*2021 6th International Conference for Convergence in Technology (I2CT)*
*Pune, India. Apr 02-04, 2021*

# Machine Learning Model to Predict Work Force Attrition

Priyanka Sadana
*Innovative Business Solutions*
*SAP Labs India Pvt. Ltd.*
Gurugram, India
priyanka.sadana@sap.com

Divya Munnuru
*Innovative Business Solutions*
*SAP Labs India Pvt. Ltd.*
Gurugram, India
divya.munnuru@sap.com

***Abstract* – Employee attrition problem is becoming worse day by day in many tech firms across the globe. It has been observed that the employee attrition (churn) rate is increasing and is on a higher side in IT firms due to the unforeseen reasons of quitting or switching jobs frequently. This is applicable from small scale to large scale industries. Cost of hiring a new employee is way more than retaining an existing one. Attaining a new resource requires training, knowledge transfer, time for getting comfortable with new company's or team's environment. Critical resources who are familiar with the technologies, platforms and environment of the company create a huge impact on the customer success and if such resources leave then it becomes a loss for the company, be it a short term loss but it multiplies if the number of attrition is huge. This leads to escalations from customer as well. This paper explains you in detail on how to address this problem across the IT firms using Machine Learning Model. This model predicts whether a resource is about to leave or not in near future and predicts the possible reason behind leaving. Notifying this information to Human Resource team will help them understand the list of resources who are not satisfied with the work, office environment, work life balance, opportunities, job location, etc. and to take on time action to retain them.**

***Index Terms* – resource, IT firms, attrition rate, predictor variable, target variable, work life balance, rewards, logistic regression, random forest, decision tree, workforce.**

## I. INTRODUCTION

Let us start by explaining the idea behind this application. In recent years a huge increase in employee attrition rate and its impact on deliverables has been observed in IT Product & Service firms. We did the Literature Survey and found that Forbes reports 64% of Employees may leave their jobs in 2020 and Positive Feedback & Recognition might be the magic bullet to retain them. Achievers found that 82% employees "strongly" or "somewhat" agree they wished they received more recognition [1]. Gartner reports that organizations that effectively deliver their Employee Value Preposition can decrease the turnover by nearly 70% [2]. Also, it can be seen in the table that the voluntary employee attrition rate has increased or remained constant but did not decrease as per the fiscal year reports from organizations like Infosys [3] & [4], Accenture [5], Wipro [6], TCS [7].

| Company | Attrition % in 2018 | Attrition % in 2019 |
|---|---|---|
| Infosys | 23 | 20 |
| Accenture (Q1, Q2, Q3, Q4) | 13, 17, 18, 15 | 15, 18, 19, 14 |
| Wipro | 17.7 | 17 |
| TCS | 10.9 | 11.5 |

There can be two cases possible when a critical resource decides to leave a company, 1) An alternate resource is available in or across team with same skillset 2) A replacement is not available and is required to be hired. In case 1, a handover is planned in the notice period of the resource who is about to leave. In case 2, cost is higher and requires training and knowledge transfer for the new resource. Having same skillset is not enough in an ongoing and critical project. Critical resources are very important for the customer as they know the project functionally, technically and understand the customer's requirement and the customer's comfort zone. Many a times, the stage of the project does not permit the new resource's onboarding at a calm pace and the need of the hour becomes to immediately jump into development and support. Deploying a new resource might pacify the problem but doesn't solve it completely. Higher rate of attrition is not just limited to development area but generates a prominent effect in sales, R&D, Quality, Consulting, User Experience departments as well.

Here we are not only losing the employee rather the customer's trust and base too. Whenever company loses a customer it needs to spend an additional cost on marketing and staffing to acquire new one. Employee decides to leave an organization because of one of the following reasons or more.

- More Salary or better Role offered outside
- Not suitable Working Environment
- Don't feel aligned with company goals
- Work Life Balance missing & stress
- Relocation & Higher Education
- Not feeling appreciated or feeling underutilized
- Seeing good employees leave

## II. SOLUTION APPROACH

Retaining the workforce has become a major issue for many IT firms today. Many companies are trying to figure out various ways to retain their workforce like generating yearly people survey reports and finding concerns of employees. In this paper, we are applying Machine Learning algorithm on the data collected from both alumni and current employees of a company and creating a model. This can be used by the HR team to know the reasons behind attrition which are specific to their firm and will give an early indication and time to retain before it's late. Machine Learning algorithm makes data driven decisions easy and quick because it can achieve great accuracy when trained on the data and the model implemented can each

2021 6th International Conference for Convergence in Technology (I2CT) | 978-1-7281-8876-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/I2CT51068.2021.9418140

year be retrained and used by the HR via a simple API exposed to the UI. Below is the flowchart explaining the solution approach

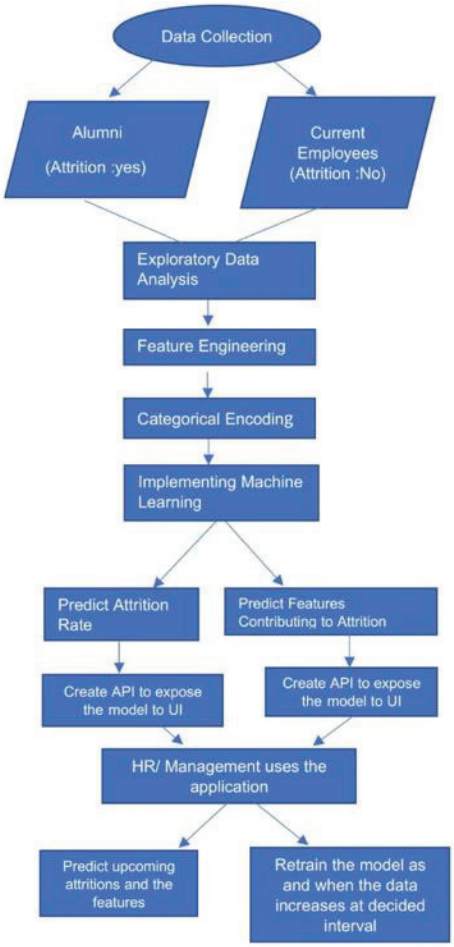


Fig. 1. Flowchart of complete Solution Approach

This approach reduces the manual work of creating reports, performing analysis and it never goes outdated as new data for retraining gets available every year. In further sections, we will explain each step followed to implement this solution.

### *A. Data Collection*

In order to apply Machine Learning algorithm to derive the reasons behind leaving and to predict the attrition, we needed good quality dataset. Therefore, we prepared two surveys, one for the existing employees [8] and another one for the alumni of an IT company [9]. Further, in this paper we will call this company as 'AXZ'. The questions were similar in both the surveys with a difference of present and past tense in text. The survey had following questions:

- Date of Birth, Gender, Marital Status
- Experience with Management & Role (on a scale of 1 to 30) at company AXZ: 1) Years with company AXZ, 2) Years with your current manager, 3) Years since last promotion, 4) Years in current role
- Experience (on a scale of 1 to 30): 1) Total Experience (Years), 2) Number of Companies you have worked with till today
- Which department you work for? What is your Job Title? What is the business Travel Frequency? Do you usually work overtime? What is your daily travel time (in minutes) from home to office and back home?
- Rate your current job based on 1) Working Environment, 2) Involvement in job, 3) Work life balance, 4) Annual Rewards, 5) learning opportunity & Skill Development, 6) Job Location, Feeling Appreciated

The above-mentioned questions became the predictor variables of the dataset. The target variable is "Attrition" with values Yes and No. "Yes" for the alumni who filled the survey 2 and "No" for the current employees who filled the survey 1. In total we received 528 responses from alumni and 1121 responses from current employees. We performed a union of both the datasets after making common column names for example, "YearsWithCurrentManager" was a column for current employees and "YearsWithLastManager" was a column for alumni and we combined the data by renaming both the columns to "YearsWithCurrentOrLastManager".

### *B. Data Cleaning*

- Check if there are null values in the data set
- Years with Manager, Years in Role, Years since last promotion should be less than or equal to Years with Company
- If Total Experience and Years with Company is same, then Total No. of companies should be 1
- Convert the value in the features to same tense for example, Travel Rarely was an option for current employees whereas Traveled Rarely was for alumni. We changed Traveled Rarely to Travel Rarely in the combined dataset.

### *C. Exploratory Data Analysis*

**Predictor Features Vs Target Variable**

Let's find out the relationship between the predictor and target variable. These days a lot is talked about work life balance and it has become a major driving force for people to change their workplace. Now, more and more people want to spend time with their family and wish to pursue their hobbies as well along with their job. A concept of giving time to self is encouraged a lot these days.


Fig. 2. Work Life balance Vs. Attrition

From the above graph, we can see that the attrition has decreased as the rating has increased from 3 to 4. The attrition still exists at a rating of 4 and 5 which shows that for alumni of company AXZ, work life balance was not the major reason to leave and the count of "No" at rating 4 is quite high which shows that this company provides a good work life balance and that becomes a driving force for workforce to stay with the company.

Let's observe Gender, Marital Status, Overtime and Business Travel Frequency's effect.

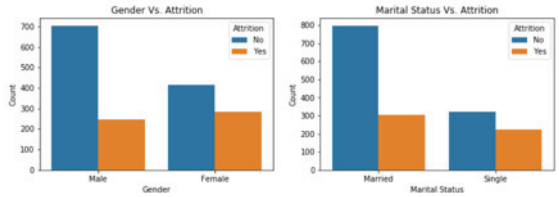


Fig. 3. Gender Vs. Attrition & Marital Status Vs. Attrition

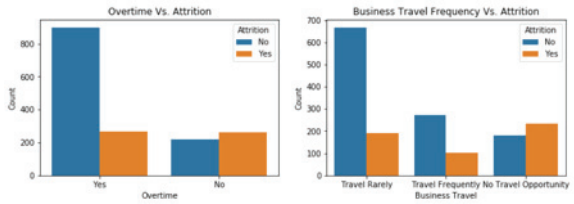


Fig. 4. Overtime Vs. Attrition & Business Travel Frequency Vs. Attrition

Attrition is observed more in Females when compared to males and in married people when compared to Single. Attrition exists irrespective of the overtime done or not. No or rare Travel opportunity surely seems to be a major reason of attrition.

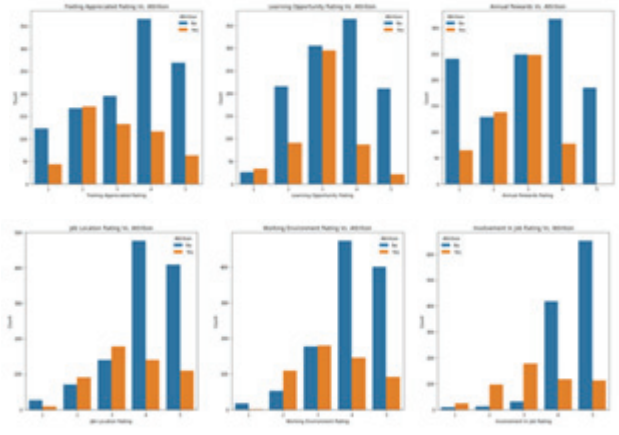

Fig. 5. Predictor Variables Vs. Attrition

The graphs clearly show that the attrition (number of "Yes") is higher where workforce has given a rating of 2 or 3 whereas when the rating increased to 4 or 5, the attrition has decreased. Attrition is almost 0 where the rating is 5 for Annual Rewards. From this dataset, it can be inferred that Annual Reward is a very important factor for the workforce to stay with a company and feeling appreciated, involved in job, working environment, job location and learning opportunities also play a vital role. The above analysis does not involve multiple predictor variables together to see the combined effect on attrition and therefore, the above inferences might change when multivariate analysis is performed.

Let's have a look at the correlation of features. From the heat map below drawn using .corr(), we can see that lots of columns seem to be poorly correlated to each other. It is preferred to train a model with features that are not too correlated so that we do not keep redundant variables. Here Age is highly correlated to Total Experience which is factual also.

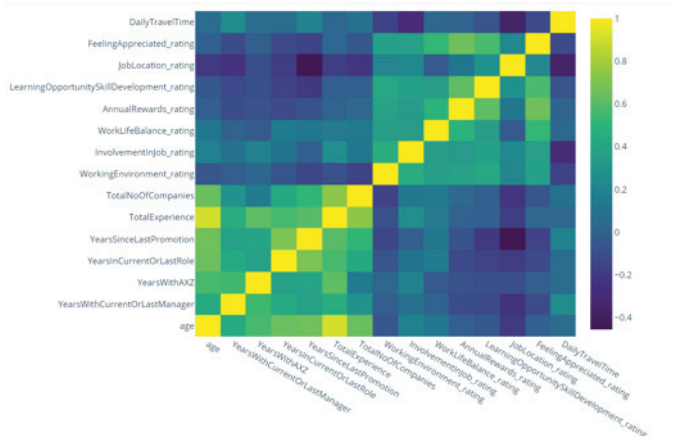


Fig. 6. Pearson Correlation of numerical features

## D. *Feature Engineering and Categorical Encoding*

Derive Age from date of birth and drop DateOfBirth. Numerically encode the Target Variable and drop Attrition.

```
target_map = {'Yes':1, 'No':0}
wr["Attrition_Binary"] = wr["Attrition"].apply(lambda x: target_map[x])
```

Fig. 7. Encode Target Variable

There is a mix of categorical and numerical data in the dataset. Many algorithms cannot handle the categorical variables, therefore use get_dummies method to numerically encode these variables.

```
wr_cat.columns
```

```
Index(['Gender', 'MaritalStatus', 'Department', 'JobTitle',
       'BusinessTravelFrequency', 'Overtime'],
      dtype='object')
```

Fig. 8. Categorical Columns

```
wr_cat = pd.get_dummies(wr_cat)
```

Fig. 9. Numerically encode categorical variables

Creation of a new column for each category in categorical variables led to 109 more columns. Let's drop the categorical variable and combine the newly created dummies with rest of the numerical columns resulting in total 125 columns.

## E. *Implementing Machine Learning Models – Predict Attrition*

### Splitting Data in Train and Test

```
from sklearn.model_selection import train_test_split
y = wr_final['Attrition_Binary']
X = wr_final.drop(['Attrition_Binary'],axis=1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
print("Number of records in X_train dataset: ", X_train.shape)
print("Number of records in y_train dataset: ", y_train.shape)
print("Number of records in X_test dataset: ", X_test.shape)
print("Number of records in y_test dataset: ", y_test.shape)
```

```
Number of records in X_train dataset:  (1154, 124)
Number of records in y_train dataset:  (1154,)
Number of records in X_test dataset:  (495, 124)
Number of records in y_test dataset:  (495,)
```

Fig. 10. Split dataset in train and test

32.6% of the training dataset is for Attrition "Yes/1" and rest is for Attrition "No/0" i.e. there is skewness in the target variable data.

```
# Percentage of attrition data in training data set
pec_attrition_data = sum(y_train==1)/(sum(y_train==1)+sum(y_train==0))
print(pec_attrition_data)
```

```
0.3266897746967071
```

Fig. 11. Skewness in the target variable data

## SMOTE to oversample

```
from imblearn.over_sampling import SMOTE
print("Before OverSampling, counts of label '1' in Pattern Dataset : {}".format(sum(y_train==1)))
print("Before OverSampling, counts of label '0' in Pattern Dataset: {} \n".format(sum(y_train==0)))
sm = SMOTE(random_state=2)
X_train_balanced, y_train_balanced = sm.fit_sample(X_train, y_train.ravel())

print('After OverSampling, the shape of train_X in Pattern: {}'.format(X_train_balanced.shape))
print('After OverSampling, the shape of train_y in Pattern: {} \n'.format(y_train_balanced.shape))
print("After OverSampling, counts of label '1' in Dataset: {}".format(sum(y_train_balanced== 1)))
print("After OverSampling, counts of label '0' in Dataset: {} \n".format(sum(y_train_balanced== 0)))
```

```
Before OverSampling, counts of label '1' in Pattern Dataset : 377
Before OverSampling, counts of label '0' in Pattern Dataset: 777

After OverSampling, the shape of train_X in Pattern: (1554, 124)
After OverSampling, the shape of train_y in Pattern: (1554,)

After OverSampling, counts of label '1' in Dataset: 777
After OverSampling, counts of label '0' in Dataset: 777
```

Fig. 12. SMOTE applied to oversample

Now the training data is balanced. Balance the test data also for future use.

## Principal Component Analysis

The dimensionality of the dataset at use increased after numerically encoding the categorical variables. PCA is a linear dimensionality reduction technique that is used for extracting information from a high-dimensional dataset by projecting it into lower-dimensional sub-space. It preserves the essential parts that have more variation of the data and removes the non-essential part with fewer variation. The result comes in form of eigen values.

```
from sklearn.decomposition import PCA
wr_pca = PCA(svd_solver='randomized', random_state=42)
wr_pca.fit(X_train_balanced)
colnames = list(X_train.columns)
wr_pca_df = pd.DataFrame({'PC1':wr_pca.components_[0],'PC2':wr_pca.components_[1], 'Feature':colnames})
wr_pca_df.head()
```

| | Feature | PC1 | PC2 |
|---|---|---|---|
| 0 | YearsWithCurrentOrLastManager | 0.005065 | 0.094806 |
| 1 | YearsWithAXZ | 0.002765 | 0.219410 |
| 2 | YearsInCurrentOrLastRole | 0.000534 | 0.129037 |
| 3 | YearsSinceLastPromotion | 0.004265 | 0.137010 |
| 4 | TotalExperience | 0.000979 | 0.614807 |

Fig. 13. Find PCA components

Let's have a look at the Scree plot to find the number of principal components needed. Scree plot is a line plot of eigenvalues of factors. It is used to determine the number of factors to retain in PCA in order to not lose any information. The line is getting linear around 25 components. Let's fit the model.

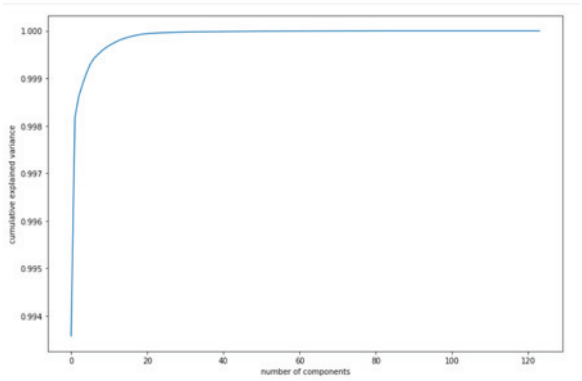


Fig. 14. Scree Plot

```
from sklearn.decomposition import IncrementalPCA
wr_pca_final = IncrementalPCA(n_components=25)
```

```
df_train_pca = wr_pca_final.fit_transform(X_train_balanced)
df_train_pca.shape
```

```
(1554, 25)
```

Fig. 15. Apply PCA

The correlation matrix for the principal components should show little to no correlation i.e. all 25 features are required.

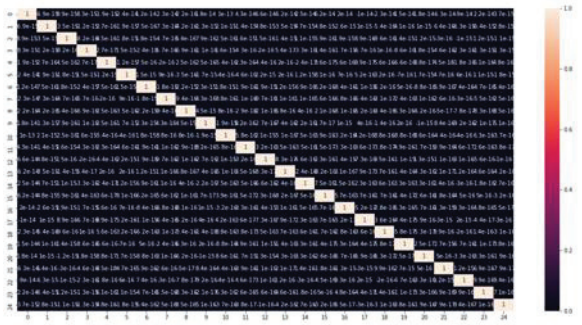

Fig. 16. Correlation matrix for principal components

and apply PCA on the balanced test data as well.

## Logistic Regression

It is a supervised regression method used for solving the binary classification problem. It computes the probability of an event occurrence. Train the model on the training data.

```
from sklearn.linear_model import LogisticRegression
from sklearn import metrics

lr = LogisticRegression(class_weight="balanced")
lr_model = lr.fit(df_train_pca, y_train_balanced)
#Making prediction on the test data
predictions = lr_model.predict_proba(df_test_pca)[:,1]
"{:2.2f}".format(metrics.roc_auc_score(y_test_balanced, predictions))
```

```
'0.93'
```

Fig. 17. Apply Logistic Regression

The roc auc score comes out to be 93%. ROC curve is an aggregated metric that evaluates how well logistic regression model classifies positive and negative outcomes at all possible cutoffs. While modelling we will calculate various metrics like recall, accuracy, precision, etc. Our focus will be Recall

(sensitivity) as we need to focus on correctly predicting attrition "Yes" and not attrition "No".

```
confusion = metrics.confusion_matrix( y_pred_final.Attrition, y_pred_final.predicted )
confusion
```

```
array([[312,  32],
       [ 51, 293]], dtype=int64)
```

Fig. 18. Logistic Regression Confusion matrix

Recall is 85.1%.

```
print("accuracy", metrics.accuracy_score(y_pred_final.Attrition, y_pred_final.predicted))
print("precision", metrics.precision_score(y_pred_final.Attrition, y_pred_final.predicted))
print("recall", metrics.recall_score(y_pred_final.Attrition, y_pred_final.predicted))
```

```
accuracy 0.8793604651162791
precision 0.9015384615384615
recall 0.8517441860465116
```

Fig. 19. Logistic Regression – Accuracy, Precision, Recall

### Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(class_weight="balanced")
rfc.fit(X_train_balanced,y_train_balanced)
predictions = rfc.predict(X_test_balanced)
from sklearn.metrics import classification_report,confusion_matrix, accuracy_score
print(classification_report(y_test_balanced,predictions))
```

```
              precision    recall  f1-score   support

           0       0.99      0.98      0.99       344
           1       0.98      0.99      0.99       344

avg / total       0.99      0.99      0.99       688
```

```
print(confusion_matrix(y_test_balanced,predictions))
```

```
[[338   6]
 [  2 342]]
```

```
print("accuracy", metrics.accuracy_score(y_test_balanced, predictions))
print("precision", metrics.precision_score(y_test_balanced, predictions))
print("recall", metrics.recall_score(y_test_balanced, predictions))
```

```
accuracy 0.9883720930232558
precision 0.9827586206896551
recall 0.9941860465116279
```

Fig. 20. Random Forest Classifier Accuracy, Precision, Recall

It is a supervised learning algorithm. It can be used for both classification as well as regression. It creates decision trees on randomly selected data samples. It is a very good indicator of feature importance as well and is considered as a highly accurate and robust method because of the number of decision trees involved in the process.

Recall is 99%. Random Forest does not suffer from the overfitting problem generally because it takes the average of all the predictions, which cancels out the biases. But as the recall is too high therefore, to avoid overfitting the hyperparameters like n_estimators, max_features, max_depth, etc. should be tuned.

```
seed = 0
rf_params = {
    'n_jobs': -1,
    'n_estimators': 1000,
    'max_features': 0.3,
    'max_depth': 3,
    'min_samples_leaf': 2,
    'max_features' : 'sqrt',
    'random_state' : seed,
    'verbose': 0
}
rf = RandomForestClassifier(**rf_params)
rf.fit(X_train_balanced,y_train_balanced)

rf_predictions = rf.predict(X_test_balanced)

print("Accuracy score: {}".format(accuracy_score(y_test_balanced, rf_predictions)))
print("="*80)
print(classification_report(y_test_balanced, rf_predictions))
```

```
Accuracy score: 0.9331395348837209
================================================================================
              precision    recall  f1-score   support

           0       0.92      0.95      0.93       344
           1       0.95      0.91      0.93       344

avg / total       0.93      0.93      0.93       688
```

Fig. 21. Hyperparameter tuning in Random Forest Classifier

With the hyperparameter tuning, the recall is 93% and the precision and recall of both 0 and 1 are good hence, it is good model and far better than a random guess.

### *F. Implementing Machine Learning Models – Feature Ranking*

Model for finding the important features will help the business to understand the indicators of attrition. Algorithm is applied on the dataset that was prepared before PCA because PCA produces eigenvalues which cannot be used to understand the important features.

### Random Forest

Random Forest also has an attribute ***featureimportances.*** For this, we are using the dataframe prepared before applying PCA.

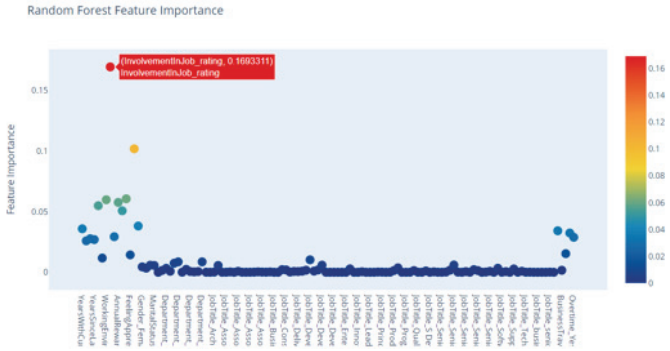


Fig. 22. Random Forest – Feature Ranking

From the above figure we can conclude that **Involvement in job is the top-ranking variable.** Daily Travel Time, Working Environment, Job Location, Annual Rewards, Total Experience, Years with Manager, Years since last promotion, Years with Company, BusinessTravelFrequency NoTravelOpportunity,Business TravelFrequency_TravelRarely are also on top rank compared to other features.

### Decision Tree

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute, each branch represents the outcome of the test and each leaf node represents a class label. The paths from root to leaf represents classification rules.

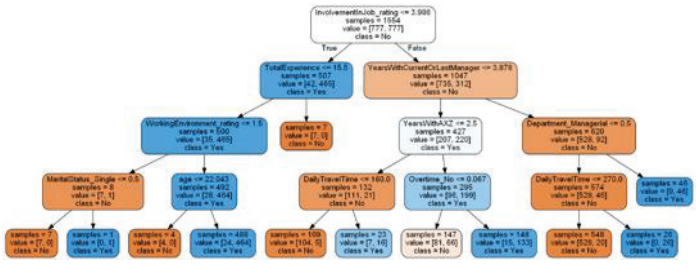

Fig. 23. Decision Tree– Feature Ranking

As you can see from the above figure, **involvement in job** is the main factor considered by the decision tree.

### Gradient Boosting

Gradient Boosting produces a prediction model in the form of ensemble of weak prediction models, typically decision trees. Gradient Boosting starts combining the processes at the beginning whereas Random Forest uses average or majority at the end of the process.

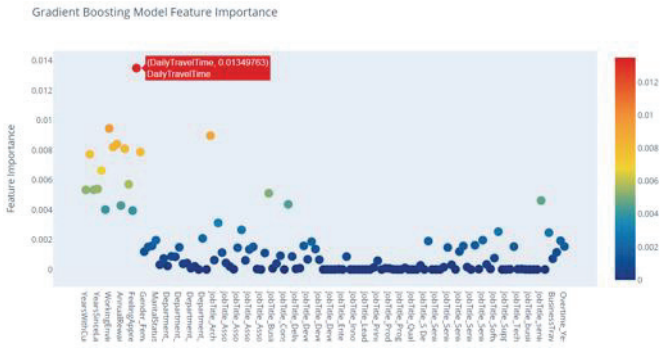


Fig. 24. Gradient Boosting – Feature Ranking

From the above figure we can conclude that **Daily Travel Time** is the top-ranking variable.

## III. CONCLUSION AND FUTURE SCOPE

This paper explains the basic pipeline of predicting workforce attrition in IT companies. It included basic Exploratory Data Analysis, Feature Engineering, Implementing Logistic Regression, Random Forest, Decision Tree and Gradient Boosting. Features like travel time, involvement in job, working environment, job location, annual rewards, travel opportunity, learning opportunity, years with manager, years since last promotion were common in top ranks in all the models. The API from the model can be exposed and used in a User Interface where the HR would be able to input the features of existing employees and the model will predict whether they are about to leave or not. If yes, then the HR can focus on the important features derived from the model and take on time action to retain the workforce if required.

In the survey for alumni, another question has been asked "What was the reason of leaving AXZ?", where options are provided like Higher Education, Salary, Working Environment, Skill Development, Better Role Outside, Work Life Balance, Manager, Stress, Relocation, Not felt appreciated, felt underutilized, seeing good employees leave, others. Alumnus filling the survey had to choose one of the above-mentioned options. All these reasons can be used in future and can act as categories. Machine Learning algorithm can be applied on the data set with Attrition equal to "Yes" where target variable will be the reason of leaving. Let's call this model as "B". So, once the model to predict whether a resource is about to leave or not has predicted that an employee "A" is about to leave in near future, then the features of A can be passed to model B to predict the category in which the reason to leave falls.

This analysis is done for one company, in future the data can be collected from more than one IT firm and a combined analysis can be done. Also, dynamic modeling can be done as and when the data increases.

## REFERENCES


[1] https://www.forbes.com/sites/karlynborysenko/2020/02/03/64-of-employees-may-leave-their-jobs-in-2020/#183412245cd0

[2] https://www.humanresourcestoday.com/retention-and-turnover/?open-article-id=12269558&article-title=employee-retention-and-turnover-solutions&blog-domain=bonus.ly&blog-title=bonusly

[3] https://economictimes.indiatimes.com/tech/ites/infosys-board-wants-to-monitor-steps-to-bring-down-attrition/articleshow/70371816.cms

[4] https://economictimes.indiatimes.com/tech/ites/infosys-reports-1-4-involuntary-attrition-in-q2/articleshow/71544187.cms

[5] https://investor.accenture.com/~/media/Files/A/Accenture-IR-V3/quarterly-earnings/2020/q1fy20/q1-fy20-supporting-materials.pdf

[6] https://www.firstpost.com/business/wipro-may-promote-5000-employees-this-financial-year-move-aims-at-checking-attrition-preparing-company-future-ready-7531621.html#:~:text=officer%20at%20Wipro.-,Wipro's%20attrition%20rate%20reportedly%20stood%20at%2017%20percent%2C%2060%20basis,to%20a%20report%20in%20Mint.

[7] https://economictimes.indiatimes.com/news/company/corporate-trends/tcs-says-large-contracts-will-help-it-absorb-cost-of-keeping-bench-talent/articleshow/70336196.cms

[8] https://qtrial2019q3az1.az1.qualtrics.com/jfe/form/SV_bHlZiM8DuuTZYKp

[9] https://qtrial2019q3az1.az1.qualtrics.com/jfe/form/SV_8DqAmOBF3KA27xH