and hard to interpret by human we may use unsupervised learning method, such as K-means clustering, Principal component analysis (PCA) to find the inside information for attrition prediction for human resource management.

VI. References

[1] Rohit Punnoose, Pankaj Ajit , BITS Pilani, "Prediction of Employee Turnover in Organizations using Machine Learning Algorithms." <https://pdfs.semanticscholar.org/fa49/19810eaaa67e851ad13775b78c94217a7908.pdf>