Objective & Overview of data

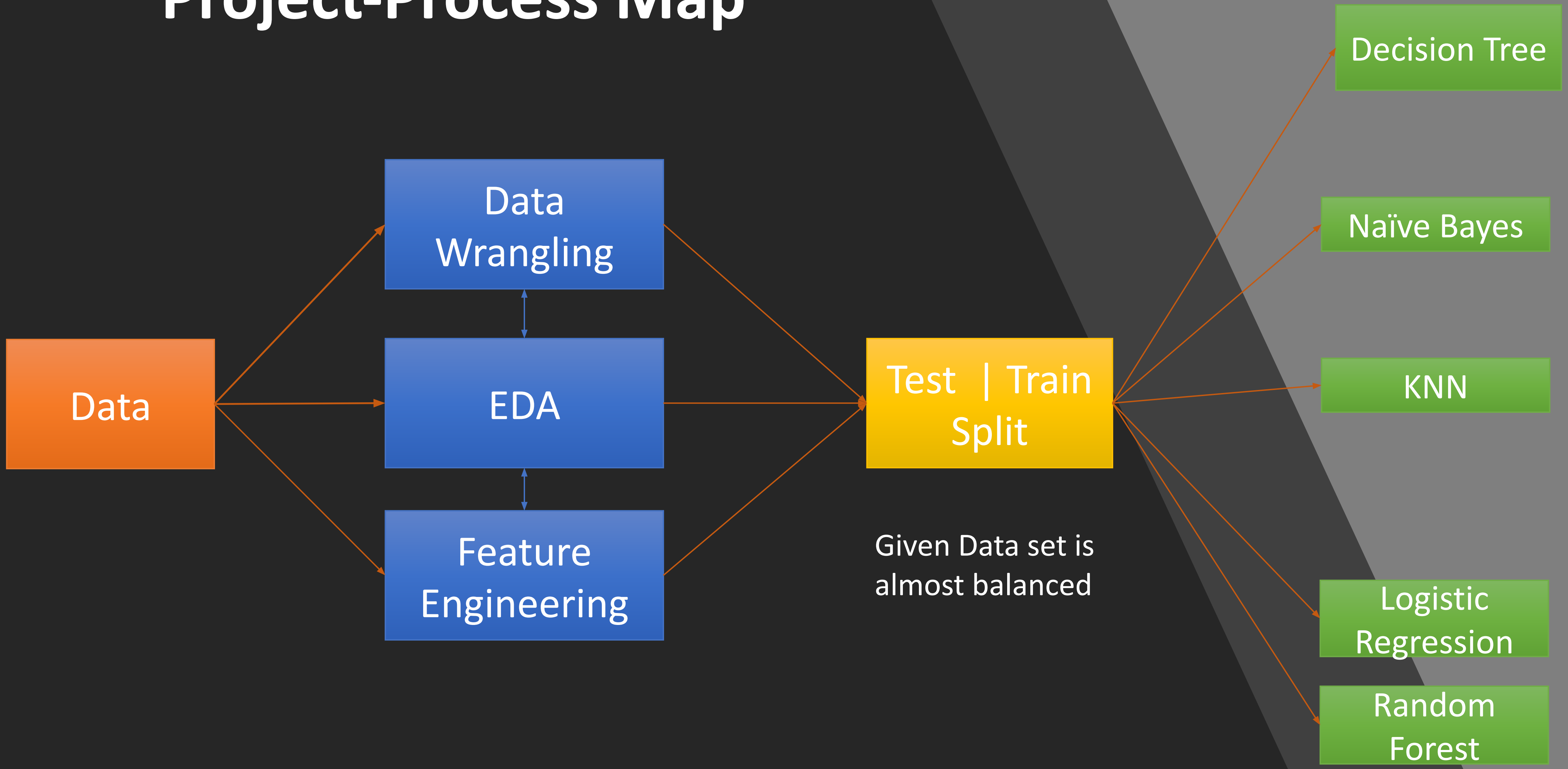
- **Number of observations:** 9612 entries
- **Number of attributes: 27**
- NULL values are present in REFERRAL_SOURCE for more than 10%
- REFERRAL_SOURCE also had a value of "Unknown".
- The given data set is almost balanced
  - Active : 5394
  - Terminated : 4218
- We have some data discrepancies that are discussed in the next slides

Overview:
Employee Attrition (also known as "employee churn") is a costly problem for companies. The true cost of replacing an employee can often be quite large. Understanding why and when employees are most likely to leave can lead to actions to improve employee retention as well as possibly planning new hiring in advance

Objective:
- What is the likelihood of an active employee leaving the company?
- What are the key indicators of an employee leaving the company?
- We will use this dataset to predict when employees are going to quit by understanding the main drivers of employee churn.

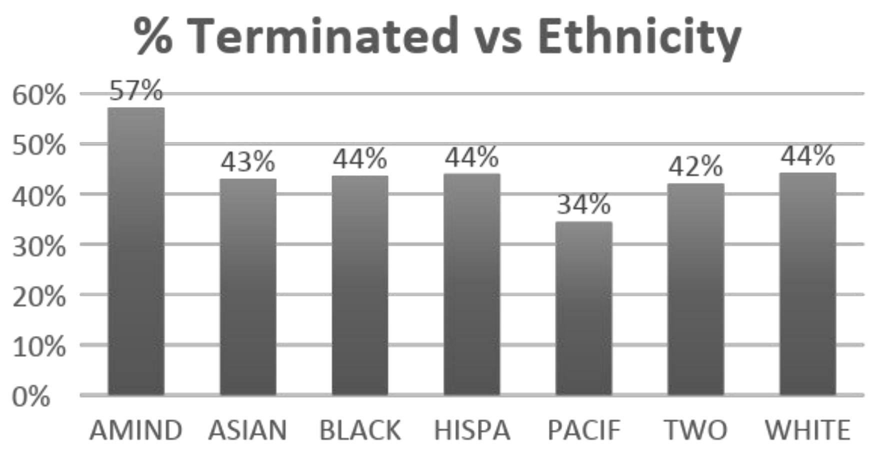
	ANNUAL_RATE	HRLY_RATE	JOBCODE	JOB_SATISFACTION	AGE	PERFORMANCE_RATING
count	9.612000e+03	9612.000000	9612.000000	9612.000000	9612.000000	9612.000000
mean	8.938563e+04	49.953808	51485.811174	2.757491	40.151581	3.002081
std	5.843344e+04	28.148113	22853.906872	1.411257	13.664378	1.406909
min	1.678600e+04	14.000000	10006.000000	1.000000	18.000000	1.000000
25%	5.085550e+04	32.000000	33534.000000	2.000000	28.000000	2.000000
50%	7.421050e+04	43.000000	52981.000000	3.000000	39.000000	3.000000
75%	1.088115e+05	59.000000	69401.000000	4.000000	52.000000	4.000000
max	1.250924e+06	608.000000	99793.000000	5.000000	64.000000	5.000000

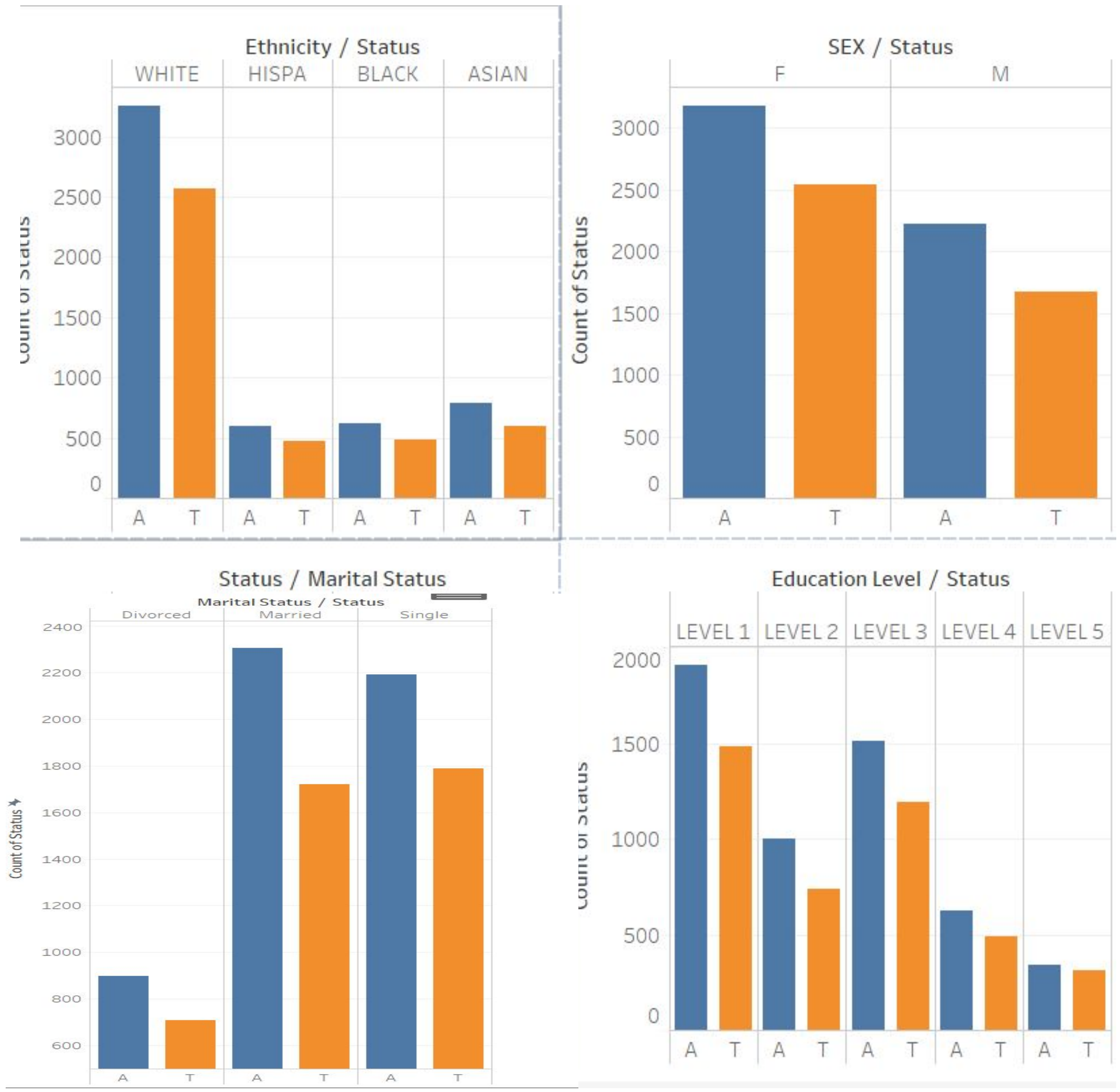
Project-Process Map



EDA-Count of Status

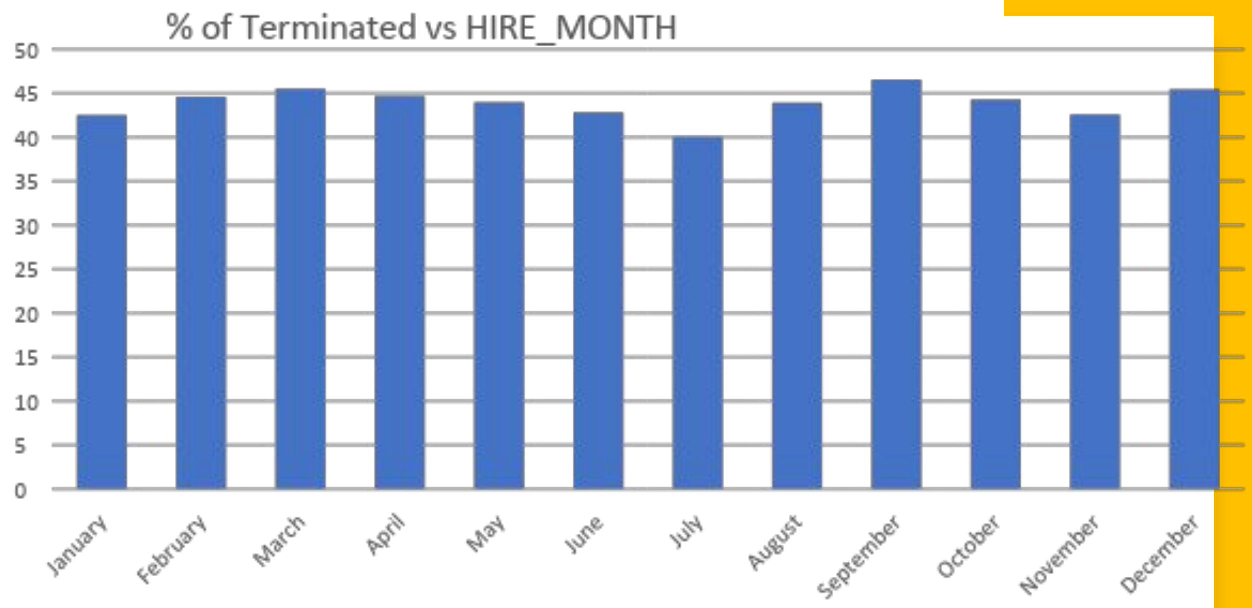
- Slightly more attrition in female employees and Single employees
- Education level 5 have slightly more percentage of terminated employees
- Though AMINDS (ethnicity) are less in number, % of termination is high among them

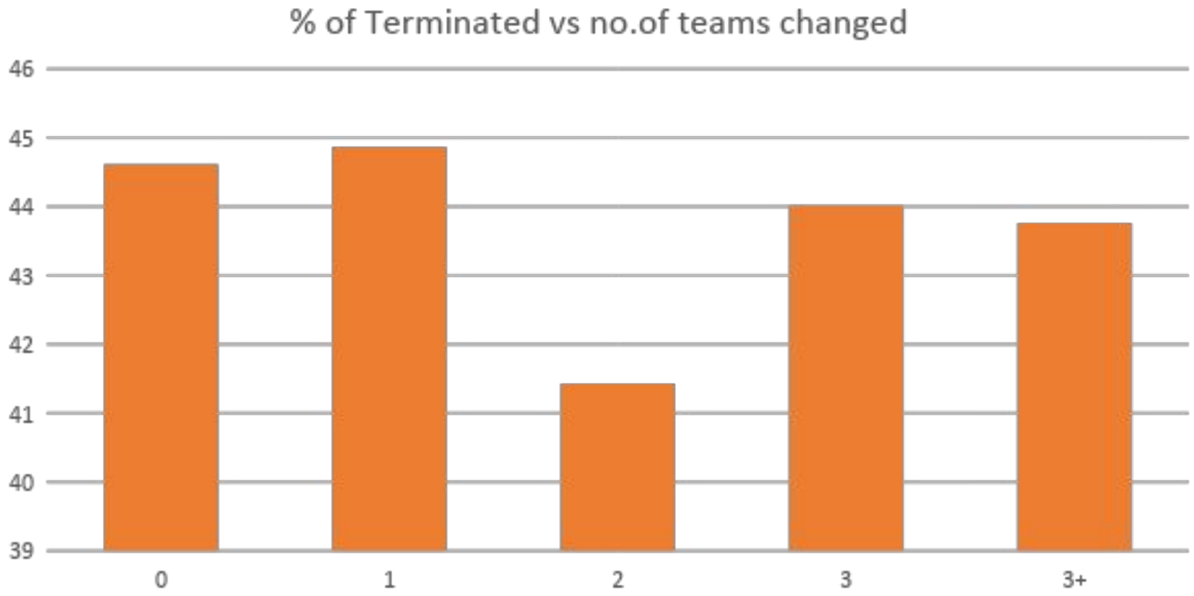




EDA

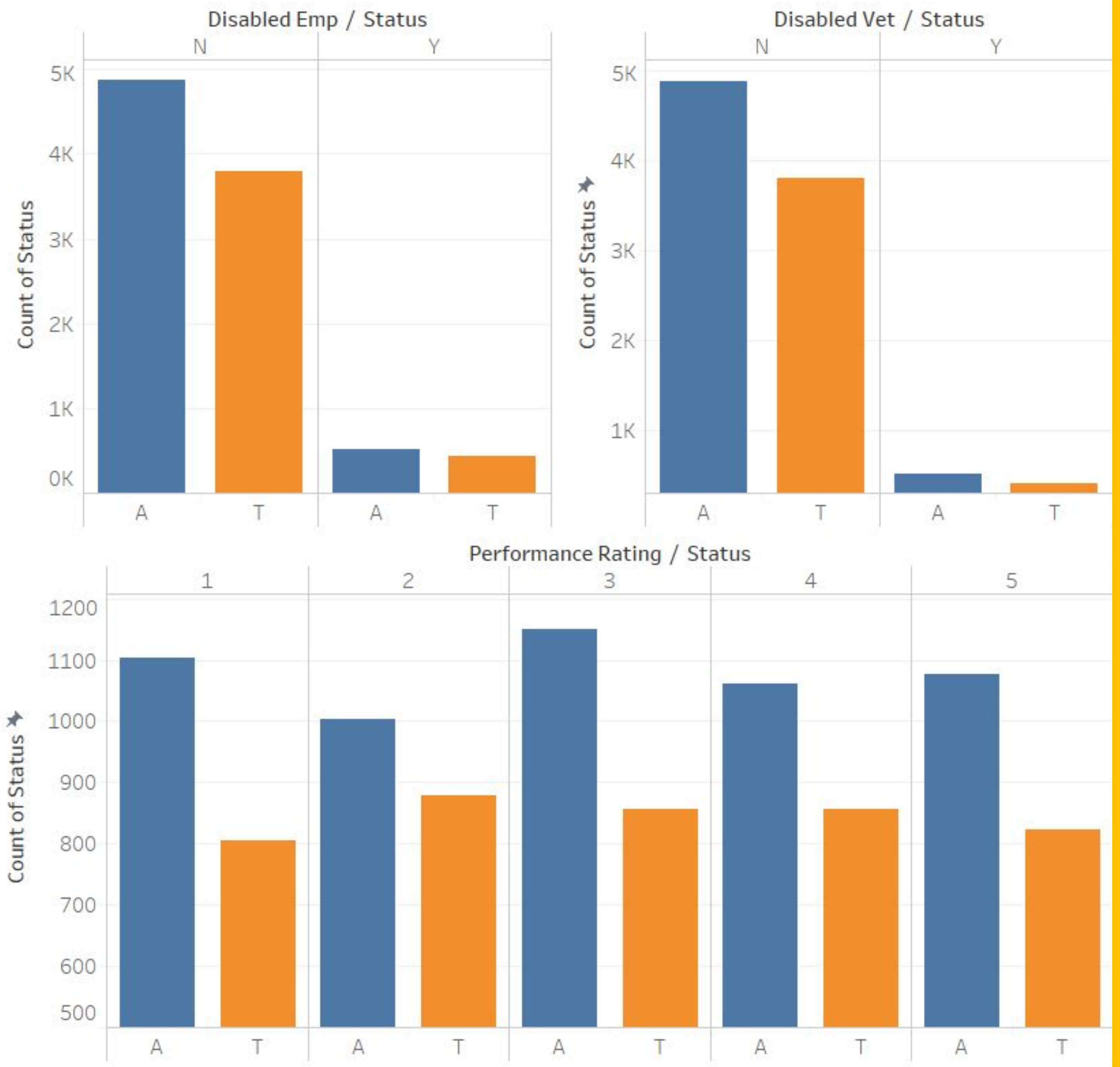
- HIRE_MONTH has very little effect on percentage of terminated
- For an employees who changed just 2 teams has higher attrition rate when compared to others
- Employee who have job satisfaction : 1 or (very less) are likely to leave the company, which is quite intuitive





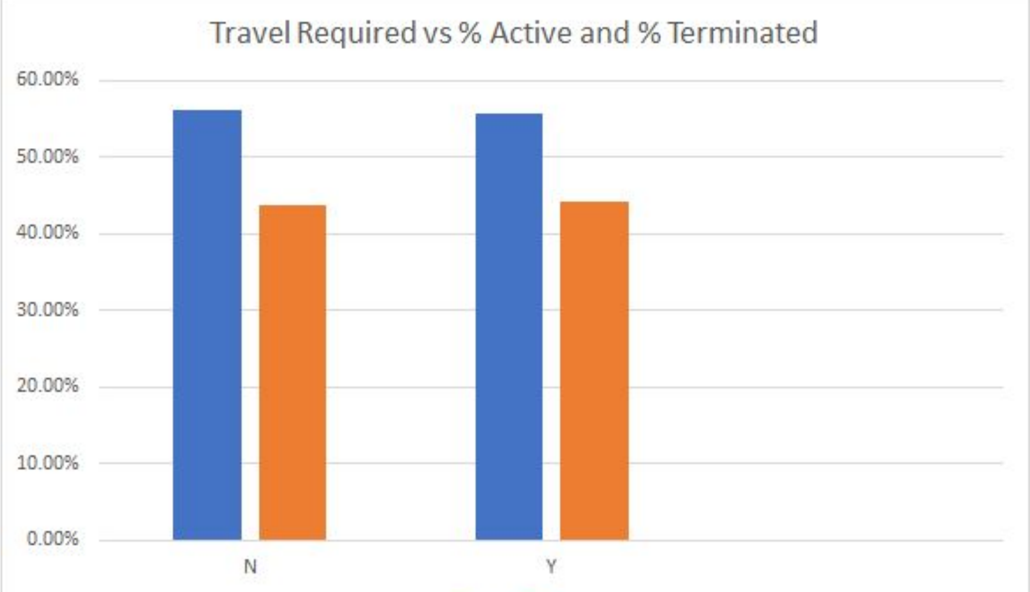
EDA

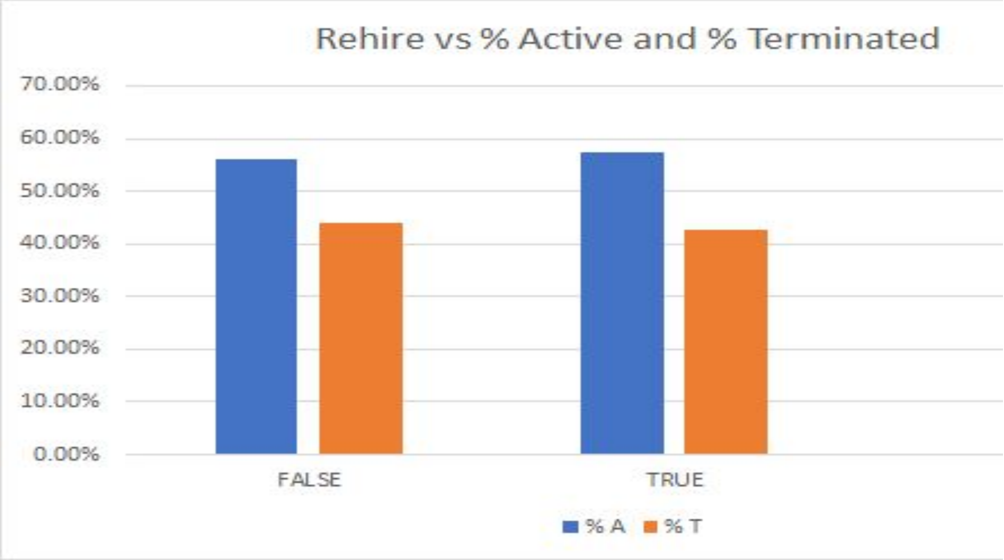
- Slightly more attrition in Disabled Veterans and Disabled Employee
- Performance rating 2 has more attrition when compared to others

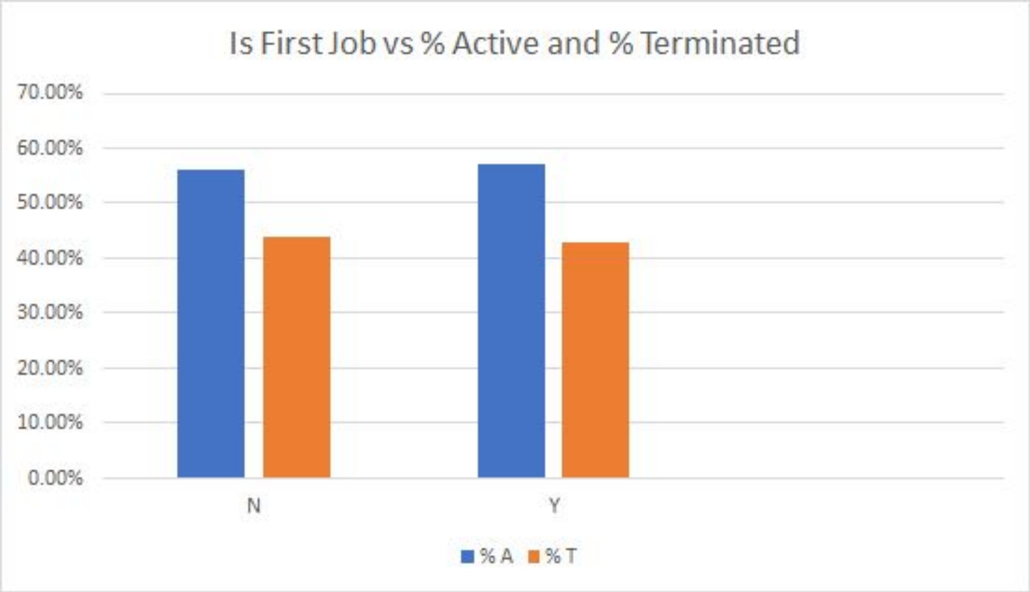


EDA

- Employees for whom this is their first job seems to churn at a slightly lower rate of 42.8% than their counterparts at 44%
- Employees requiring travel seem to be churning at slightly higher rates than their counterparts







Data Discrepancy

- As we can see that Employee (3626639527) who is terminated in 2017 and not a rehire has a PREVYR_1 rating as 0 and PREVYR_2 as 1, which indicates that the employee was present in the company 2 years ago and data was collected in 2019

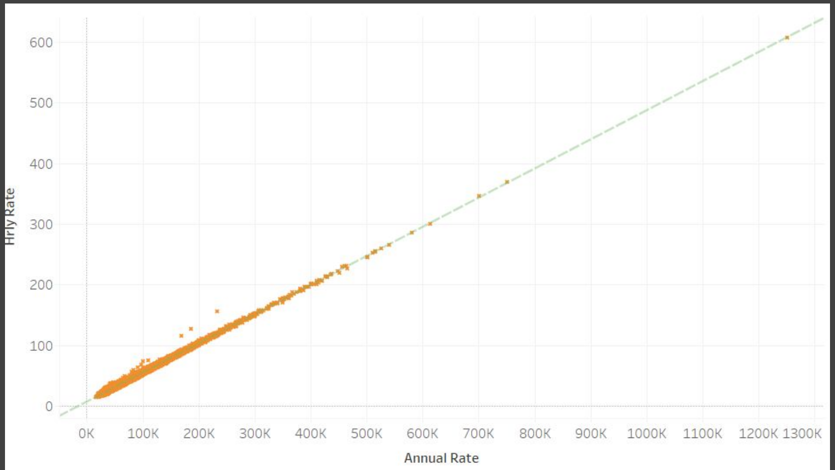
- Whereas the Employee (5127603797) who is terminated in 2014 and not a rehire has a PREVYR_1 rating as 0 and PREVYR_2 as 3 ,which indicated that the employee was present in the company 2 years ago and data was collected in 2016.

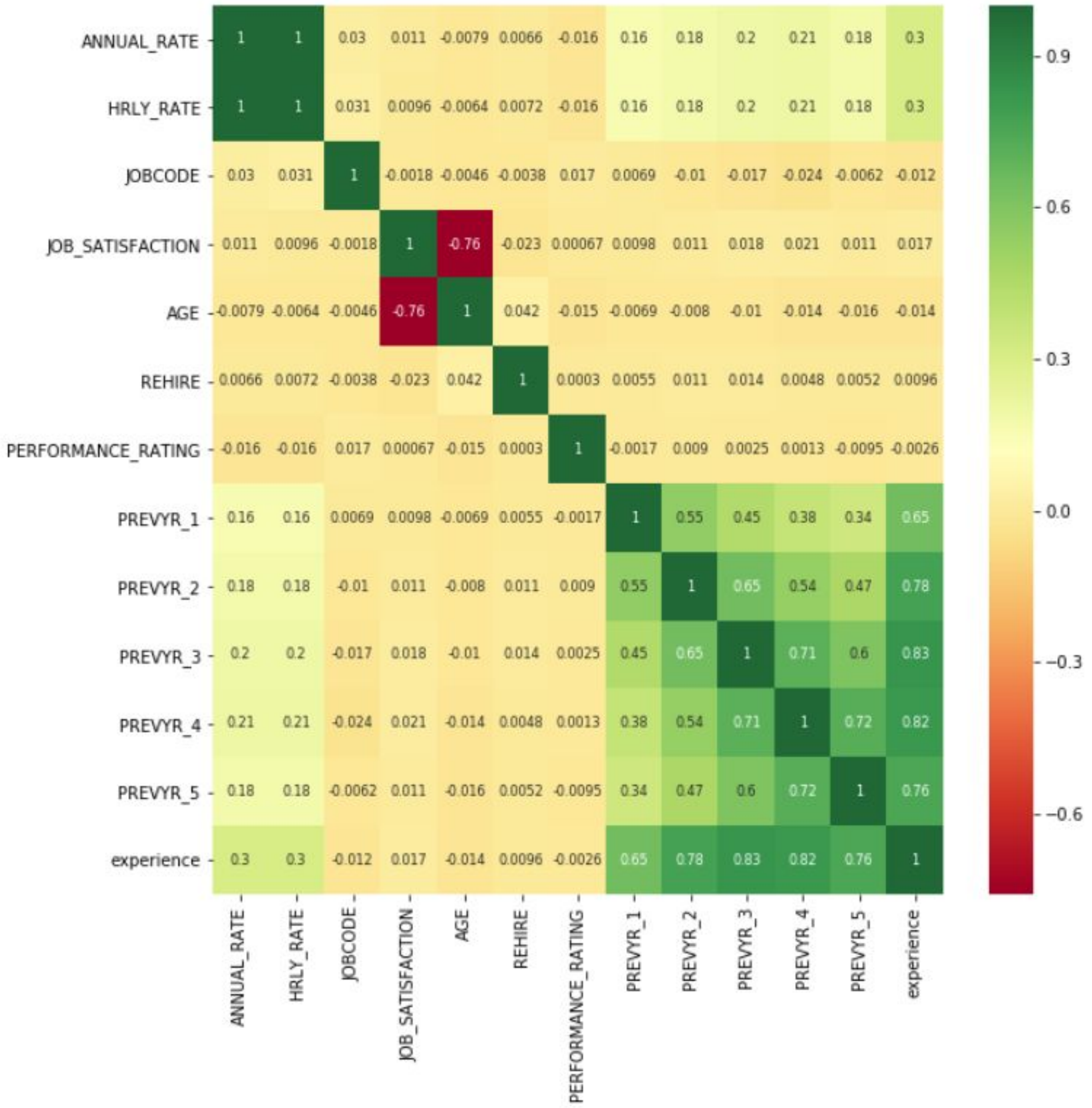
- **Such discrepancies in the dataset creates ambiguity about the data collection year i.e. 2016 or 2019**

EMP_ID	REHIRE	TERMINATION_YEAR	PERFORMANCE_RATING	PREVYR_1	PREVYR_2	PREVYR_3	PREVYR_4	PREVYR_5	EXPERIENCE_AT_COMPANY
3626639527	FALSE	2017	1	0	1	0	0	0	2
5127603797	FALSE	2014	1	0	3	3	3	3	5

Correlation Matrix

- Each cell in the table shows the **correlation** between two variables
- Annual rate and hourly rate are highly correlated
- Experience (calculated variable) is correlated with all previous year ratings
- Surprisingly, Job satisfaction and Age are negatively correlated – This correlation is just spurious in nature





Feature Engineering ctd..

- Created a calculated variable called "EXPERIENCE_AT_COMPANY" – How many years of experience a person has in the past 5 years
- Logic for Experience:
  - If the previous year rating is not zero, then the employee worked for that year in the company.
  - Hence that year counts in employee's experience at the company

Calculated Variable:

EXPERIENCE_AT_COMPANY

```
df['PREVYR_1_PRESENT'] = [1 if value > 0 else 0 for value in df['PREVYR_1']]
df['PREVYR_2_PRESENT'] = [1 if value > 0 else 0 for value in df['PREVYR_2']]
df['PREVYR_3_PRESENT'] = [1 if value > 0 else 0 for value in df['PREVYR_3']]
df['PREVYR_4_PRESENT'] = [1 if value > 0 else 0 for value in df['PREVYR_4']]
df['PREVYR_5_PRESENT'] = [1 if value > 0 else 0 for value in df['PREVYR_5']]
df['EXPERIENCE_AT_COMPANY'] = df['PREVYR_1_PRESENT'] + df['PREVYR_2_PRESENT']
```

PREVYR_1	PREVYR_2	PREVYR_3	PREVYR_4	PREVYR_5
3	3	3	2	0

Experience=4

Feature Engineering ctd..

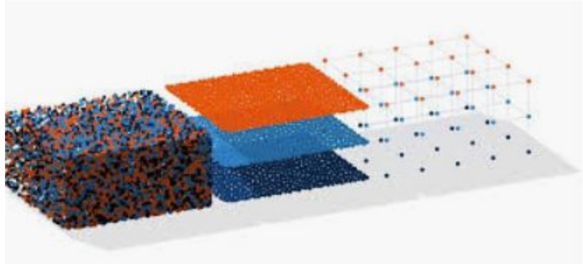
Calculated Variable:

Diversity of a Job group (Ethnicity)

- Created calculated variables "Diversity rate for a Job Group based on ethnicity" – How diverse is a job group is
- Combined the Ethnicity Data frame to the Attrition data set on Job_Group

- Logic for Diversity rate :

```
Ethin_df=df.groupby(['JOB_GROUP']).apply(lambda x: pd.Series(dict(
    employee_cnt=x.EMP_ID.nunique(),
    terminated_cnt=x[x['STATUS']==1].EMP_ID.nunique(),
    terminated_rate=x[x['STATUS']==1].EMP_ID.nunique()/x.EMP_ID.nunique(),
    white_rate=x[x['ETHNICITY']=='WHITE'].EMP_ID.nunique()/x.EMP_ID.nunique(),
    asian_rate=x[x['ETHNICITY']=='ASIAN'].EMP_ID.nunique()/x.EMP_ID.nunique(),
    black_rate=x[x['ETHNICITY']=='BLACK'].EMP_ID.nunique()/x.EMP_ID.nunique(),
    hispa_rate=x[x['ETHNICITY']=='HISPA'].EMP_ID.nunique()/x.EMP_ID.nunique(),
))).reset_index()
Ethin_df.sort_values('employee_cnt',ascending=False)
```



JOB_GROUP	employee_cnt	terminated_cnt	terminated_rate	white_rate	asian_rate	black_rate	hispa_rate
Production & Operations	1714.0	819.0	0.477830	0.601517	0.150525	0.119603	0.104434
Marketing - Direct	849.0	542.0	0.638398	0.586572	0.141343	0.124853	0.124853
Physical Flows	816.0	169.0	0.207108	0.601716	0.150735	0.115196	0.115196
Finance	525.0	284.0	0.540952	0.634286	0.127619	0.097143	0.108571
Human Resources	396.0	153.0	0.386364	0.580808	0.146465	0.121212	0.136364
Customer Care	355.0	147.0	0.414085	0.639437	0.140845	0.109859	0.092958
General Administration	343.0	166.0	0.483965	0.600583	0.177843	0.116618	0.072886
Marketing - Global	296.0	111.0	0.375000	0.594595	0.135135	0.125000	0.131757
R&I General Management	250.0	211.0	0.844000	0.612000	0.120000	0.124000	0.108000

Feature Engineering ctd..

Calculated Variable: **COMBINED_JOB_GROUP**

- Binned certain JOB_GROUP values into a broader group
- Exact binning mentioned in the last page of the report
- ** These values were used in the models but the accuracies did not improve, so we discarded them from the model and used them for EDA

	JOB_GROUP	COMBINED_JOB_GROUP
0	Plant & Facilities Maintenance	Manufacturing & Production
1	Customer Care	Business
2	Customer Care	Business
3	Finance	Finance
4	Marketing - Direct	Marketing
5	Physical Flows	Manufacturing & Production
6	Marketing - Direct	Marketing
7	Finance	Finance
8	Tax	Finance
9	General Administration	General
10	Production & Operations	Manufacturing & Production
11	R&I Development/Pre-Develpmnt	Research & Development
12	Sourcing	Human Resources
13	IT Business Applications	IT
14	Production & Operations	Manufacturing & Production
15	Human Resources	Human Resources
16	Promotional Purchasing	Marketing
17	Creative Service/Copy	Research & Development
18	Sourcing	Human Resources
19	R&I Development/Pre-Develpmnt	Research & Development

Feature Engineering ctd..

Calculated Variable: **DISCRETIZED_AGE**

- Discretized age in ranges of 5 years starting from 18
- More than 60 years discretized to “60 or above”

Code for Age Discretization:

```
discretized_age = []
for age in df['AGE']:
    if age >= 18 and age <= 23:
        discretized_age.append('18-23')
    elif age > 23 and age <= 29:
        discretized_age.append('24-29')
    elif age > 29 and age <= 35:
        discretized_age.append('30-35')
    elif age > 35 and age <= 41:
        discretized_age.append('36-41')
    elif age > 41 and age <= 47:
        discretized_age.append('41-47')
    elif age > 47 and age <= 53:
        discretized_age.append('48-53')
    elif age > 53 and age <= 59:
        discretized_age.append('54-59')
    elif age > 59:
        discretized_age.append('60 or above')
df['DISCRETIZED_AGE'] = discretized_age
```

Sample output:

	AGE	DISCRETIZED_AGE
0	35	30-35
1	18	18-23
2	18	18-23
3	50	48-53
4	34	30-35
5	31	30-35
6	39	36-41
7	21	18-23

Feature Engineering ctd..

Calculated Variable: **DISCRETIZED_ANNUAL_RATE**

- Discretized ANNUAL_RATE to LOW, MEDIUM, HIGH, and VERY HIGH
- Discretization based on quantiles because it is evenly distributed by number of employees as per slide number 2

Code for Annual rate Discretization:

```
quantiled_annual_rate = df['ANNUAL_RATE'].quantile([0.25,0.5,0.75])
discretized_annual_rate_list = []
for annual_rate in df['ANNUAL_RATE']:
    if quantiled_annual_rate[0.25] > annual_rate:
        discretized_annual_rate_list.append('LOW')
    elif quantiled_annual_rate[0.25] <= annual_rate and quantiled_annual_rate[0.50] > annual_rate:
        discretized_annual_rate_list.append('MEDIUM')
    elif quantiled_annual_rate[0.50] <= annual_rate and quantiled_annual_rate[0.75] > annual_rate:
        discretized_annual_rate_list.append('HIGH')
    elif quantiled_annual_rate[0.75] <= annual_rate:
        discretized_annual_rate_list.append('VERY HIGH')
df['DISCRETIZED_ANNUAL_RATE'] = discretized_annual_rate_list
```

Sample output:

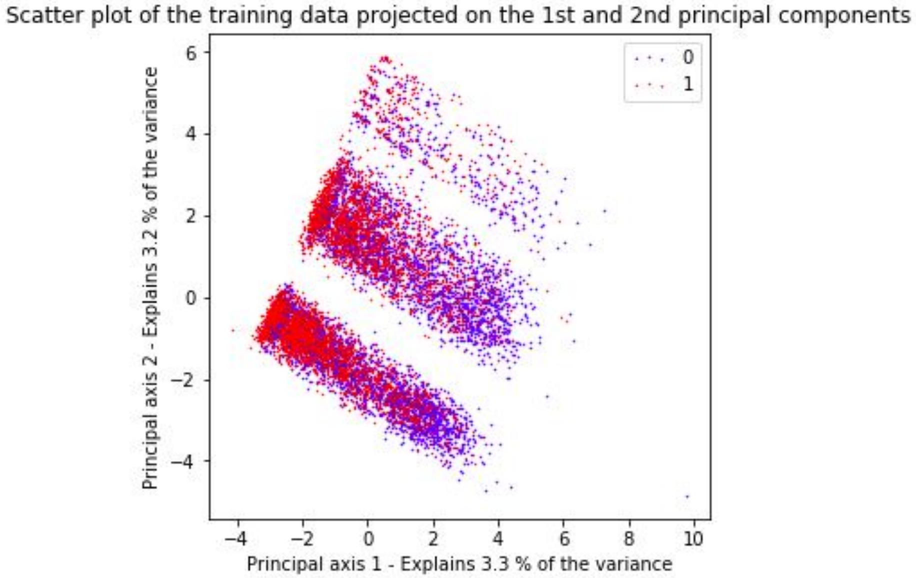
	ANNUAL_RATE	DISCRETIZED_ANNUAL_RATE
0	33615	LOW
1	70675	MEDIUM
2	34320	LOW
3	103199	HIGH
4	141801	VERY HIGH
5	31615	LOW
6	91425	HIGH
7	189200	VERY HIGH
8	144069	VERY HIGH
9	205811	VERY HIGH

Dimensionality Reduction– Principal Component Analysis (PCA)

- PCA is defined as an orthogonal linear transformation technique that transforms the data into a new coordinate system.
- It is used to emphasize variation and bring out strong patterns in a dataset

```
explained_variance = pca.explained_variance_ratio_
explained_variance
```

```
array([0.033, 0.032, 0.020, 0.018, 0.017, 0.017, 0.017, 0.015, 0.015,
       0.015, 0.012, 0.011, 0.011, 0.011, 0.010, 0.010, 0.010, 0.010,
       0.010, 0.010, 0.010, 0.010, 0.010, 0.010, 0.010, 0.010, 0.010,
       0.010, 0.009, 0.009, 0.009, 0.009, 0.009, 0.009, 0.009, 0.009,
       0.009, 0.009, 0.009, 0.009, 0.009, 0.009, 0.009, 0.009, 0.009,
       0.009, 0.009, 0.009, 0.009, 0.009, 0.009, 0.009, 0.009, 0.009,
       0.009, 0.009, 0.009, 0.009, 0.009, 0.009, 0.009, 0.009, 0.009,
       0.009, 0.009, 0.009, 0.009, 0.009, 0.009, 0.009, 0.009, 0.009,
       0.008, 0.008, 0.008, 0.008, 0.008, 0.008, 0.008, 0.008, 0.008,
       0.008, 0.008, 0.008, 0.008, 0.008, 0.008, 0.008, 0.008, 0.008,
       0.008, 0.008, 0.008, 0.008, 0.008, 0.007, 0.007, 0.007, 0.006,
       0.006, 0.004, 0.003, 0.003, 0.002, 0.002, 0.001, 0.000, 0.000,
       0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000])
```

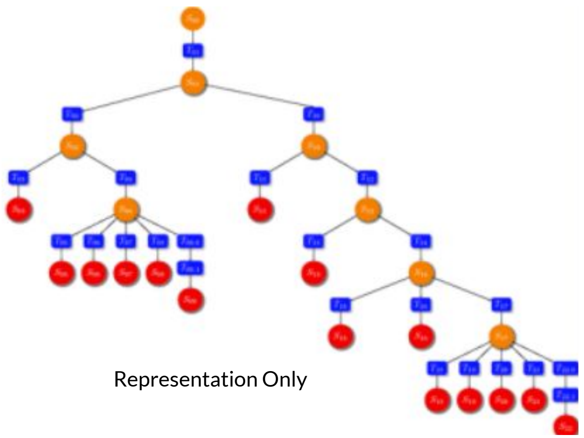


Dimensionality Reduction– Principal Component Analysis (PCA)

- PCA analysis is done on the dataset after excluding TERMINATION_YEAR and EMP_ID.
- We used label encoding where the hierarchy is important such as for discretized age, job satisfaction etc.
- One-hot encoding where the hierarchy is not important such as for marital status, job group etc.
- Total 116 columns (including the encoded columns). By looking at the variance ratio distribution bar plot and the scree plot we can infer that selecting 80 components out of 116 would be ideal to explain 80% of the variance in the data set
- It does not make much of a difference in performance after dropping just about 10 columns. So according to our analysis, we discard applying PCA to our data before using classification algorithms.

Decision Tree

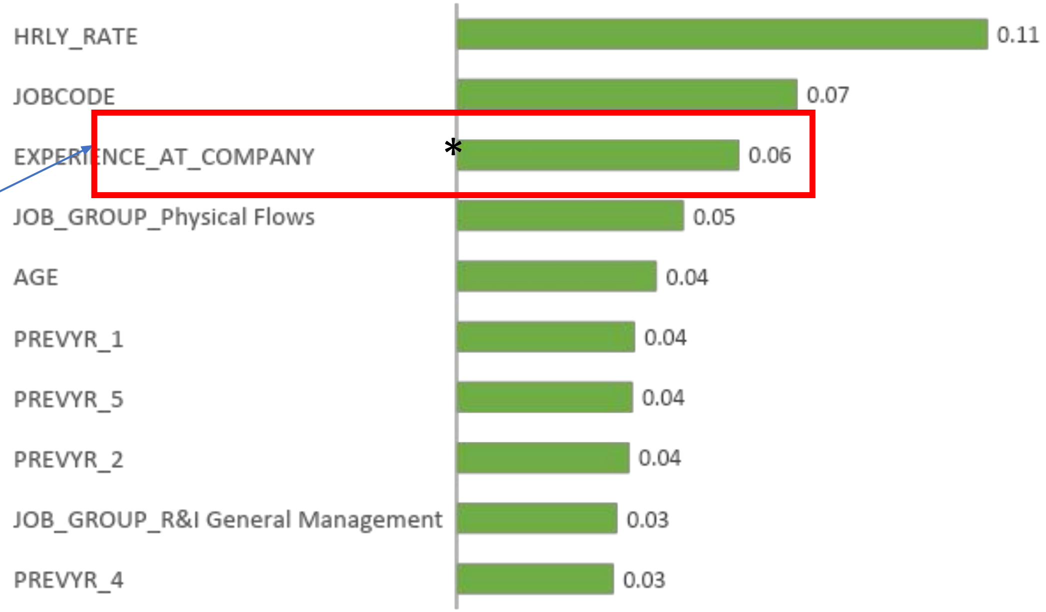
Decision Trees (DTs) are non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.



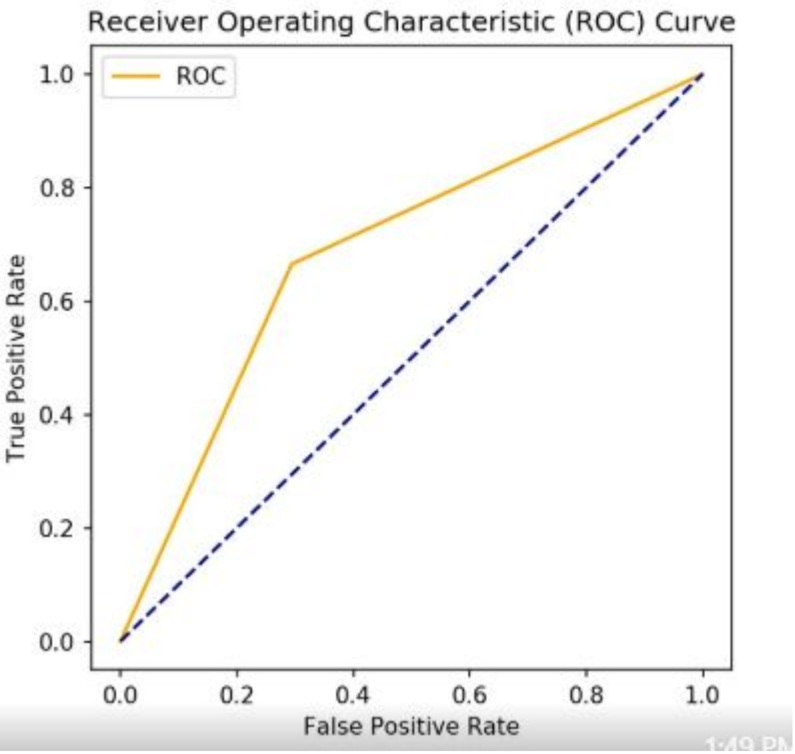
Representation Only

68.79%
Accuracy

Top 10 Features



Our Calculated Variable, **Experience** is one of the important features in the model



Confusion Matrix

		Predicted	
		A	T
Actual	A	1147	478
	T	422	837

KNN

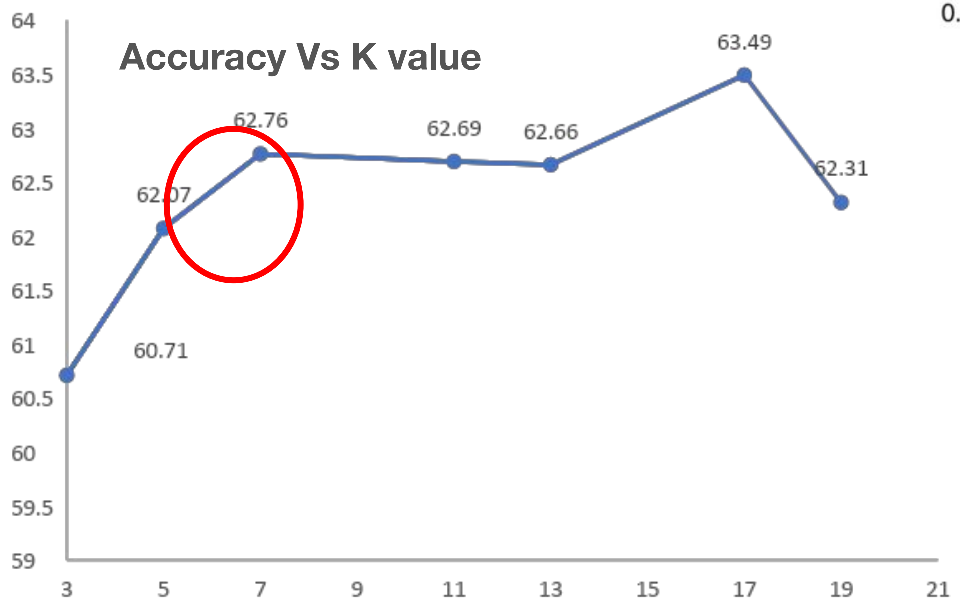
KNN is a supervised machine learning algorithm that relies on labeled input data to learn a function that produces an appropriate output when given new unlabeled data.

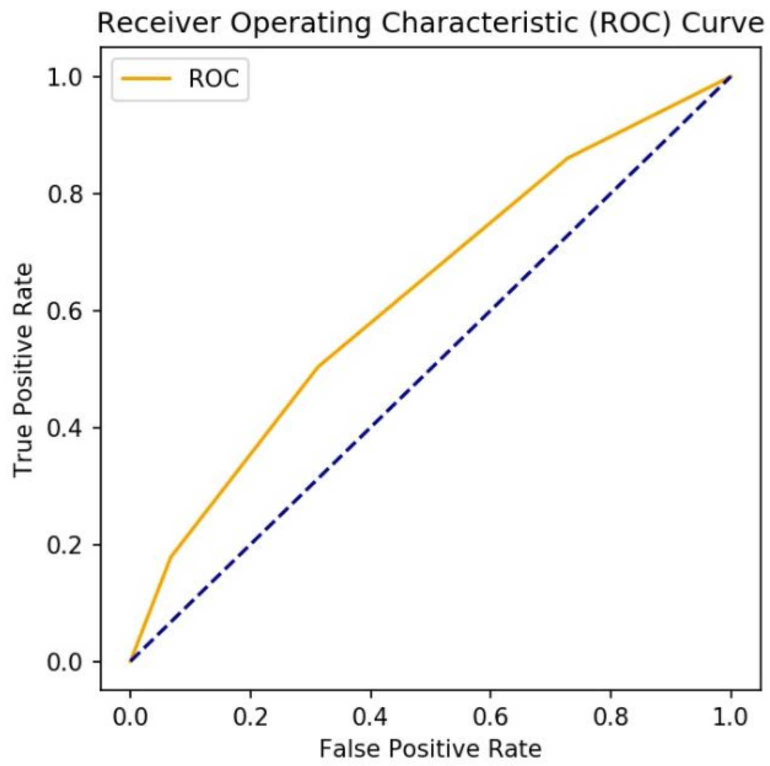
We have checked the Accuracies for K values ranging from 3 – 19. The Elbow curve starts at 7, therefore we choose K=7 as our final value

62.76% Accuracy

Advantages:

- The algorithm is simple and easy to implement.
- There's no need to tune several parameters
- The algorithm is versatile. It can be used for classification & regression



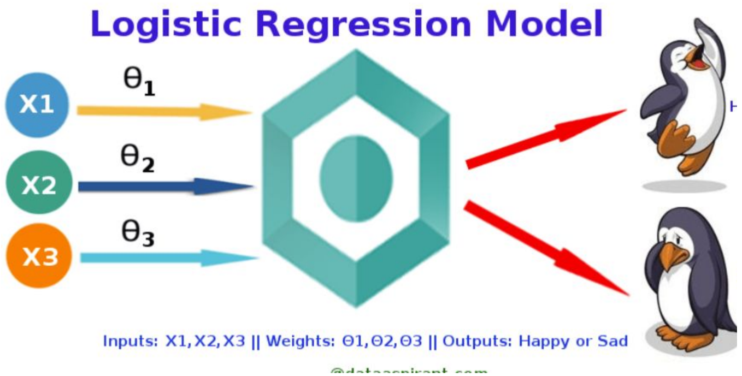


Confusion Matrix

		Predicted A	Predicted T
Actual	A	1117	508
Actual	T	625	634

Logistic Regression

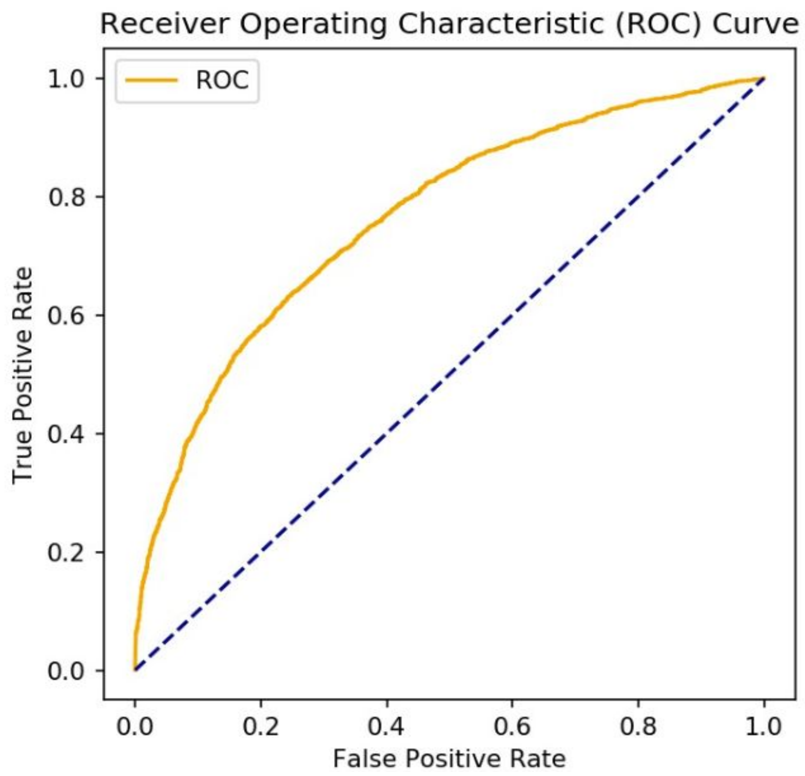
Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, in this case (Active or Terminated)

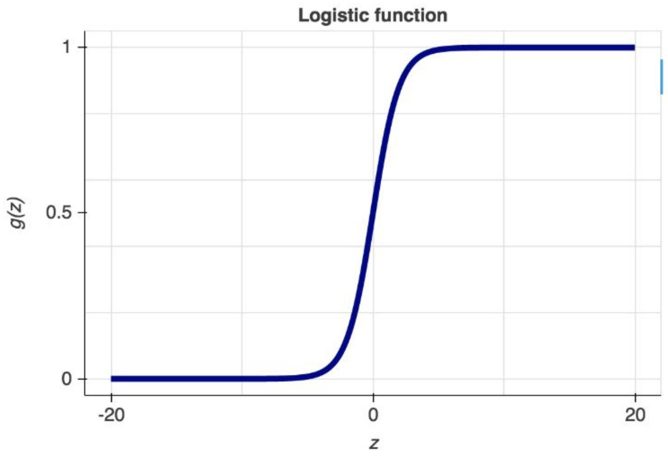


Active

Terminated

71.64%
Accuracy





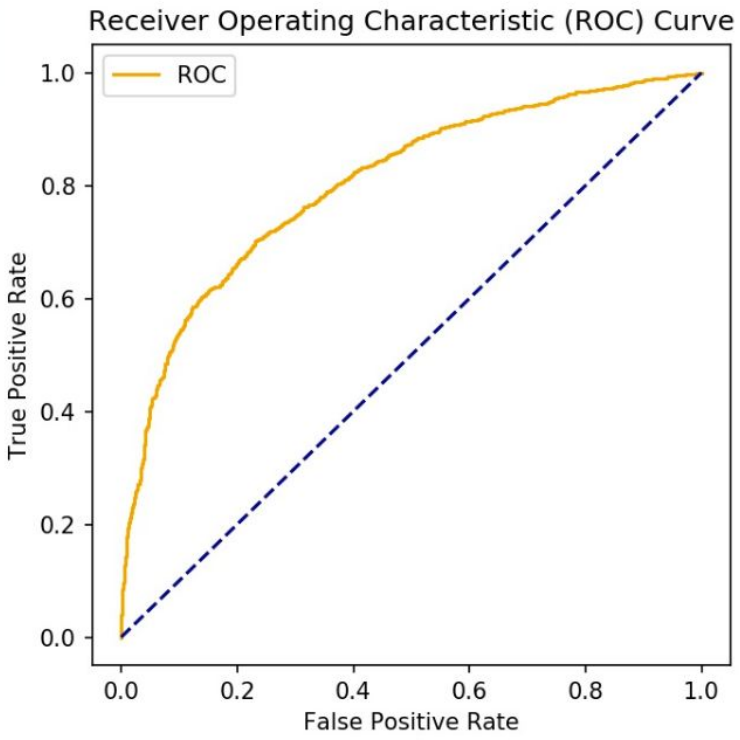
Representation Only

Confusion Matrix

		Predicted A	Predicted T
Actual	A	1259	366
	T	452	807

Random Forest

The random forest is a classification algorithm consisting of many decision trees. It uses bagging and feature randomness while building individual trees. It tries to create an uncorrelated forest of trees whose predictions by a committee is more accurate than that of any individual tree.



Hyper Parameter Tuning with Grid search and K-fold Cross Validation

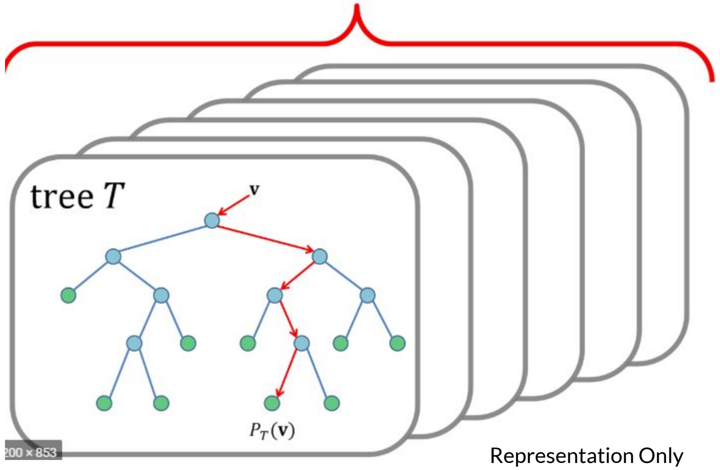
Default Hyper Parameters

73.93% Accuracy

Tuned Hyper Parameters

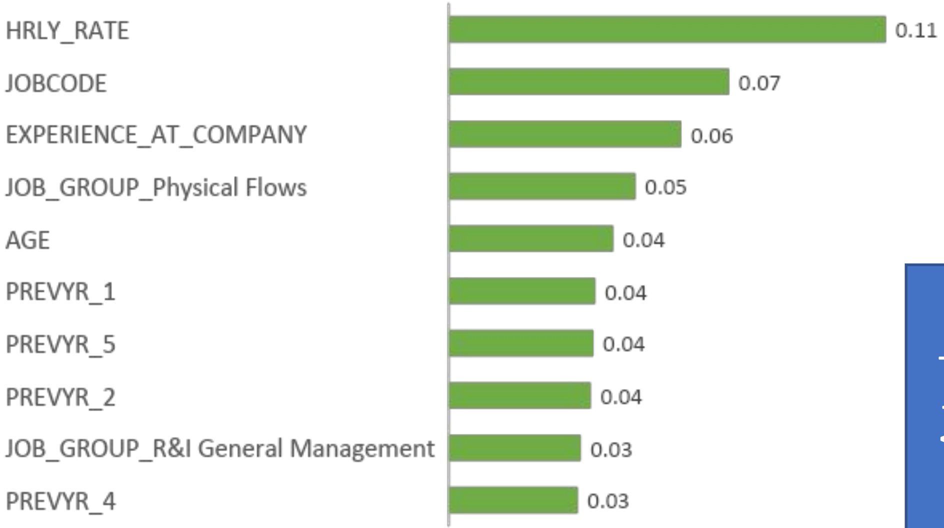
74.45% Accuracy

Random Forest



Representation Only

Top 10 Features



Confusion Matrix

		Predicted A	Predicted T
Actual	A	1350	275
	T	462	797

Naïve Bayes

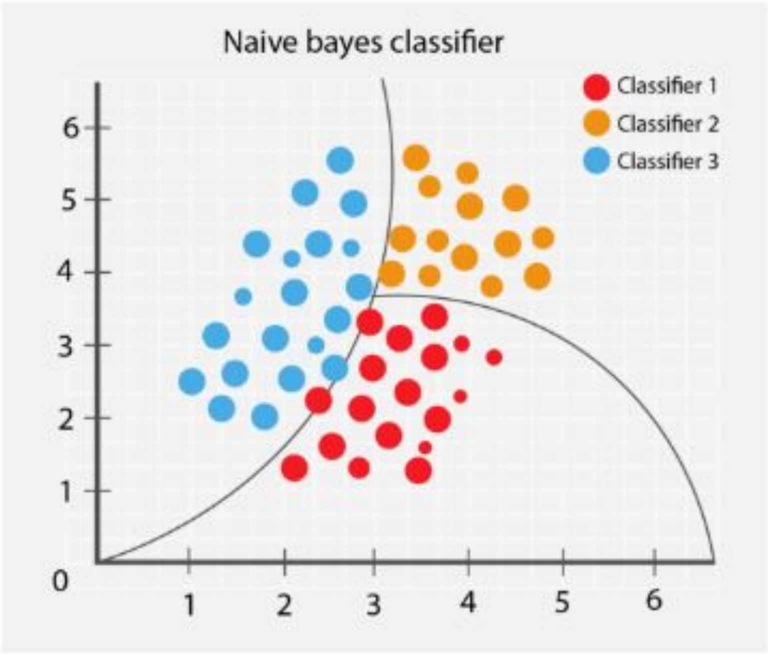
Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable

60.64%
Accuracy

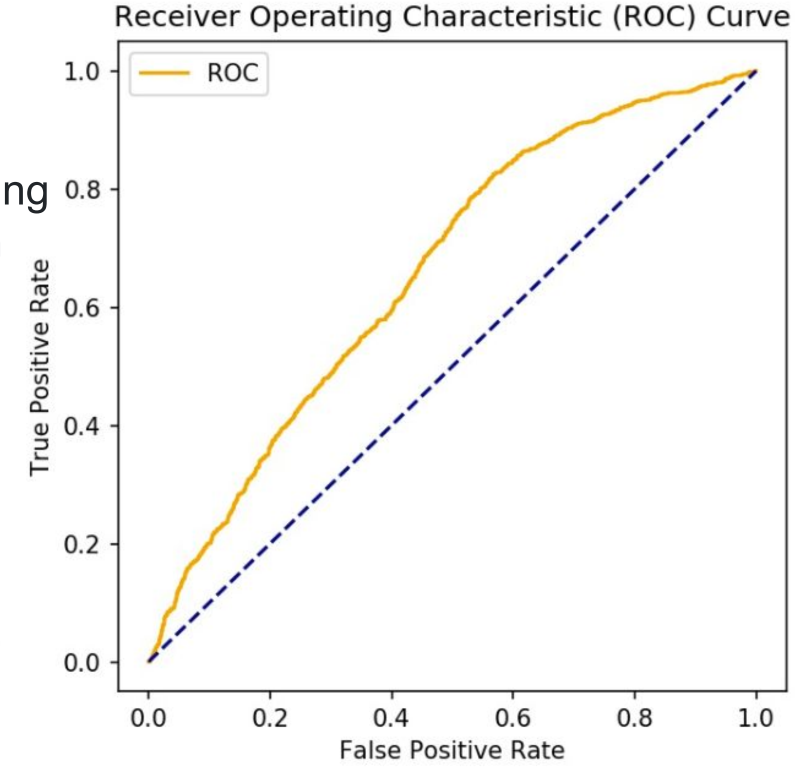
$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

using Bayesian probability terminology, the above equation can be written as

$$\text{Posterior} = \frac{\text{prior x likelihood}}{\text{evidence}}$$



Representation Only



Confusion Matrix

	Predicted A	Predicted T
Actual A	759	866
Actual T	269	990

Model Comparison Table

Rank	Model	Accuracy
1	Random Forest	74.45%
2	Logistic Regression	71.64%
3	Decision Tree	68.79%
4	KNN	62.76%
5	Naïve Bayes	60.64%

Conclusion

Based on the implementation of various models along with hyper parameter tuning, grid search with k-fold cross validation, we found that boosting algorithms predict with maximum accuracy

We inferred that most important factors that cause attrition are Job_Groups, Performance ratings of previous years, Job code, Experience and Annual rate which are inline with the general intuitions

If we had more data like Year Joined, Distance from home, Number of companies worked at, Years since last promotion etc , we can get more accurate predictions

Some job groups had 100% attrition which implies that these job group could have been shut down. We could make better predictions had we had more data, as this is one of the important factors in many algorithms

References:

- Lectures and Notes by Professor Christopher
- Larose, D. T., & Larose, C. D. (2014). Discovering knowledge in data: An introduction to data mining. Hoboken: Wiley.
- Galarnyk, M. (2020, May 01). PCA using Python (scikit-learn). Retrieved May 03, 2020, from https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60
- Chauhan, N. (2020, April 18). Real world implementation of Logistic Regression. Retrieved May 03, 2020, from https://towardsdatascience.com/real-world-implementation-of-logistic-regression-5136cefb8125
- Swaminathan, S. (2019, January 18). Logistic Regression - Detailed Overview. Retrieved May 03, 2020, from https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc
- S, Y. (2019, September 13). An Introduction to Naïve Bayes Classifier. Retrieved May 03, 2020, from https://towardsdatascience.com/introduction-to-na%C3%AFve-bayes-classifier-fa59e3e24aaf