Employee Attrition Prediction

NBA 6070 - Designing Data Product

Team 18

Raising Questions

- How can we predict whether an employee will leave our company?

- What are the most leading factors to employee attrition?

Our Objectives

- Establish credible ML model to predict and analyze employee attrition

- Discover actionable solutions to the talent loss

EDA (Explanatory Data Analysis)

Dataset Introduction

The dataset is transformed from a questionnaire of IBM employees, trying to uncover the factors that lead to employee attrition

Numerical Questions

Education
1 Below College
2 College
3 Bachelor
4 Master
5 Doctor

EnvironmentSatisfaction
1 Low
2 Medium
3 High
4 Very High

Age
____ (enter an integer)

Categorical Questions

Attrition
- ☐ Yes
- ☐ No

Department
- ☐ Healthcare Representative
- ☐ Human Resources
- ☐ Laboratory Technician
- ☐ Manager
- ☐ Manufacturing Director
- ... ...

Check variable types, missing values, duplicate values

```
HRdata.dtypes
```

```
Age                         int64
Attrition                  object
BusinessTravel             object
DailyRate                   int64
Department                 object
DistanceFromHome            int64
Education                   int64
EducationField             object
EmployeeCount               int64
EmployeeNumber              int64
EnvironmentSatisfaction     int64
Gender                     object
HourlyRate                  int64
JobInvolvement              int64
JobLevel                    int64
JobRole                    object
JobSatisfaction             int64
MaritalStatus              object
MonthlyIncome               int64
MonthlyRate                 int64
NumCompaniesWorked          int64
Over18                     object
OverTime                   object
PercentSalaryHike           int64
PerformanceRating           int64
RelationshipSatisfaction    int64
StandardHours               int64
StockOptionLevel            int64
TotalWorkingYears           int64
TrainingTimesLastYear       int64
WorkLifeBalance             int64
YearsAtCompany              int64
YearsInCurrentRole          int64
YearsSinceLastPromotion     int64
YearsWithCurrManager        int64
dtype: object
```

```
HRdata.isna().sum()
```

```
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               0
EmployeeNumber              0
EnvironmentSatisfaction     0
Gender                      0
HourlyRate                  0
JobInvolvement              0
JobLevel                    0
JobRole                     0
JobSatisfaction             0
MaritalStatus               0
MonthlyIncome               0
MonthlyRate                 0
NumCompaniesWorked          0
Over18                      0
OverTime                    0
PercentSalaryHike           0
PerformanceRating           0
RelationshipSatisfaction    0
StandardHours               0
StockOptionLevel            0
TotalWorkingYears           0
TrainingTimesLastYear       0
WorkLifeBalance             0
YearsAtCompany              0
YearsInCurrentRole          0
YearsSinceLastPromotion     0
YearsWithCurrManager        0
dtype: int64
```

```
HRdata.duplicated().sum()
```

```
0
```

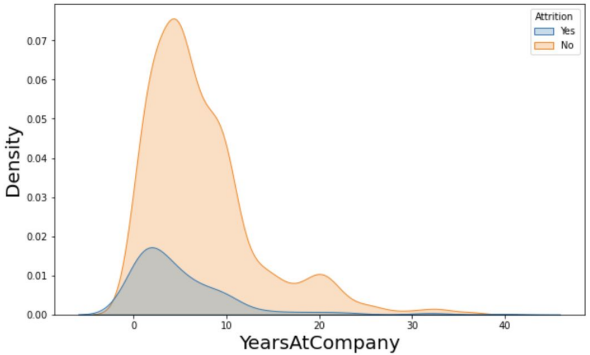
Consists of 35 columns and 1470 entries

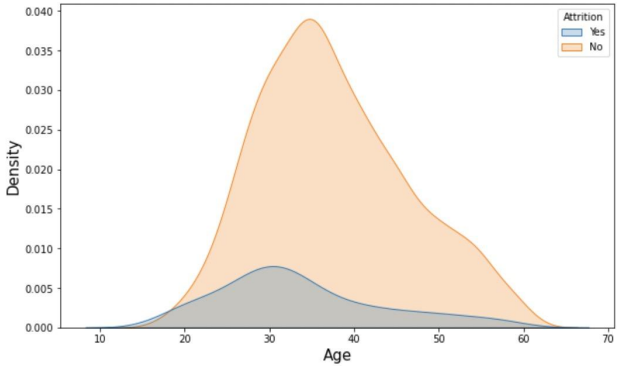
Pretty clean

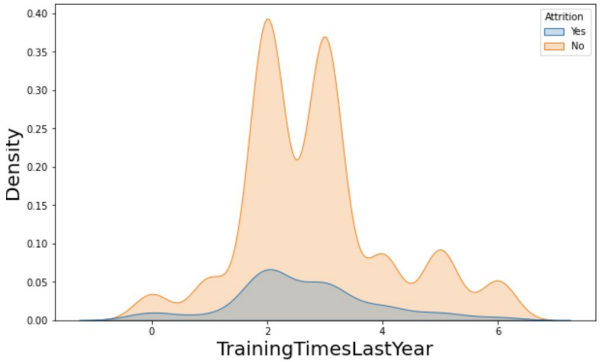
- ○ ***no missing values***
- ○ ***No duplicate values***

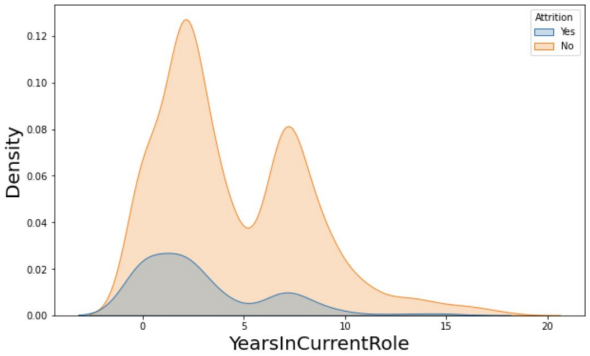
EDA - Feature Distribution

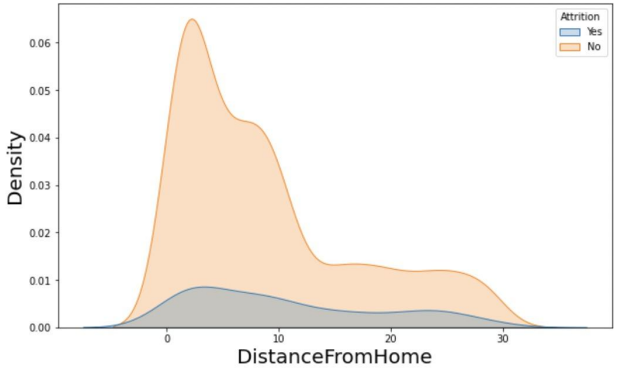
Kernel Density Plots





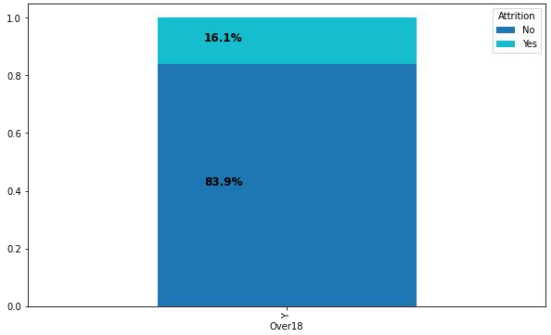


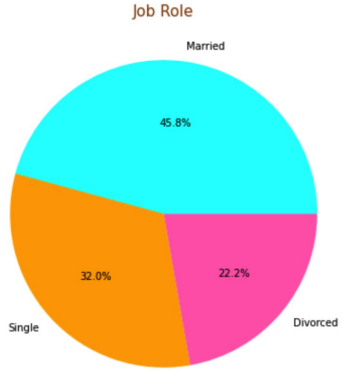


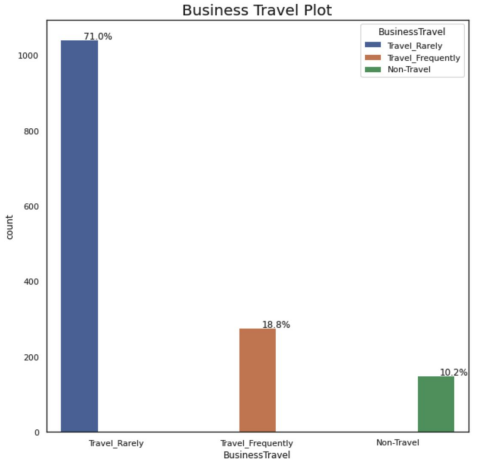


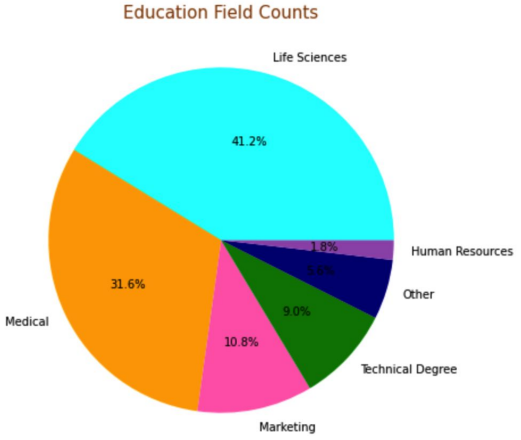
EDA - Feature Distribution

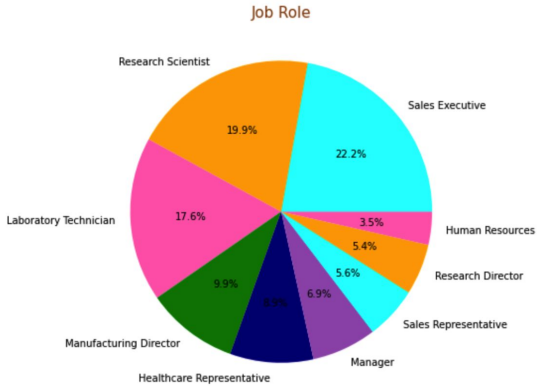
Pie Chart/ Percentage Bar Plots

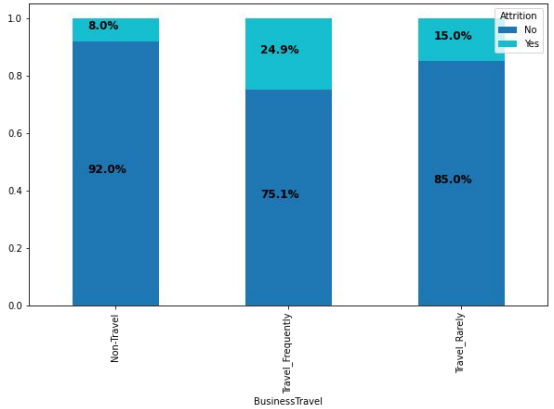






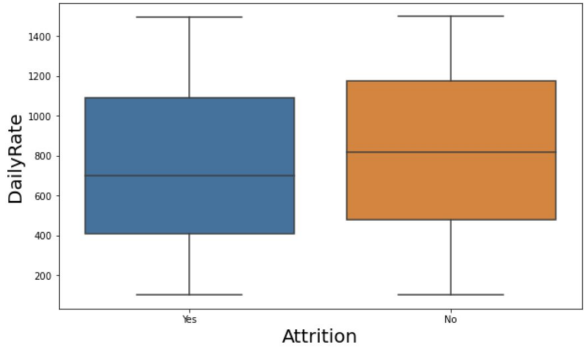


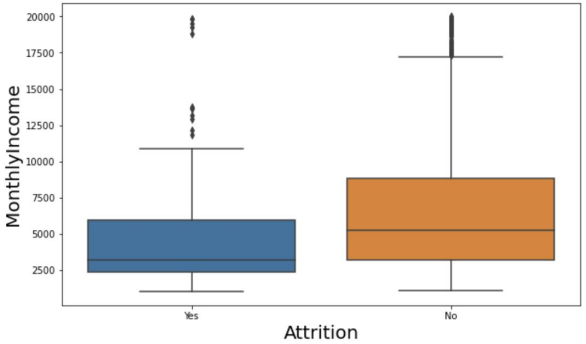


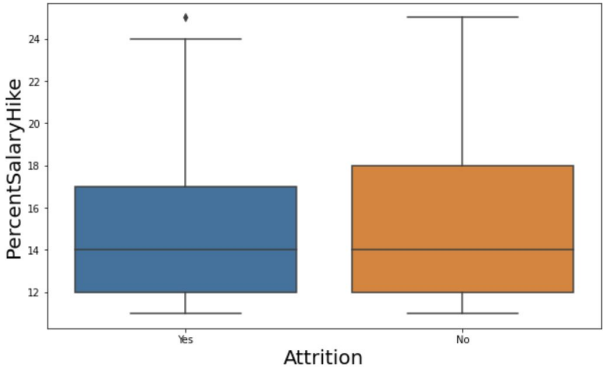


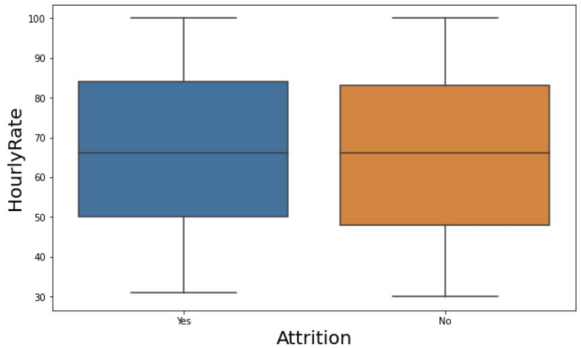
EDA - Feature Distribution

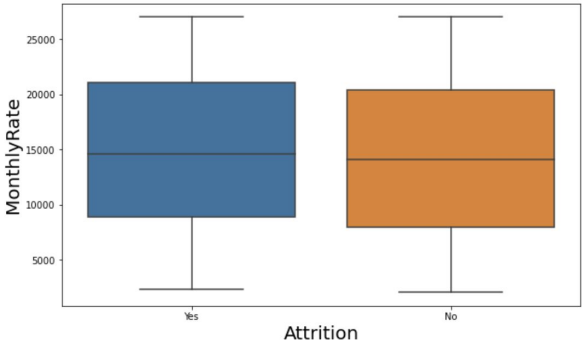
Box & Whisker Plots











Data Collection

Source of the Data

- Built up by IBM
- Missing and null value
- Survey designed

Merging Data

- Filtered and merged the data
- Still had some problem

Cleaning Data

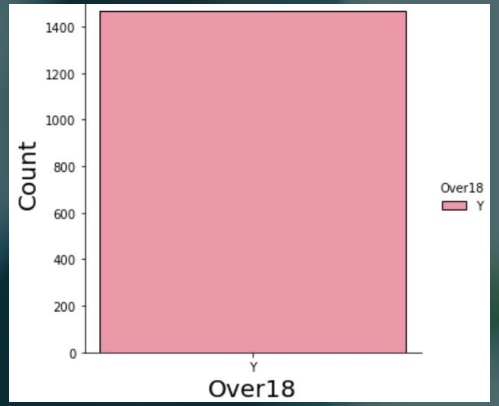
- Delete 3 imbalanced independent variables
- Handle imbalanced data (evaluate different methods)
  1. Use oversampling to balance skew dependent variable (Attrition)
  2. Undersampling
  3. Create dummy variables for categorical variables
  4. Feature engineering

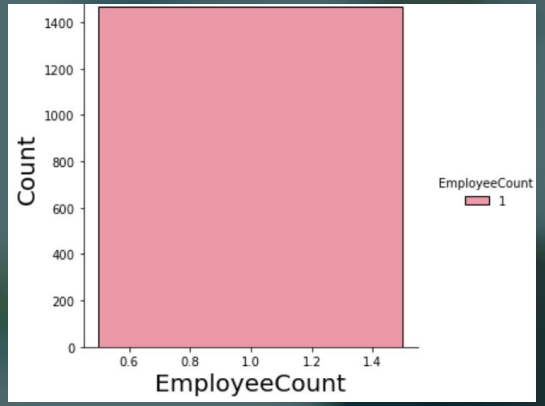
Competitive Advantage of the Data

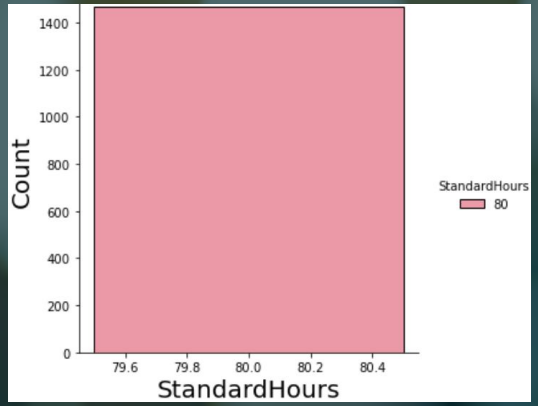
- The source of data is stable
- The data is highly predictable
- The amount of data will continue to increase
- The accuracy of the data can be enriched through cross-comparison.

Feature Engineering

Step 1 - Delete 3 Imbalanced Independent Variables



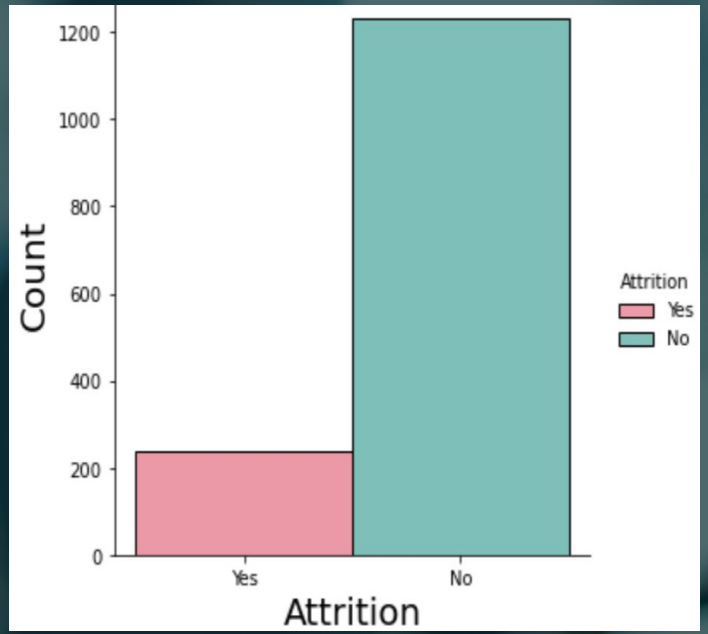




There are three irrelevant variables having only one value no matter whether the employee leave or not, which won't be contributed to our prediction.

Feature Engineering

Step 2 - Address Imbalanced Dependent Variable



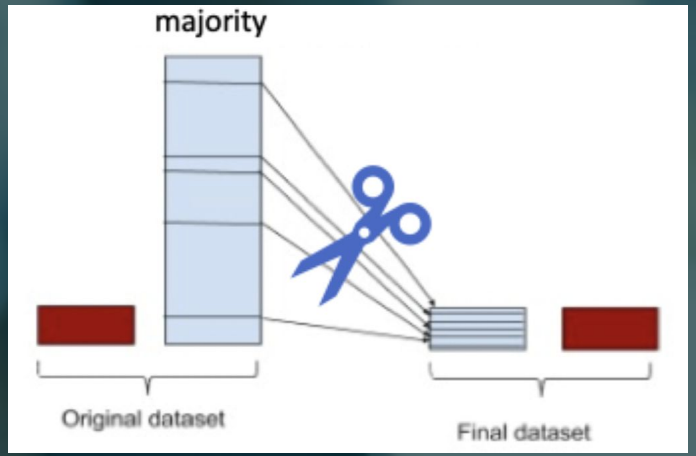
Dataset Imbalance: *The leaving employees are fewer than those who stay, leading to an uneven dataset.*

- **Possible Reasons**:
  - The sampling bias when choosing interviewees
  - The difficulty to survey people who drop out
  - …
- **Consequence**: Classifiers tend to make biased learning models → Poor predictive performance
- **Solution**: **Resampling**

Feature Engineering

Step 2 - Address Imbalanced Dependent Variable

Undersampling → Not feasible



Strategy: Randomly remove samples from the majority class

Why didn't we choose undersampling?

- The chosen samples from majority class may be biased and unable to represent the real world
- It may leave out instances that provide important information
- The remaining data after undersampling (200+) will be insufficient for building models
- …

Feature Engineering

Step 2 - Address imbalanced dependent variable

Oversampling - Two methods

Random Oversampling

ACTUAL VS. PREDICTED	No	Yes	ACTUAL	RECALL
No	197	54	251	78.49%
Yes	13	30	43	69.77%
PREDICTED	210	84	294	74.13% AVG. RECALL
PRECISION	93.81%	35.71%	64.76% AVG. PRECISION	77.21% ACCURACY

- Strategy: Randomly select samples with replacement from the minority class, and add them to training set
- Test result: Overall accuracy = **77.21%**

SMOTE algorithm

ACTUAL VS. PREDICTED	0	1	ACTUAL	RECALL
0	230	21	251	91.63%
1	20	23	43	53.49%
PREDICTED	250	44	294	72.56% AVG. RECALL
PRECISION	92.00%	52.27%	72.14% AVG. PRECISION	86.05% ACCURACY

- Strategy: Synthesize new samples from the existing samples, and add them to training set
- Test result: Overall accuracy = **86.05%**

```
HRdata_train['Attrition'].value_counts()

No     982
Yes    194
Name: Attrition, dtype: int64
```

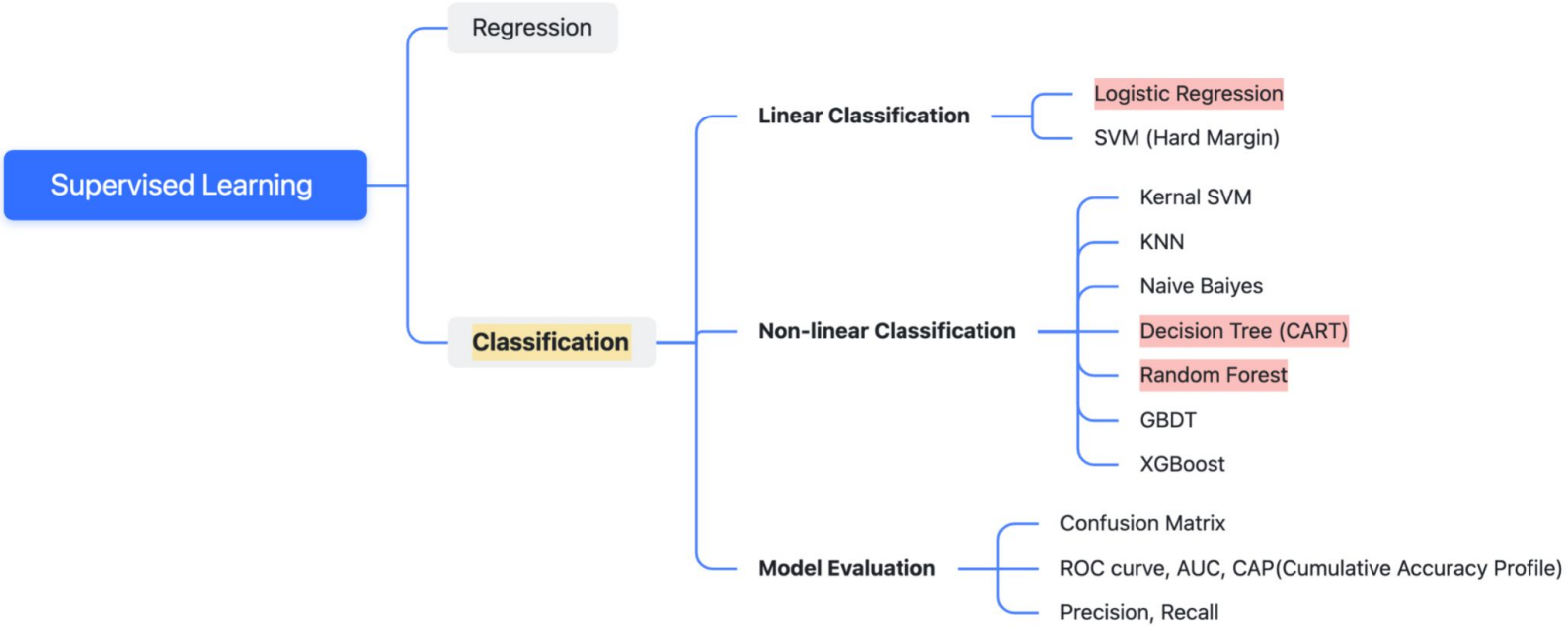
```
HRdata_train_res_y.value_counts()

1    982
0    982
Name: Attrition_Yes, dtype: int64
```

Choosing The Right Model

Our objectives:

1) Establish credible ML model to predict and analyze employee attrition
2) Discover actionable solutions to the talent loss



Decision Tree (First Try)

Library used: pandas, scikit-learn, seaborn, matplotlib

Step 1: get training and test data

```
train = pd.read_csv('HRdata_train_oversampled_smote.csv')
```

```
test = pd.read_csv('HRdata_test_dummied_smote.csv')
```

```
x_train = train.drop(columns=['Attrition_Yes'])
y_train = train['Attrition_Yes']
```

```
x_test = test.drop(columns=['Attrition_Yes'])
y_test = test['Attrition_Yes']
```

*We processed our data before by removing trivial variables and applying SMOTE. The training/test data split is 80/20.

Step 2: create decision tree model

```
clf = tree.DecisionTreeClassifier(random_state=0)
clf.fit(x_train,y_train)
y_train_pred = clf.predict(x_train)
y_test_pred = clf.predict(x_test)
```

Step 3: visualize decision tree

```
plt.figure(figsize=(20,20))
features = x_train.columns
classes = ['Attrition No','Attrition Yes']
tree.plot_tree(clf,feature_names=features,class_names=classes,filled=True)
plt.show()
```

Step 4: evaluate the model

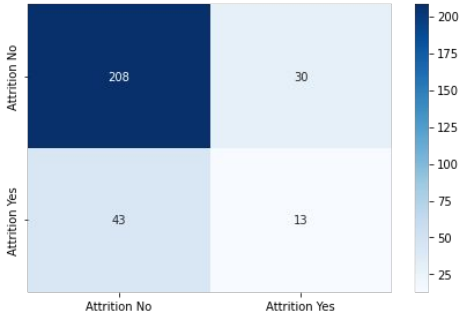
```
# helper function
def plot_confusionmatrix(y_train_pred,y_train,dom):
    print(f'{dom} Confusion matrix')
    cf = confusion_matrix(y_train_pred,y_train)
    sns.heatmap(cf,annot=True,yticklabels=classes
                ,xticklabels=classes,cmap='Blues', fmt='g')
    plt.tight_layout()
    plt.show()
```

```
print(f'Train score {accuracy_score(y_train_pred,y_train)}')
print(f'Test score {accuracy_score(y_test_pred,y_test)}')
plot_confusionmatrix(y_train_pred,y_train,dom='Train')
plot_confusionmatrix(y_test_pred,y_test,dom='Test')
```

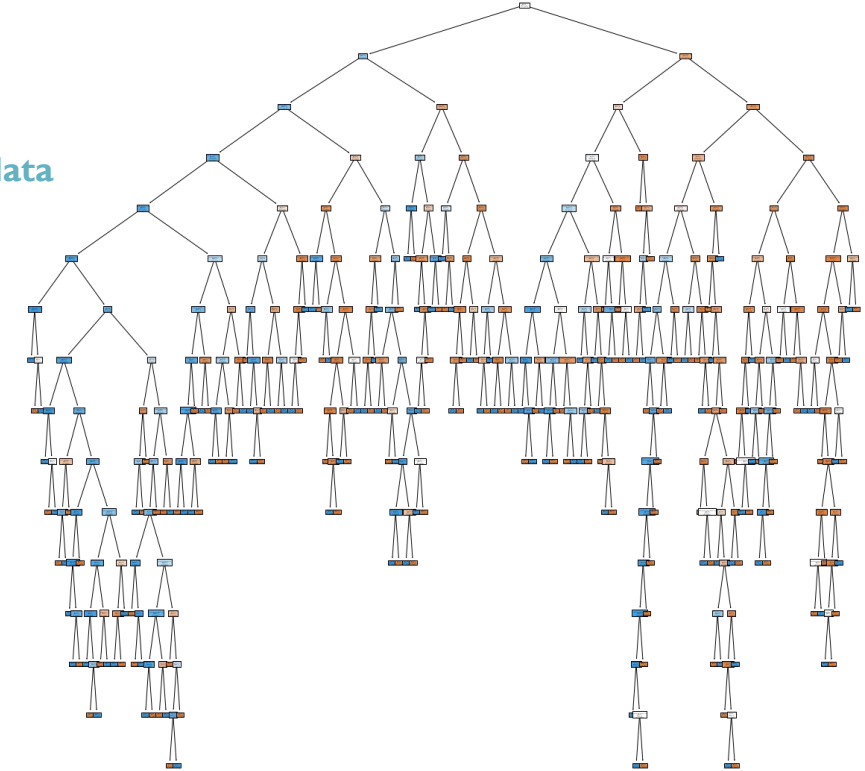
Decision Tree (First Try)

Problem: our decision tree overfitted the training data

Training Dataset
Accuracy: 100%



Test Dataset
Accuracy: 75.17%

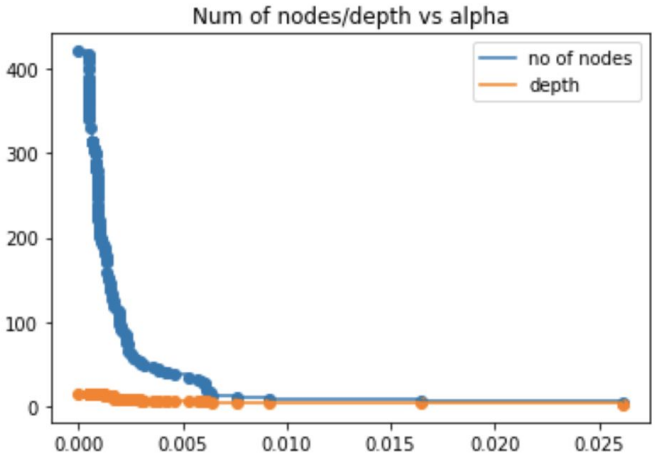


Decision Tree (Second Try with Cost Complexity Pruning)

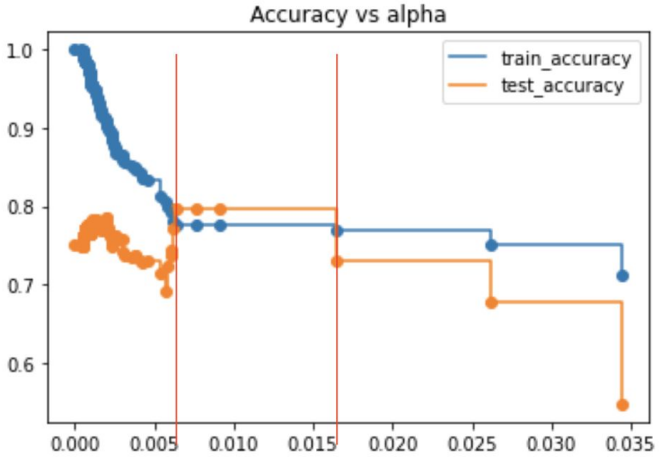
Basic idea: find an alpha value* that gives us the best accuracy on test dataset

* different alpha values correspond to different levels of model complexity (i.e. num of nodes & tree depth)

Step 1: visualize the relationship between num of node/depth and alpha



Step 2: identify the best alpha value(s)

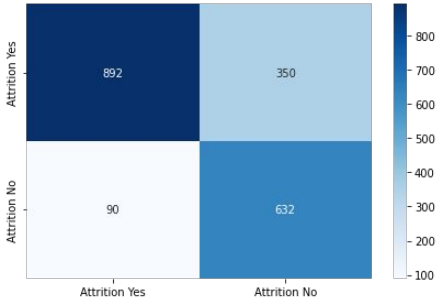


The best accuracy on test data can be achieved with an alpha value between 0.006 to 0.016

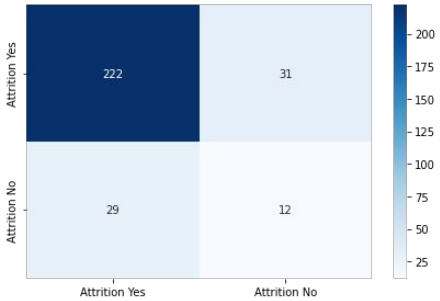
* Implementation details can be found in our report

Decision Tree (Second Try with Cost Complexity Pruning)

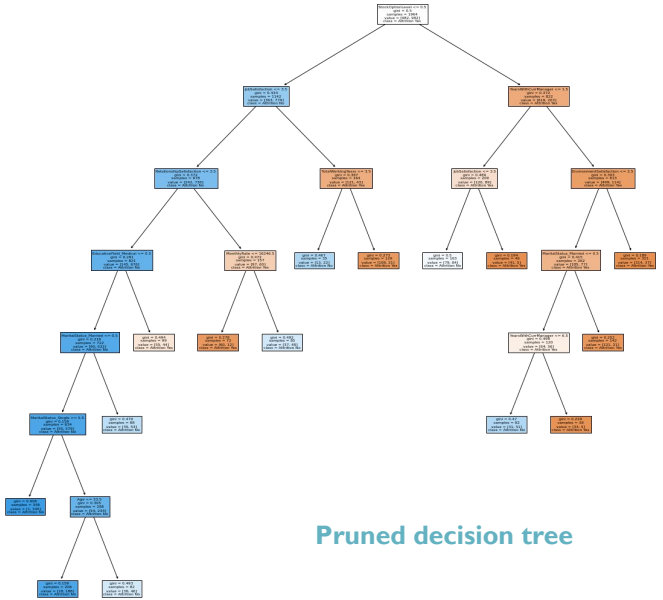
With an alpha value of 0.009, we can visualize the decision tree and evaluate the model again



Training Dataset
Accuracy: 77.6%

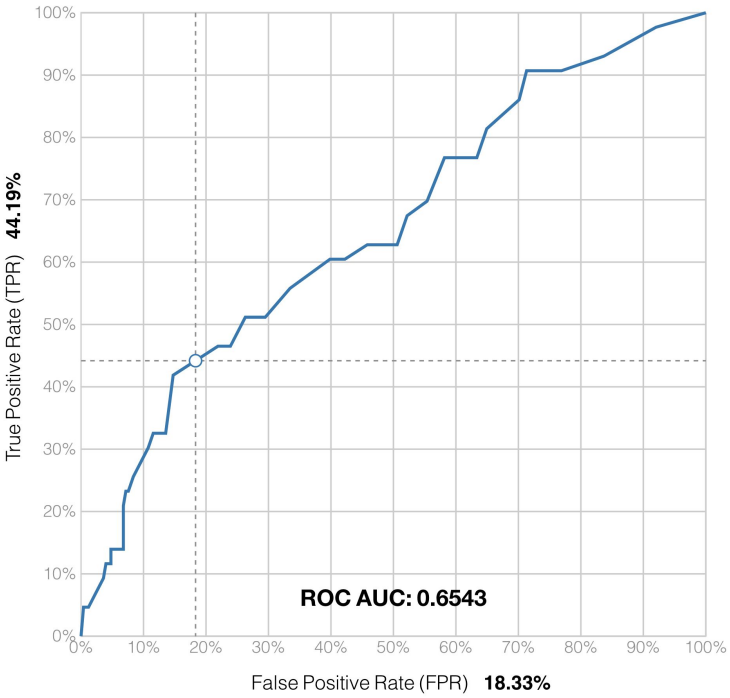


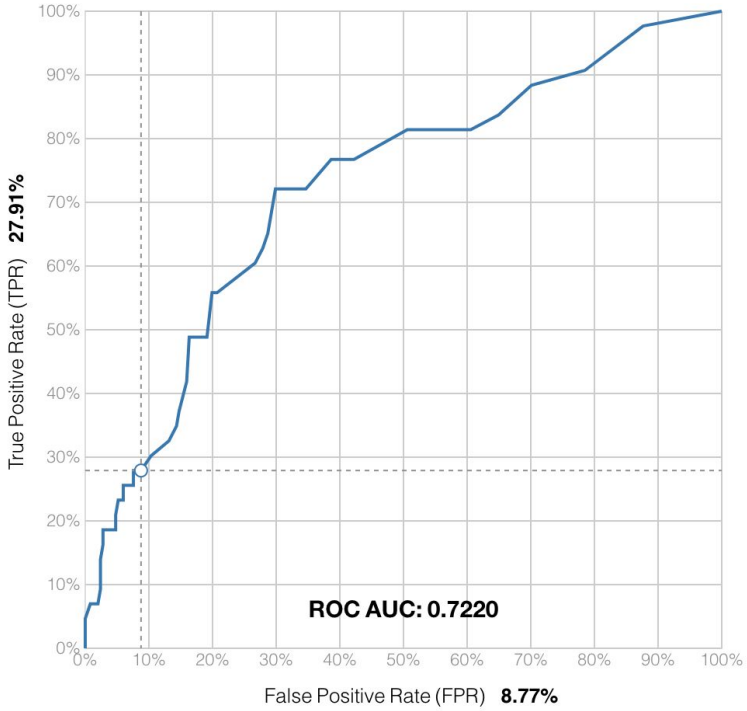
Test Dataset
Accuracy: 79.6%
(Original: 75.17%)



Pruned decision tree

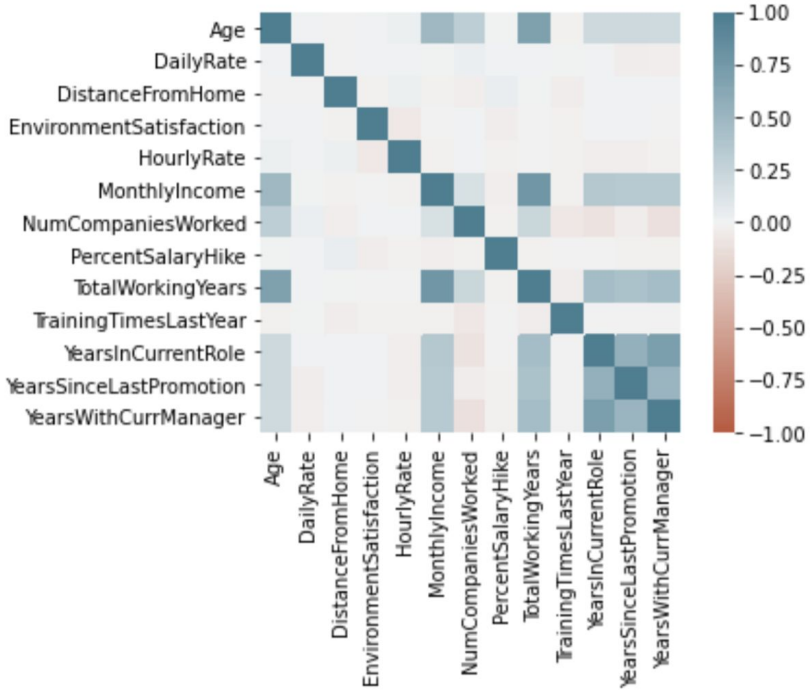
Decision Tree (ROC AUC Comparison)





Logistic Regression

Split train and test dataset_SMOTE _*Using R*



Logistic Regression

Split train and test dataset_SMOTE _*Using SPSS*

ANOVA[a]

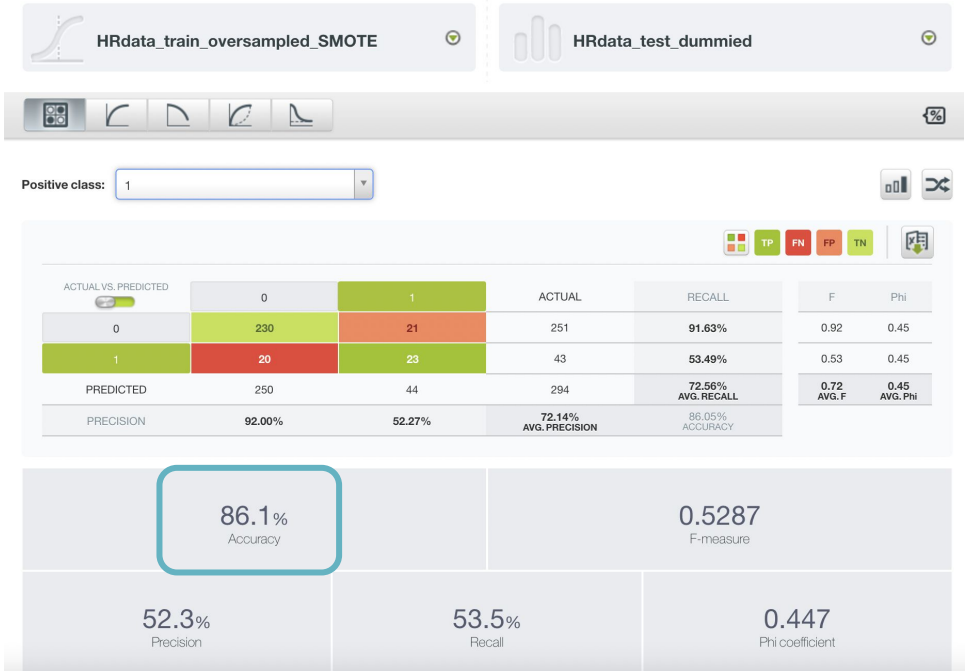
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	312.139	44	7.094	76.112	.000[b]
	Residual	178.861	1919	.093		
	Total	491.000	1963			

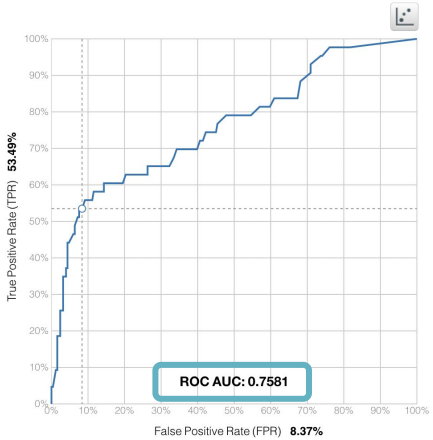
Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.797[a]	.636	.627	.305

Logistic Regression

Split train and test dataset_SMOTE _*Using BigML*

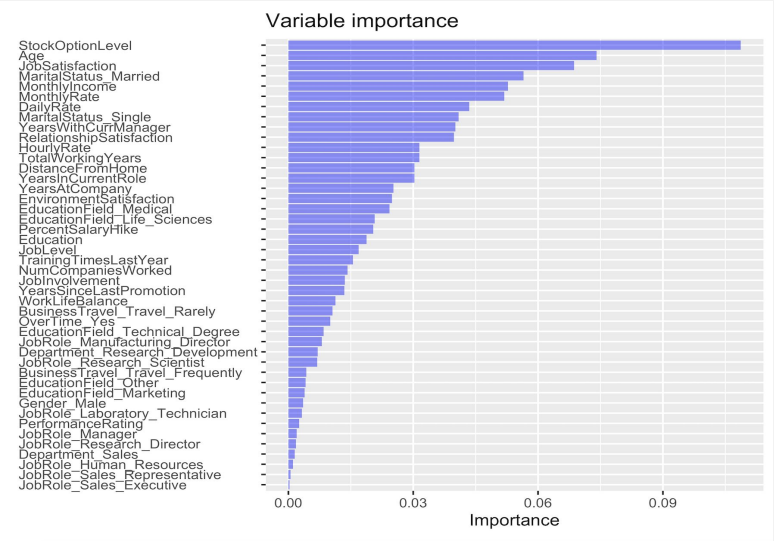




Visualize and Communicate Insights

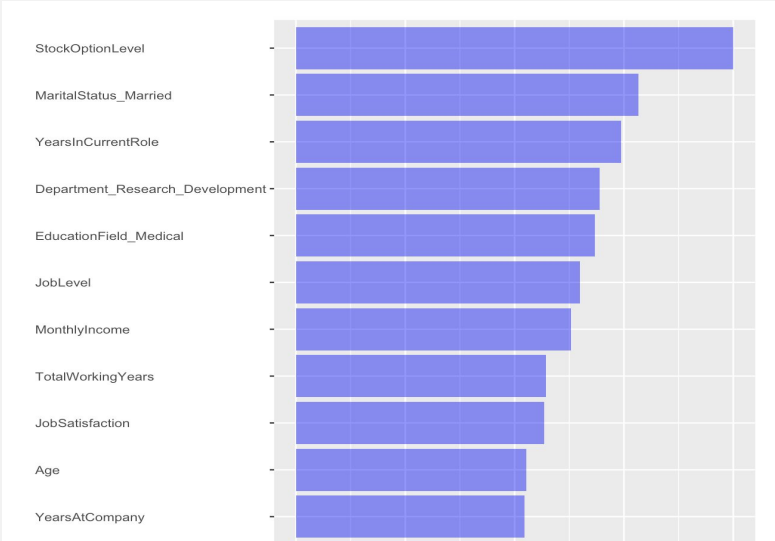
Factor Importance Analysis

Classification Tree Model



Important Significant Variables: *StockOptionLevel, Age, JobSatisfaction, Marital Status, Monthly Income, Monthly / Daily Rate*

Random Forest Model

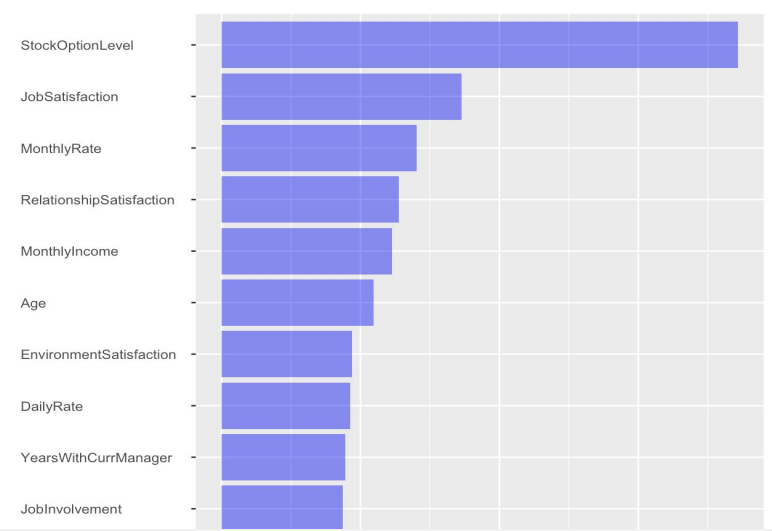


Important Significant Variables: *StockOptionLevel, Marital Status, YearsInCurrentRole, Department, EducationField, Job Level*

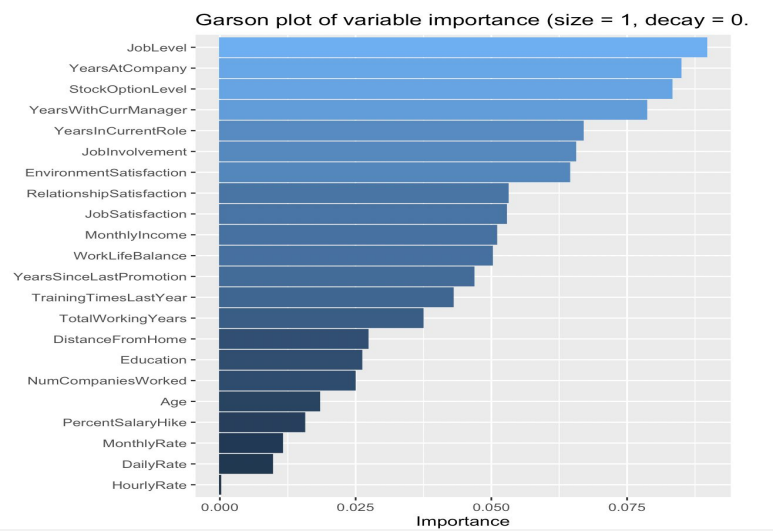
Visualize and Communicate Insights

Factor Importance Analysis

Gradient Boosted Trees Model



Neutral Network (Garson) Model



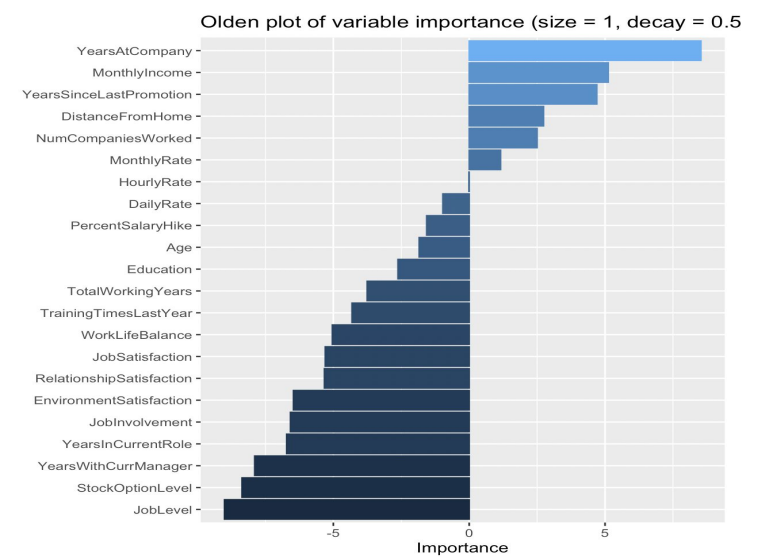
Important Significant Variables: *StockOptionLevel, Age, Job Satisfaction, Marital Status, Monthly Income, Monthly / Daily Rate, Relationship Safisfaction*

Important Significant Variables: *Job level, Years at company, StockOptionLevel, Yeas with current manager, Years in Current role, Job involvement*

Visualize and Communicate Insights

Factor Importance Analysis

Neutral Network (Olden) Model



Stock Option Level

Monthly / Annual Income

Relationship Satisfaction

Environment Satisfaction

Job Involvement

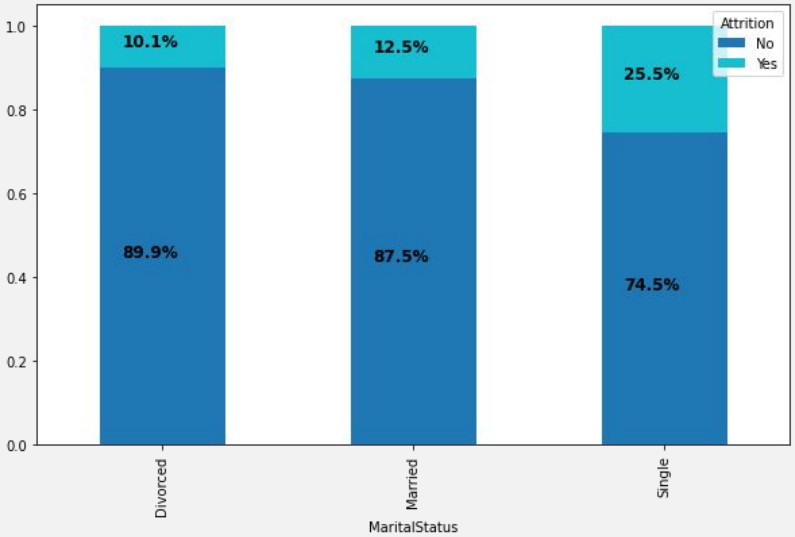
Age / Years at Company

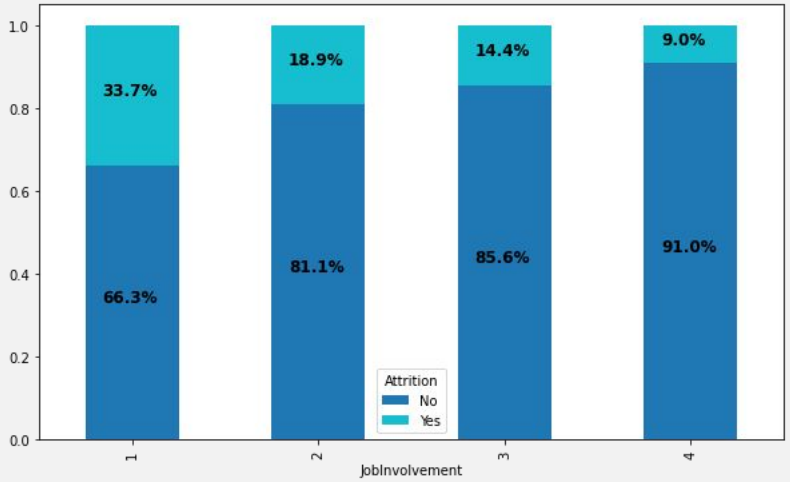
Marital Status

Important Significant Variables: *Years at company, Monthly Income, Years Since last promotion, Distance from Home, Monthly Rate*

Visualize and Communicate Insights

Indicators showing strong linear relationships with Attrition





The people who are in marriage or just divorced with their partner are more likely to stay in their company instead of resign.

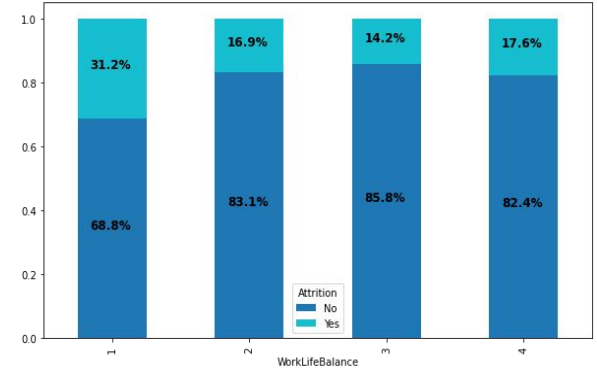
The more people get involved in their current occupation, the less likely that they are going to resign.

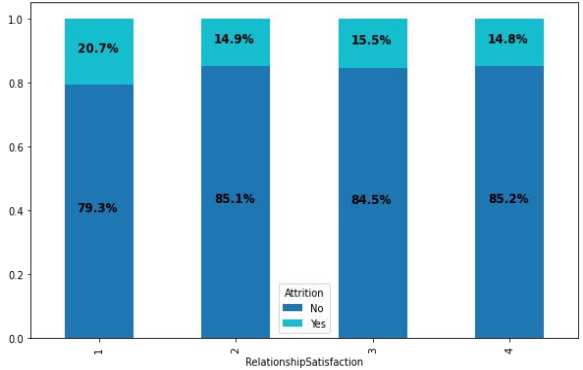
Visualize and Communicate Insights

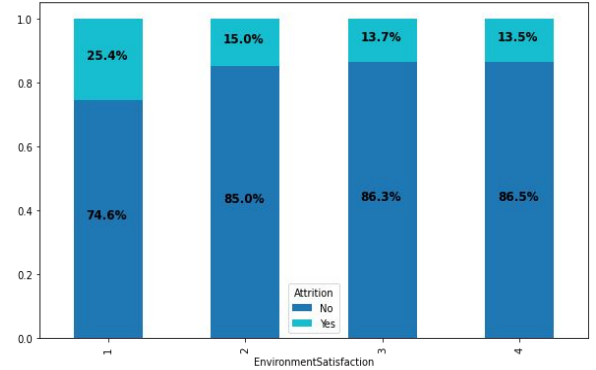
Some metrics play as thresholds for the turnover rate. *From 1 to 2 of the metrics, the talent loss rate decreases by large percent, but from 2 onward, improvement in those metrics has little or negative margin contribution on talent retention.*

Those threshold-like metrics include *WorkLifeBalance, RelationshipSatisfaction, and EnvironmentSatisfaction, etc.*

The implication of this finding for companies: *with limited budgets, companies should retain their investment in those metrics as long as metrics reach their thresholds, as the input-output effect of those factors is limited.*

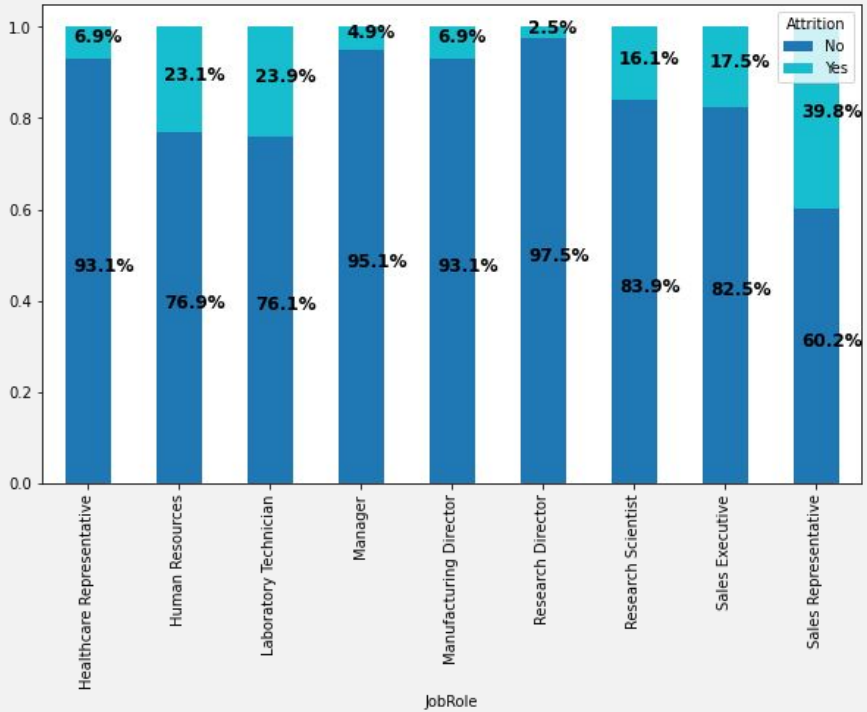






Visualize and Communicate Insights

Uneven Attrition Rates Across Different Departments



- *Departments Sales Representative (39.8%), Lab Technician (23.9%), Human Resources (23.1%) and Sales Executive (17.5%) and Research Scientists (16.1%) have higher turnover rate than the company average of 16.1%. the posible high attrition rate of core technician and research teams requires company's attention.*
- *Departments as Healthcare Representative (6.9%), Research Director (2.5%), Manager (4.9%) and Manufacturing Director (6.9%), need to reflect on whether we are offering much higher salaries than the industry average*

Productionalize

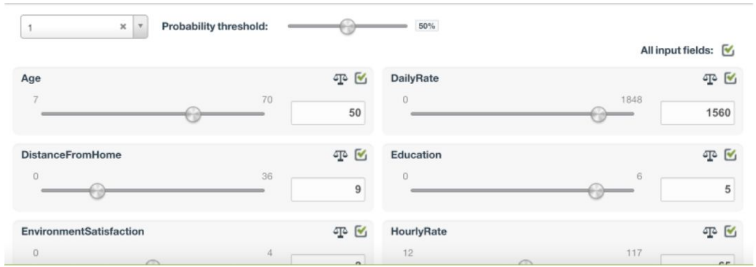
Potential products and services:

1. Feature subscriptions
2. Consulting services

Some advice to HR to reduce employee attrition rate:

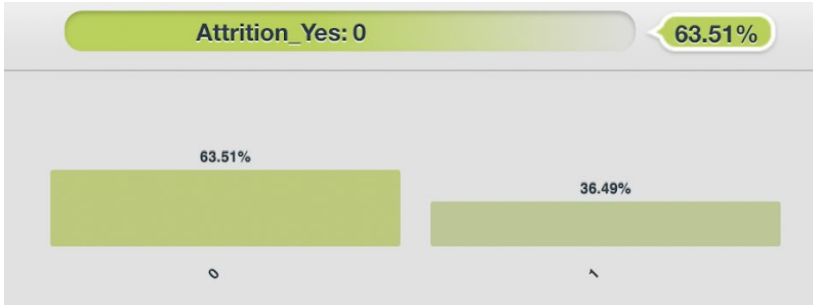
1. Equity and Salary Enhancement (Material Security)
2. Employee Satisfaction (environment and relationships), Job Involvement and Level
3. Age, Seniority and Years in the company, and their own Marital Status are also decisive factors affecting their resignation

Prediction with Model



Give & Reward?

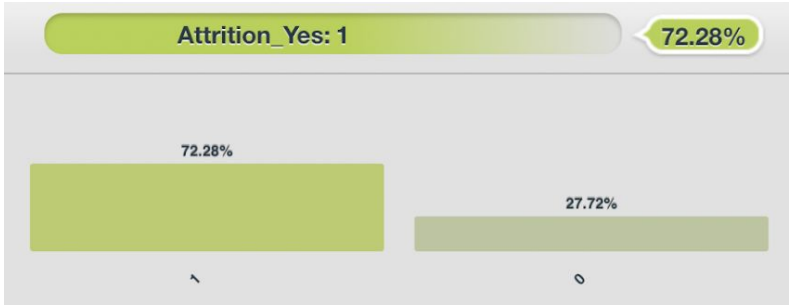
Profile 1:
R&D in Life Sciences field, Age 35, Ordinary job position



Result: Relatively high job involvement with lower pay people are more likely to stay, which is counterintuitive

High position?

Profile 2:
Sales in Marketing field, Age 50, High job level, High salary



Result: People with high position in company tend to leave, which may because job-hopping is more normal among high-level employee

Team 18

Biao Wang (bw437)
M.Eng. Engineering Management

Yu Pan (yp425)
M.Eng. Engineering Management

Shiyu Pan (sp2435)
M.P.S. in Real Estate

Yajing Wang (yw832)
M.Eng. Engineering Management

Steven He (wh383)
M.Eng. Engineering Management

THANK YOU!

Team 18

Q&A?
Email yw832@cornell.edu