

Machine Learning Engineer Nanodegree

Capstone Project

Ravi Mandava
September 21th, 2018

Definition

Project Overview

The project is to apply machine learning algorithms in ERP (Enterprise Resource Planning) domain for a study on “Employee Attrition Prediction and Analysis”. For this requirement, the attempt is to build a model with Supervised Learning algorithms that helps to predict the outcome and analyze the results with the available dataset. This helps human resources managers to give an insight for identifying risks and hence do a better planning.

Domain Background

Enterprise resource planning (ERP) refers to the integrated management of core business processes and automation. The challenge, however, lies in managing this database to get the best use out of the information held within.

Machine learning is all about automating processes that would take more time or difficult to get analytics on dataset with manual methods. The main idea is to organize, transform the available data and provide input to supervised learning algorithms to predict outcome and measure the accuracy. Supervised learning is a type of machine learning algorithm that uses a known dataset (called the training dataset with labeled data) to make predictions. As my current working domain is ERP, it has motivated to apply machine learning techniques in ERP to see the opportunities where business process can be streamlined.

Problem Statement

Employee attrition is a big problem many companies are facing nowadays. On boarding a new employee can become expensive, resources consuming in terms of time, money and sometimes may not even find a perfect replacement. Considering these factors, some of the companies are trying to keep their employees happy and satisfied. One of the ways how to keep employees from leaving is to analyze, why people who left the company decided to do so, predict who could be leaving next and try to take actions for retention. The main objective is to solve the problem by determining the factors that lead to attrition, predict the likeliness of attrition and hence reduce the employee attrition based on results

Metrics

The target class ‘attrition’ should predict correctly with different input features. Accuracy from confusion matrix is an appropriate metric to compare the predicted positive and negative results against true values. The accuracy (AC) is the proportion of the total number of predictions that were correct. It is determined using the equation:

- TN is the number of correct predictions that an instance is negative,
- FP is the number of incorrect predictions that an instance is positive,
- FN is the number of incorrect of predictions that an instance negative, and
- TP is the number of correct predictions that an instance is positive.

Confusion Matrix	Actual Positive	Actual Negative
Predicted Positive	TP	FP
Predicted Negative	FN	TN
accuracy = (TP + FP)/(TP+FP+FN+TN)		

The metric is used when evaluating the model because false negatives and false positives both will have an impact for decision making and alternative planning.

Analysis

Data Exploration

The dataset contains a csv flat file with the HR dataset, containing basic information about the employees (age, daily rate income, years of service, job role, job satisfaction rating, commute distance etc.) and whether employee left or stayed with the company. The dataset contains 1470 rows and 35 feature columns with primary feature Attrition (Yes/No). Since this dataset is relatively small, the plan is to make use 80 percent of data for training set and remaining 20 percent data for validation.

The source for dataset is [HR Employee Attrition and Performance](#).

Below is the snippet of sample dataset.

	EmployeeNumber	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	...	RelationshipS:
0	1	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	...	
1	2	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	...	
2	4	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	...	
3	5	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	...	
4	7	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	...	

5 rows × 35 columns

Identifying the key input features is very important to predict the results with higher accuracy rate. In the dataset chosen, features like employee count, gender may not have much impact on attrition rate. More information can be derived through exploratory data analysis; therefore, it is better to avoid these features in the input dataset to reduce model training time.

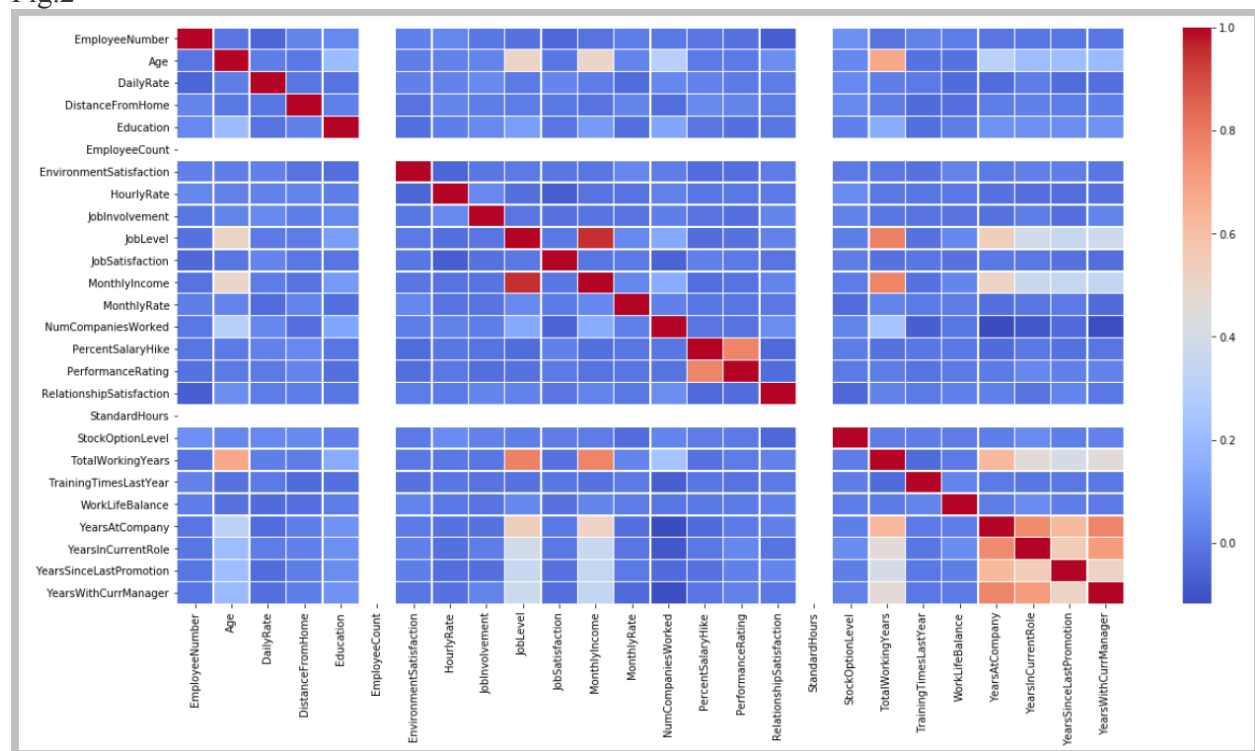
From the sample dataset, the target Class 'Attrition' is distributed mainly on input features like job role, years of service, working environment, manger relationship, overtime. The data is already preprocessed or captured with rating scales, the percentage of outliers is less and hence don't expect imbalance in classification

Exploratory Visualization

Exploring and visualizing data is important to validate the data with the noted assumptions and identify any anomalies in data. This helps to avoid feeding wrong data to the machine learning model and interpret model output to test its assumptions.

The following is a correlation plot that shows relation between the attributes.

Fig.2



The relevance between the attributes can be visualized based on color map scale. Color coding with scale value 1 has high dependency and reduces as scale ranges down from 1 to 0.

Fig.3

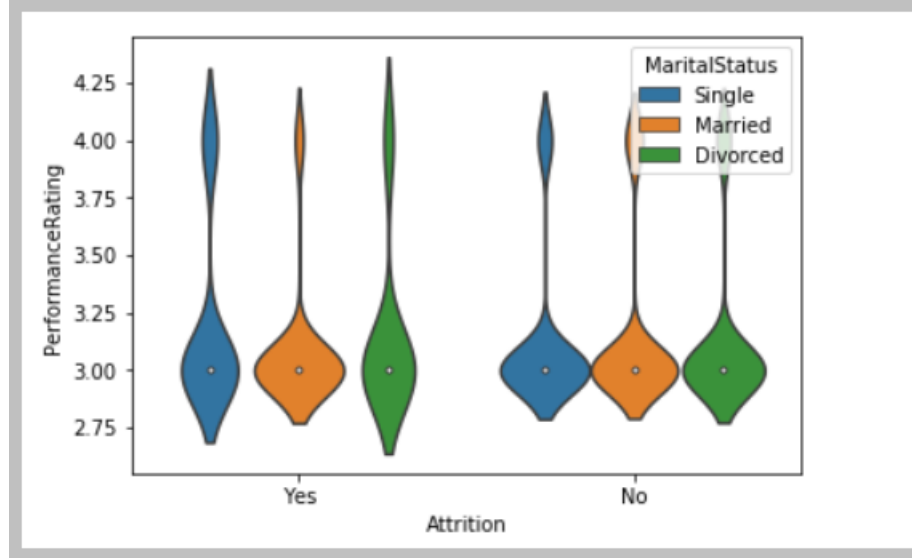


Fig3 is a density visualization plot against Attribution and Performance rating cued with Marital status.

Attrition is more likely to depend on combination of attributes rather than a single attribute. From the above visualization plot, we can easily come to an assumption that there is no relation with Employee Count, Standard Hours. These attributes can be excluded from the dataset set as part of data pre-processing.

Algorithms and Techniques

In this phase we identify the machine learning method that can applied for a dataset that suits well depending on the problem. We can choose different models **Gaussian Naive Bayes, Decision Tree, Support Vector Machine, Random Forest Classifiers** which showed good results for the analysis.

Based on the metrics the parameters in the Machine Learning models will be optimized to predict results consistently with a defined accuracy. For example, optional parameters like ($n_estimators=25$, $n_jobs=2$) are used on Random Forest Classifier that determines the initial split which has an impact on the model predictions.

Further we can build a model using Neural Networks to decide the output. This is an improvement approach that can optimized the results of our initial method. However, neural network model was not implemented for this report.

Benchmark

Benchmarking the operational characteristics is as important as evaluating the predictive characteristics of a model. The sample dataset of employee attrition contains both regression features (years of service, age, monthly rate) and classification features (job role,

education and department). A simple algorithm **Decision Tree** is chosen as benchmark model. The results of this model can be compared to understand how well the other models work.

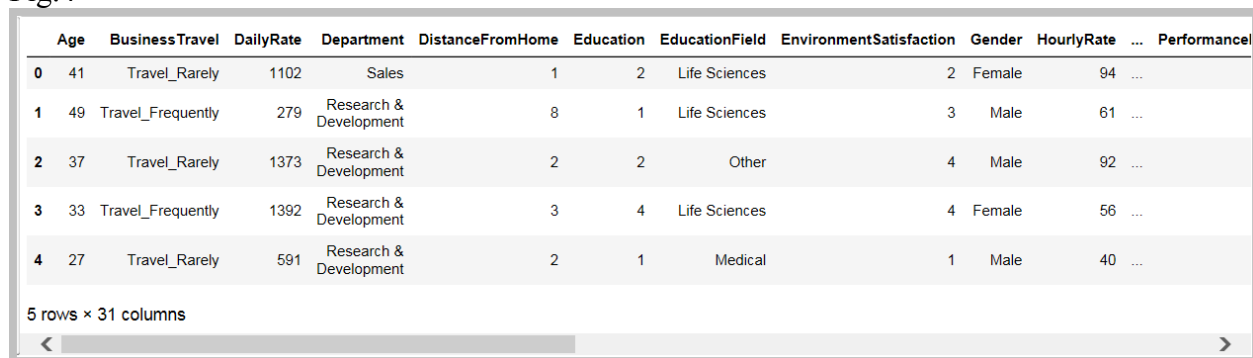
Methodology

Data Preprocessing

Before data can be used as input for machine learning algorithms, it often must be cleaned, formatted, and restructured typically known as **preprocessing**. Fortunately, for Employee Attrition dataset, there are no invalid or missing entries we must deal with, however, there are some independent features Employee Number, Employee Count, Standard Hours must be adjusted to exclude from the training data and other features can be scaled to improve the training time. This preprocessing can help tremendously with the outcome and predictive power of nearly all learning algorithms.

The dataset is now reduced to 31 attributes from initial count of 34. That means algorithm doesn't need any time to consider this data for prediction.

Fig.4



	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	Gender	HourlyRate	...	Performance
0	41	Travel_Rarely	1102	Sales	1	2	Life Sciences	2	Female	94	...	
1	49	Travel_Frequently	279	Research & Development	8	1	Life Sciences	3	Male	61	...	
2	37	Travel_Rarely	1373	Research & Development	2	2	Other	4	Male	92	...	
3	33	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	4	Female	56	...	
4	27	Travel_Rarely	591	Research & Development	2	1	Medical	1	Male	40	...	

5 rows × 31 columns

In addition to performing transformations on features that are highly skewed, it is often good practice to perform some type of scaling on numerical features. Applying a scaling to the data does not change the shape of each feature's distribution (such as Age, Daily Rate, Distance from Home, Hourly Rate, Monthly Income, Monthly Rate); however, normalization ensures that each feature is treated equally when applying supervised learners. Note that once scaling is applied, observing the data in its raw form will no longer have the same original meaning, as exemplified below.

Fig.5 Data after Min Max scaling on Numerical Features

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	Gender	HourlyRate	...	Perform:
0	0.547619	Travel_Rarely	0.715820	Sales	0.000000	2	Life Sciences	2	Female	0.914286	...	
1	0.738095	Travel_Frequently	0.126700	Research & Development	0.250000	1	Life Sciences	3	Male	0.442857	...	
2	0.452381	Travel_Rarely	0.909807	Research & Development	0.035714	2	Other	4	Male	0.885714	...	
3	0.357143	Travel_Frequently	0.923407	Research & Development	0.071429	4	Life Sciences	4	Female	0.371429	...	
4	0.214286	Travel_Rarely	0.350036	Research & Development	0.035714	1	Medical	1	Male	0.142857	...	

5 rows × 31 columns

From the table in **Exploring the Data** above, we can see there are several features for each record that are non-numeric. Typically, learning algorithms expect input to be numeric, which requires that non-numeric features (called *categorical variables*) be converted. One popular way to convert categorical variables is by using the **one-hot encoding** scheme. So, the above dataset is further processed to convert categorical features into numerical features by transposing the data as dummy columns. This processing alters the dataset view with now columns while rows remains the same. The new dataset now has shape of 5 rows and 52 columns.

Finally, this dataset is passed as an input to machine learning algorithms to predict the output.

Implementation

The implementation process can be split into 2 main steps:

1. Split the data into training and testing datasets
2. Train the model with the training dataset and measure the prediction accuracy

During the first step, preprocessed data is split into training dataset. 80% of the data is split and feed to the model as training data and remaining 20% is used for evaluation.

The most common solutions for labeled datasets can be derived by applying different Supervised Learning algorithms like Random Forest, Decision Tree, Support Vector Machine and Naive Bayes etc. To determine the Employee Attrition rate with the available feature dataset, above algorithms are implemented and derived the employee attrition rate and top factors that influence this.

The models are initialized with default parameters and a random state.

1. GaussianNB(priors=None)
2. DecisionTreeClassifier(random_state=10)
3. SVC(random_state=10)
4. RandomForestClassifier(random_state=10)

SNO	Model / Classifier	Accuracy
1	Gaussian NB	73.81%
2	Decision Tree Classifier	79.93%
3	SVC	82.99%
4	Random Forest Classifier	84.35%

From the above table, we observe Random Forest Classifier predictions are more accurate compare to other classifiers. In the next steps, this model can be chosen as a baseline for refinement. Also, the accuracy is more than the benchmark model Decision Tree Classifier identified earlier.

Refinement

As mentioned earlier Random Forest Classifier can be optimized by tuning the parameters to improve the accuracy. The minimum expectation is our model should perform at least better than a benchmark model.

Now, the best model is picked and parameters are optimized to tune the model.

RandomForestClassifier(min_samples_split=4,n_estimators=30,max_depth=10,n_jobs=2,random_state=10)

Fig.6

```
# TODO: Import the three supervised learning models from sklearn
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.metrics import make_scorer
from sklearn.metrics import fbeta_score
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix

# TODO: Initialize the three models
clf_A = GaussianNB(priors=None)
clf_B = DecisionTreeClassifier(random_state=10)
clf_C = SVC(random_state=10)
clf_D = RandomForestClassifier(min_samples_split=4,n_estimators=30,max_depth=10,n_jobs=2,random_state=10)
#clf_D = RandomForestClassifier(random_state=10)

Clist=['GaussianNB','DecisionTreeClassifier','SVC','RandomForestClassifier']
# TODO: Make an fbeta_score scoring object using make_scorer()
scorer = make_scorer(fbeta_score, beta=0.1)
i=0
for clf in [clf_A, clf_B, clf_C, clf_D]:
    predictions = (clf.fit(X_train, y_train)).predict(X_test)
    #print((np.array(y_test)).np.asarray(y_test))
    print("Model {}".format(Clist[i])) |
    print("Accuracy score on testing data: {:.4f}".format(accuracy_score(y_test, predictions)))
    print("Model Confusion Matrix on testing data: {}".format(confusion_matrix(np.asarray(y_test),np.asarray(predictions))))
    print("F-score on testing data: {:.4f}".format(fbeta_score(y_test, predictions, beta = 0.5)))
    i+=1

# Run metrics visualization for the three supervised learning models chosen
#vs.evaluate(results, accuracy, fscore)
```

This refinement to model has improved the accuracy from 84.35% to 86.05%.

Results

Model Evaluation and Validation

Dataset was split into training and testing sets during the development to evaluate the model. A complete description about the parameters used in the model is shown in Fig.6.

The following code is written to evaluate the model performance

```
for clf in [clf_A, clf_B, clf_C, clf_D]:
    predictions = (clf.fit(X_train, y_train)).predict(X_test)
    #print((np.array(y_test)).np.asarray(y_test))
    print("Model {}".format(Clist[i]))
    print("Accuracy score on testing data: {:.4f}".format(accuracy_score(y_test, predictions)))
    print("Model Confusion Matrix on testing data:
    {}".format(confusion_matrix(np.asarray(y_test),np.asarray(predictions))))
    #print("F-score on testing data: {:.4f}".format(fbeta_score(y_test, predictions, beta = 0.5)))
    i+=1
```

Below table represents metrics corresponding to each model.

Fig.7

SNO	Model / Classifier	Accuracy	Confusion Matrix
1	Gaussian NB	73.81%	True Positive 175 False Negative 64 False Positive 13 True Negative 42
2	Decision Tree Classifier	79.93%	True Positive 206 False Negative 33 False Positive 26 True Negative 29
3	SVC	82.99%	True Positive 237 False Negative 2 False Positive 48 True Negative 7
4	Random Forest Classifier	86.05%	True Positive 237 False Negative 2 False Positive 39 True Negative 16

Our objective to pick a model that predicts the chance of employee moves from the company. So, the model should be consistent with precision value for output result 1 (which means attrition

Yes). Classification Report is run on the optimal model Random Forest from list of models training with the Employee Attrition Dataset.

	Precision	Recall	F1-score	Support
0	0.86	0.99	0.92	239
1	0.89	0.29	0.44	55
Avg / Total	0.86	0.86	0.83	294

From the precision and F1 score metrics above, we can see Random Forest performs better on training data.

Justification

The main reason for selection of this model is it working well for both statistical and classification problems. Our dataset has some numerical attributes like Hourly Rate, Years of Service, Distance from Home and categorical features like Performance Rating, Job Level, Education. From the Fig.7, we can see the results of different models and the clear winner is Random Tree Forest ensemble method and hence the classifier is justified.

Conclusion

Free-Form Visualization

To better depict the solution results, top five important features of dataset that determines the Employee Attrition are identified.

Following code is used to extract import features for Random Forest Classifier

```
# TODO: Import a supervised learning model that has 'feature_importances_'
from sklearn.ensemble import RandomForestClassifier
import visuals as vs

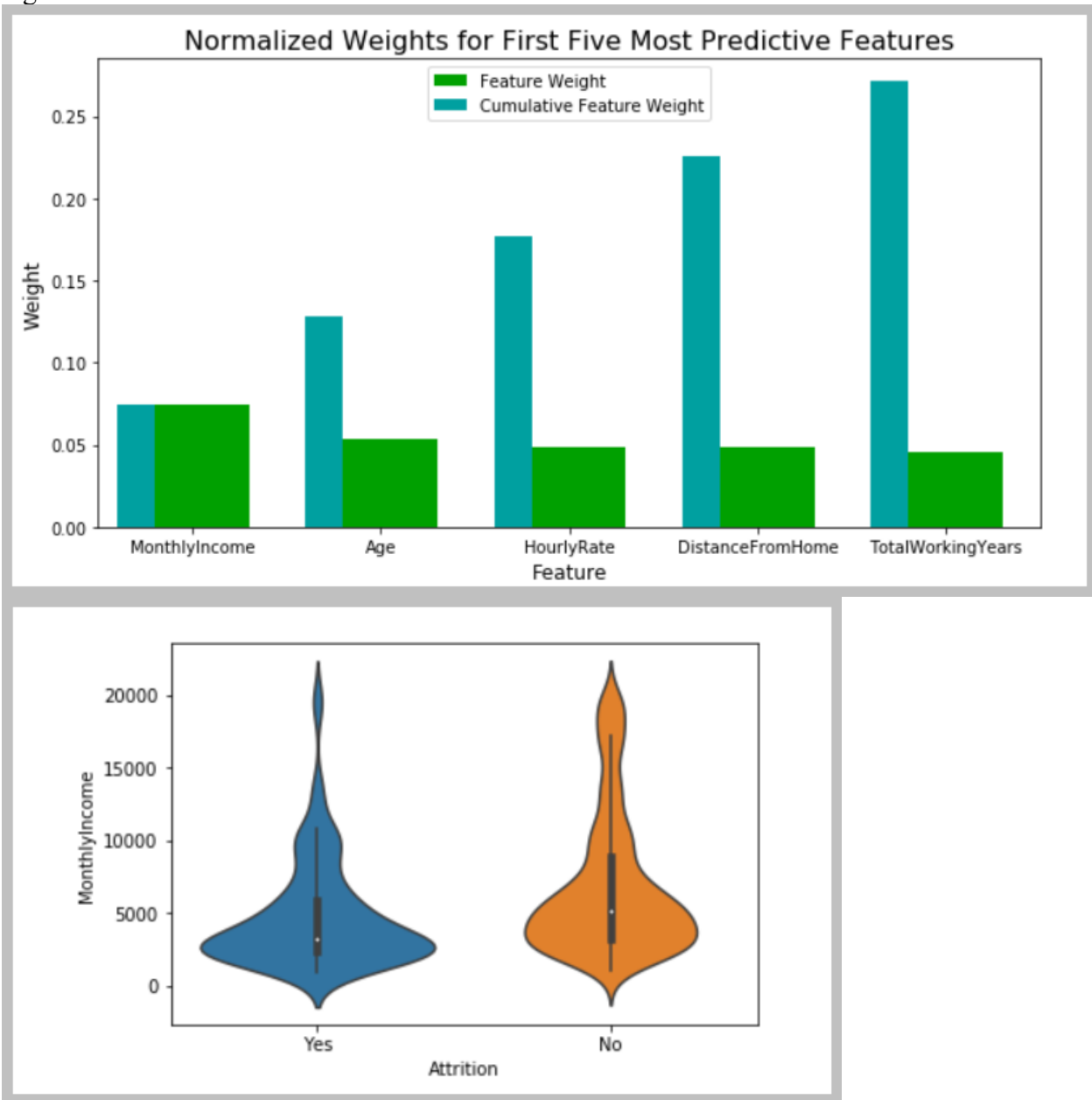
model =
RandomForestClassifier(min_samples_split=4,n_estimators=30,max_depth=10,n_jobs=2,random_state=10)

# TODO: Extract the feature importances using .feature_importances_
```

```
importances = model.fit(X_train,y_train).feature_importances_  
#print(importances)
```

```
# Plot  
vs.feature_plot(importances, X_train, y_train)
```

Fig.8



Density of datapoints is more for montly income between 0-2500 plotted against Attrition.

Features that contribute to Employee Attrition

1. Monthly Income:
Employees with less salaries look for better opportunities.
2. Age:
Younger people will look for new opportunities as their learning curve is more when compared to seniors
3. Hourly Rate:
Employees with low Hourly Rate look for better opportunities.
4. Distance from Home:
Employees who travel more for office tend to look for jobs near to home.
5. Total Working Years:
Employees with less working years' experience look for new opportunities.

From these observations, we can suggest few pointers to reduce the attrition
Employers should periodically evaluate competitive market rate and do a salary normalization.
Managers should encourage employees to take high roles for job satisfaction.

Reflection

The procedure used for this project can be summarized using following steps:

- 1) ***Feature Extraction, Selection and Transformation:***
In this step we extract the important, non-redundant features from raw data and try to get most important features that are contributing to decide the label. The data is transformed, preprocessed or scaled to reduce the outliers.
- 2) ***Perform Exploratory Data Analysis:***
In this step the dataset is analyzed to summarize the main characteristics, often with visual methods (Pair Plot, Heat Map, Violin Plot). This is helpful when we have a dataset with many features dataset and an optional step for small datasets.
- 3) ***Build a predictive model with different techniques:***
In this step the models like Decision Tree, Random Forest, and Naive Bayes are built by tuning appropriate parameters with common dataset.
- 4) ***Identify the optimal model through testing and evaluation techniques:***
In this step the results from each model are passed to the evaluation metrics like accuracy and F1score and identify the optimal model.
- 5) ***Identify the top features that has more weight to predict the result:***
In the step based on the EDA (Exploratory Data Analysis) and results from models, factors that have high influence to determine the outputs are rated.

The difficult part is understanding the dataset and modelling the data for visualization. I spend more time on data visualization techniques (Heat Map plot from Seaborn library is used to plot the correlation among the features).

Improvement

There is always a scope for improvement to further optimize the classifiers and have the accuracy increased. We can also build a model using Neural Networks to predict the output. It uses back propagation to continuously update the weights for consistent predictions.

Another improvement we can make is to build a hybrid model with a combination of Supervised Learning methods and Neural Network. The output from 4 supervised models can be fed into a neural network and the network determines the result.

References:

<https://towardsdatascience.com/solving-staff-attrition-with-data-3f09af2694cd>

<https://www.ibm.com/communities/analytics/watson-analytics-blog/hr-employee-attrition/>

<https://www.quora.com/How-can-we-use-machine-learning-in-ERP>

https://en.wikipedia.org/wiki/Supervised_learning

http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html