Employee Attrition

Dominic Ventura

Data

Fictional IBM Employee Data

1,470 Employees

31 columns

Target variable - Employee Attrition = Yes (employee left) or No (employee didn’t leave)

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

Job Involvement
1 'Low'
2 'Medium'
3 'High'
4 'Very High'

Job Satisfaction
1 'Low'
2 'Medium'
3 'High'
4 'Very High'

Performance Rating
1 'Low'
2 'Good'
3 'Excellent'
4 'Outstanding'

Relationship Satisfaction
1 'Low'
2 'Medium'
3 'High'
4 'Very High'

Work-Life Balance
1 'Bad'
2 'Good'
3 'Better'
4 'Best'

Categorical Variable Key

Attrition : ['Yes' 'No']

No	1233
Yes	237

BusinessTravel : ['Travel_Rarely' 'Travel_Frequently' 'Non-Travel']

Travel_Rarely	1043
Travel_Frequently	277
Non-Travel	150

EducationField : ['Life Sciences' 'Other' 'Medical' ' Marketing' 'Technical Degree' 'Human Resources']

Life Sciences	606
Medical	464
Marketing	159
Technical Degree	132
Other	82
Human Resources	27

Department : ['Sales' 'Research & Development' 'Human Resources']

Research & Development	961
Sales	446
Human Resources	63

JobRole : ['Sales Executive' 'Research Scientist' 'Laboratory Technician' ‘'Manufacturing Director' 'Healthcare Representative' 'Manager' 'Sales Representative' 'Research Director' 'Human Resources']

Sales Executive	326
Research Scientist	292
Laboratory Technician	259
Manufacturing Director	145
Healthcare Representative	131
Manager	102
Sales Representative	83
Research Director	80
Human Resources	52

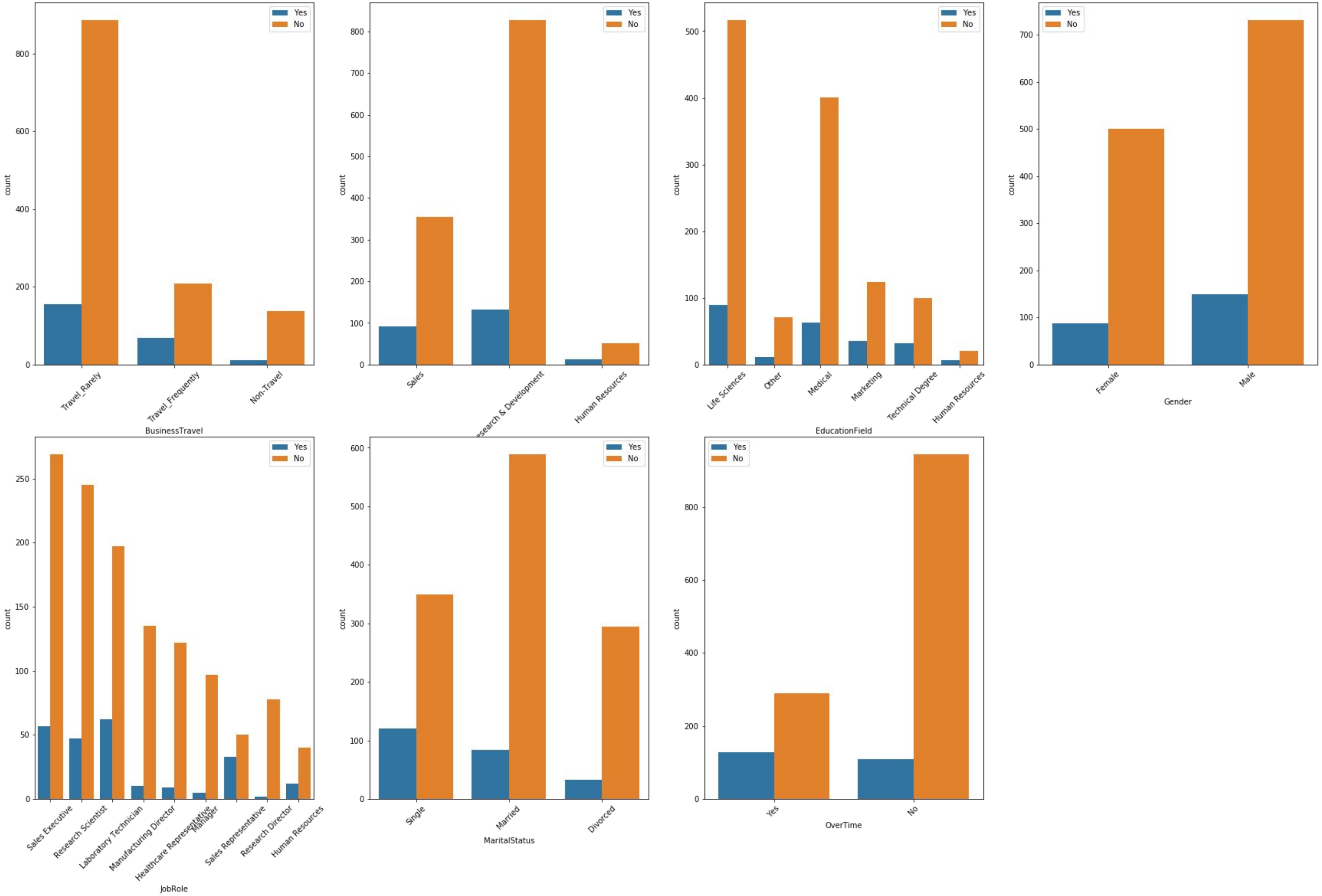
Gender : ['Female' 'Male']

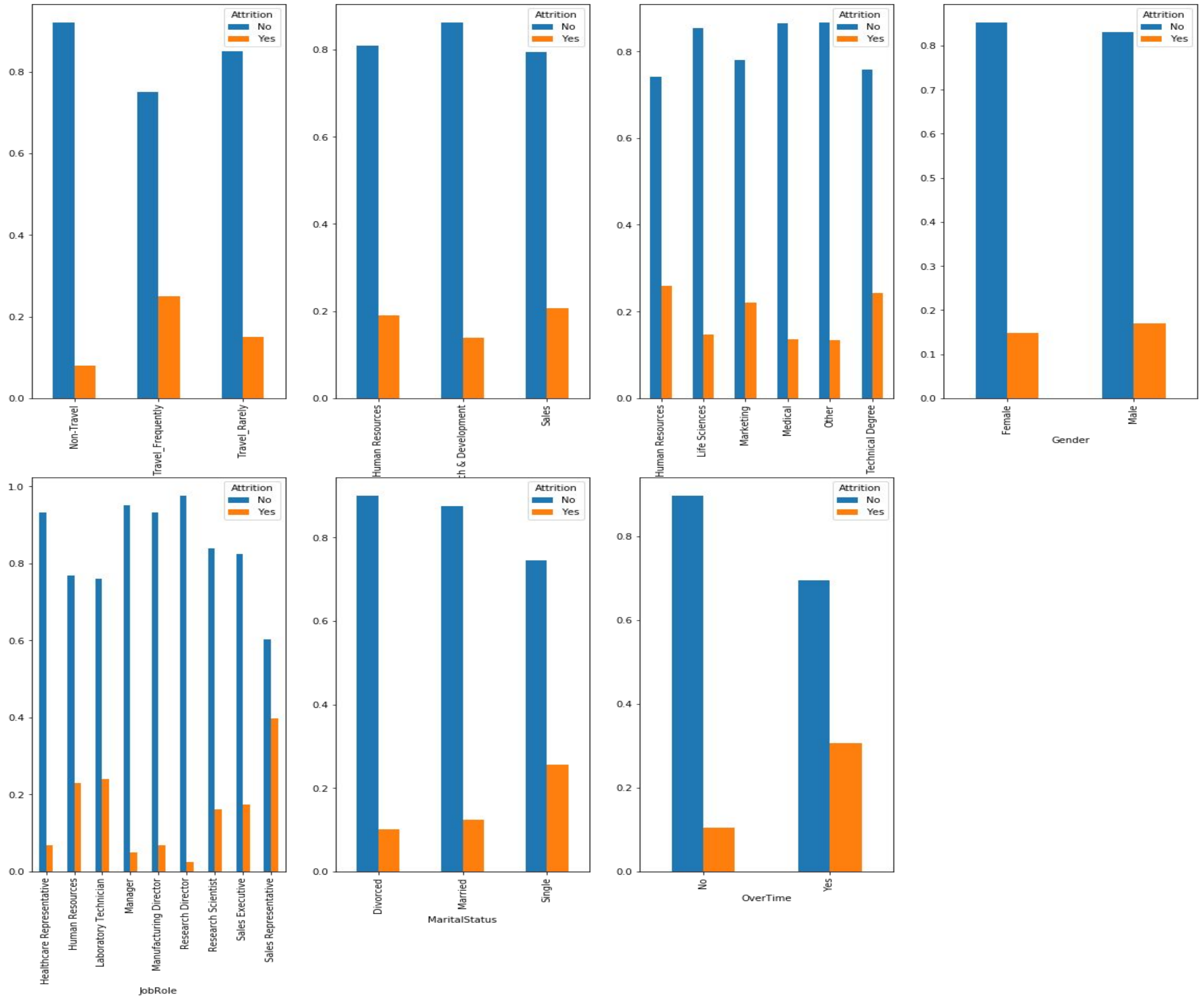
Male	882
Female	588

MaritalStatus : ['Single' 'Married' 'Divorced']

Married	673
Single	470
Divorced	327


Age
DailyRate
DistanceFromHome
Education
EnvironmentSatisfaction
HourlyRate
JobInvolvement
JobLevel
JobSatisfaction
MonthlyIncome
MonthlyRate
NumCompaniesWorked
PercentSalaryHike
PerformanceRating
RelationshipSatisfaction
StockOptionLevel
TotalWorkingYears
TrainingTimesLastYear
WorkLifeBalance
YearsAtCompany
YearsInCurrentRole
YearsSinceLastPromotion
YearsWithCurrManager


Yes
No
count
Travel_Rarely
Travel_Frequently
Non-Travel
BusinessTravel
Sales
esearch & Development
Human Resources
Life Sciences
Other
Medical
Marketing
Technical Degree
Human Resources
EducationField
Female
Male
Gender
Sales Executive
Research Scientist
Laboratory Technician
Manufacturing Director
Healthcare Representative
Manager
Sales Representative
Research Director
Human Resources
JobRole
Single
Married
Divorced
MaritalStatus
Yes
No
OverTime


Attrition
No
Yes
Non-Travel
Travel_Frequently
Travel_Rarely
Human Resources
ch & Development
Sales
Human Resources
Life Sciences
Marketing
Medical
Other
Technical Degree
Female
Male
Gender
Healthcare Representative
Human Resources
Laboratory Technician
Manager
Manufacturing Director
Research Director
Research Scientist
Sales Executive
Sales Representative
JobRole
Divorced
Married
Single
MaritalStatus
No
Yes
OverTime

Business Travel: People who traveled frequently were more likely to leave, but those who rarely traveled were more likely to leave.

Department: If you worked in the Research & Development Department, you were less likely to leave your job.

Educational Field: Workers with degrees involved with HR, Technical, and Marketing were more likely to quit than the others.

Gender: Men were more likely to leave their job than women.

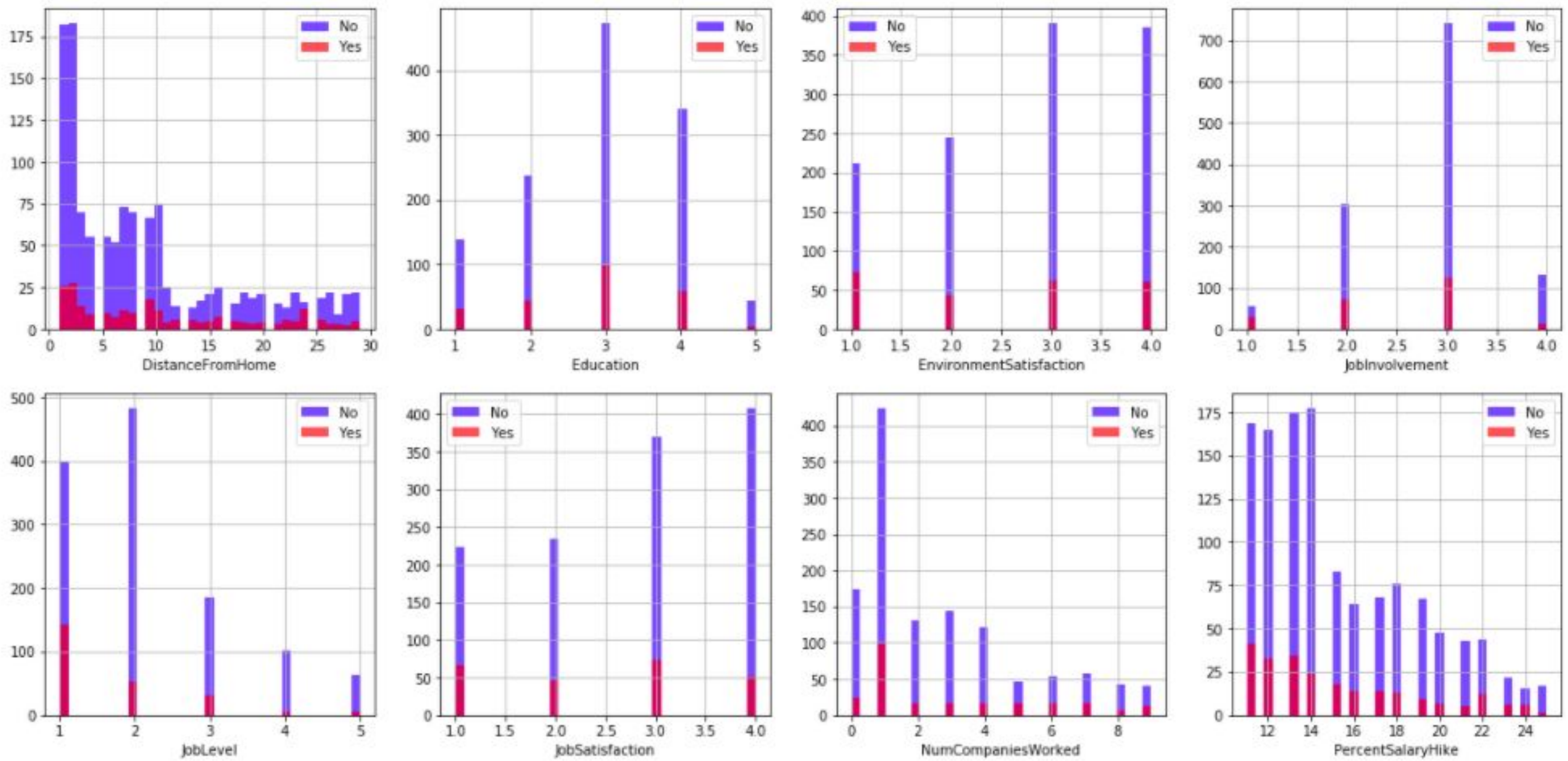
Job Role: Sales Representative, Lab Technician, and Human Resources were more likely to leave their job.

Marital Status: Single people were more likely to leave their job.

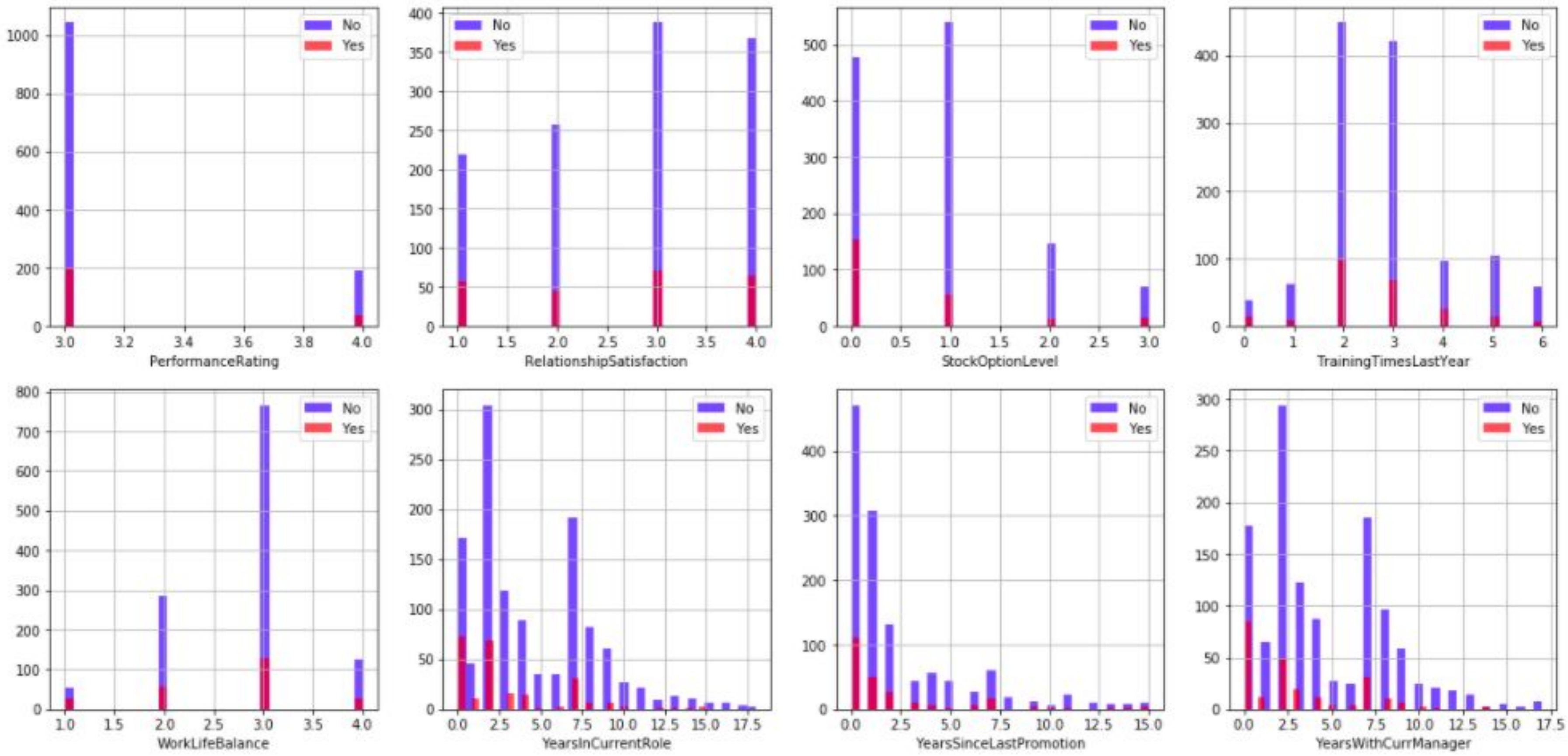
Over Time: Those that worked more hours were more likely to leave.

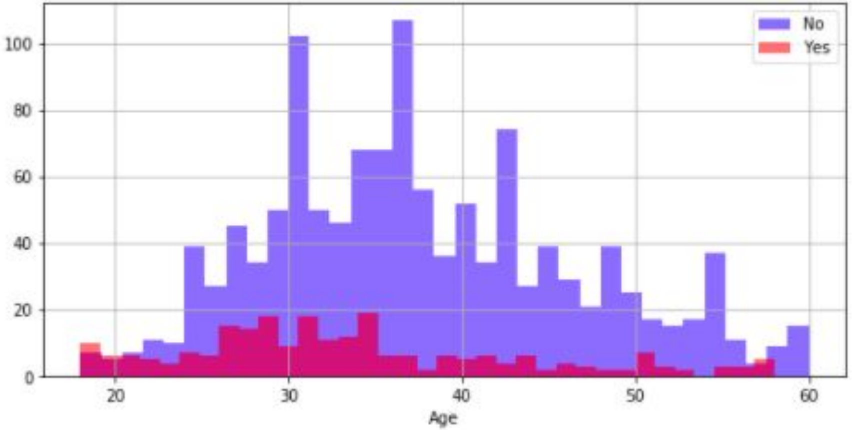
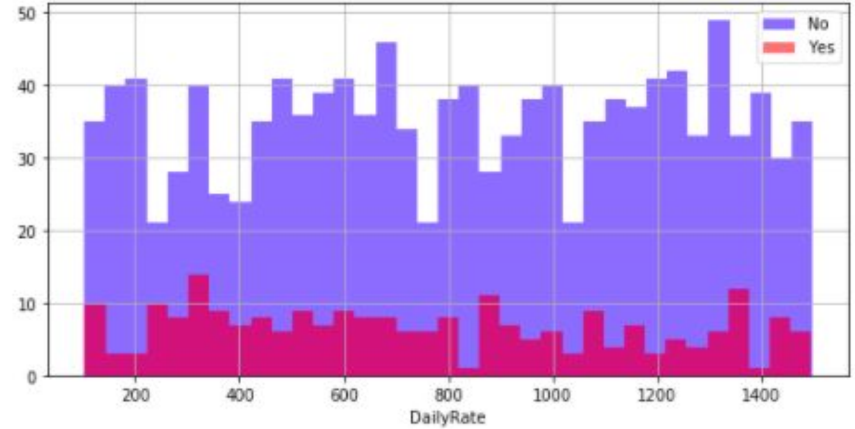
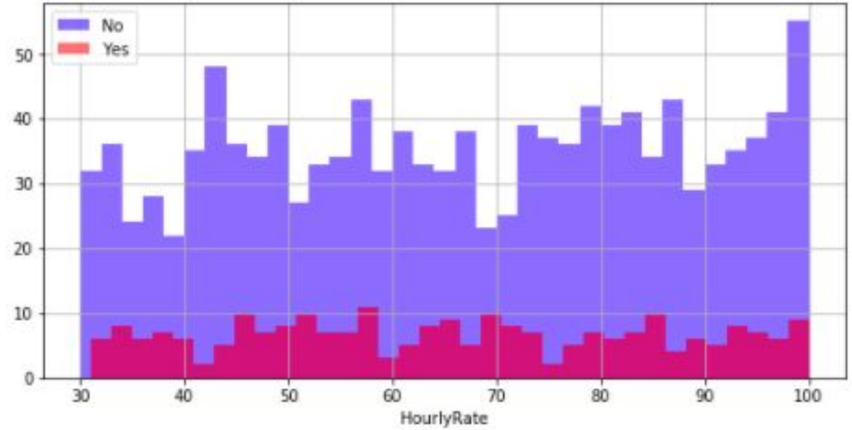
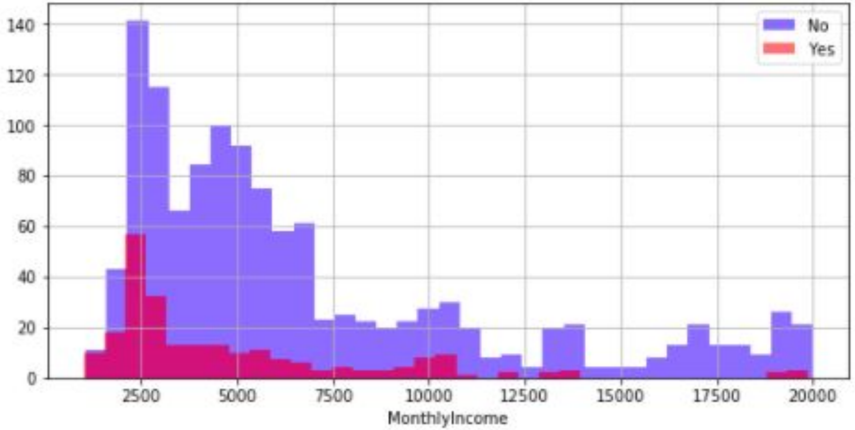
```
Attrition        No       Yes
Gender
Female     0.852041  0.147959
Male       0.829932  0.170068
All        0.838776  0.161224
=============================================
Attrition                        No       Yes
JobRole
Healthcare Representative  0.931298  0.068702
Human Resources            0.769231  0.230769
Laboratory Technician      0.760618  0.239382
Manager                    0.950980  0.049020
Manufacturing Director     0.931034  0.068966
Research Director          0.975000  0.025000
Research Scientist         0.839041  0.160959
Sales Executive            0.825153  0.174847
Sales Representative       0.602410  0.397590
All                        0.838776  0.161224
=============================================
Attrition           No       Yes
MaritalStatus
Divorced      0.899083  0.100917
Married       0.875186  0.124814
Single        0.744681  0.255319
All           0.838776  0.161224
=============================================
Attrition        No       Yes
OverTime
No         0.895636  0.104364
Yes        0.694712  0.305288
All        0.838776  0.161224
```

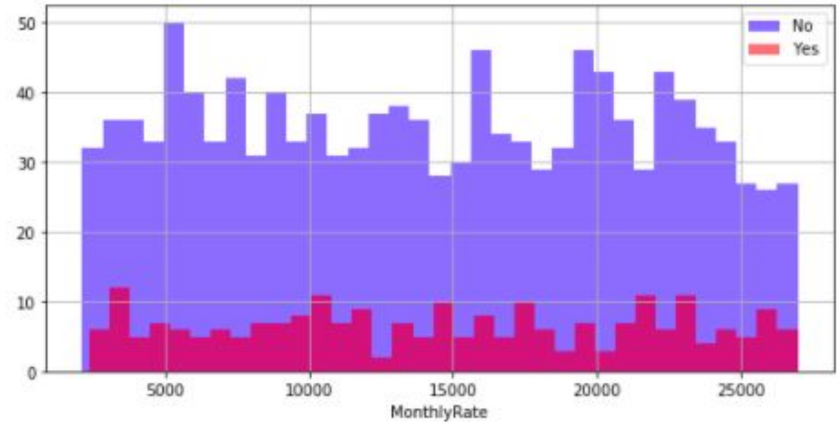
```
Attrition                No       Yes
BusinessTravel
Non-Travel         0.920000  0.080000
Travel_Frequently  0.750903  0.249097
Travel_Rarely      0.850431  0.149569
All                0.838776  0.161224
=============================================
Attrition                     No       Yes
Department
Human Resources         0.809524  0.190476
Research & Development  0.861602  0.138398
Sales                   0.793722  0.206278
All                     0.838776  0.161224
=============================================
Attrition               No       Yes
EducationField
Human Resources   0.740741  0.259259
Life Sciences     0.853135  0.146865
Marketing         0.779874  0.220126
Medical           0.864224  0.135776
Other             0.865854  0.134146
Technical Degree  0.757576  0.242424
All               0.838776  0.161224
```

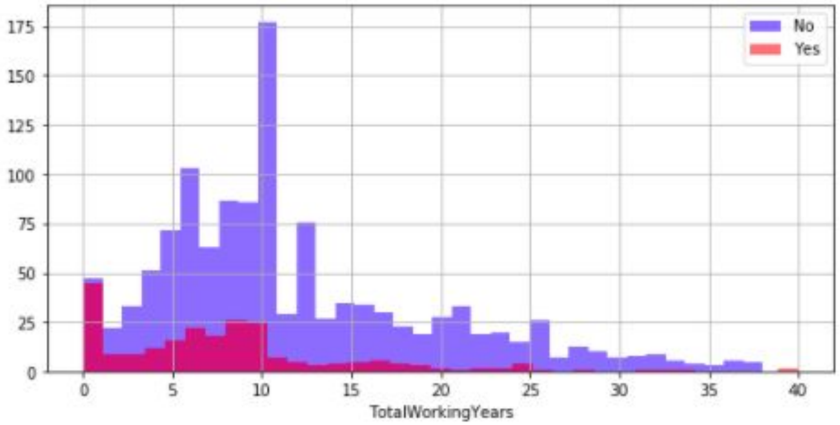


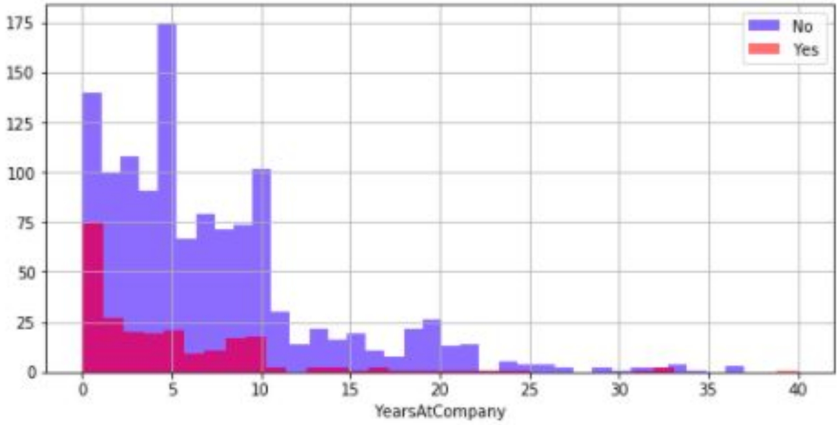
No
Yes
DistanceFromHome
Education
EnvironmentSatisfaction
JobInvolvement
JobLevel
JobSatisfaction
NumCompaniesWorked
PercentSalaryHike

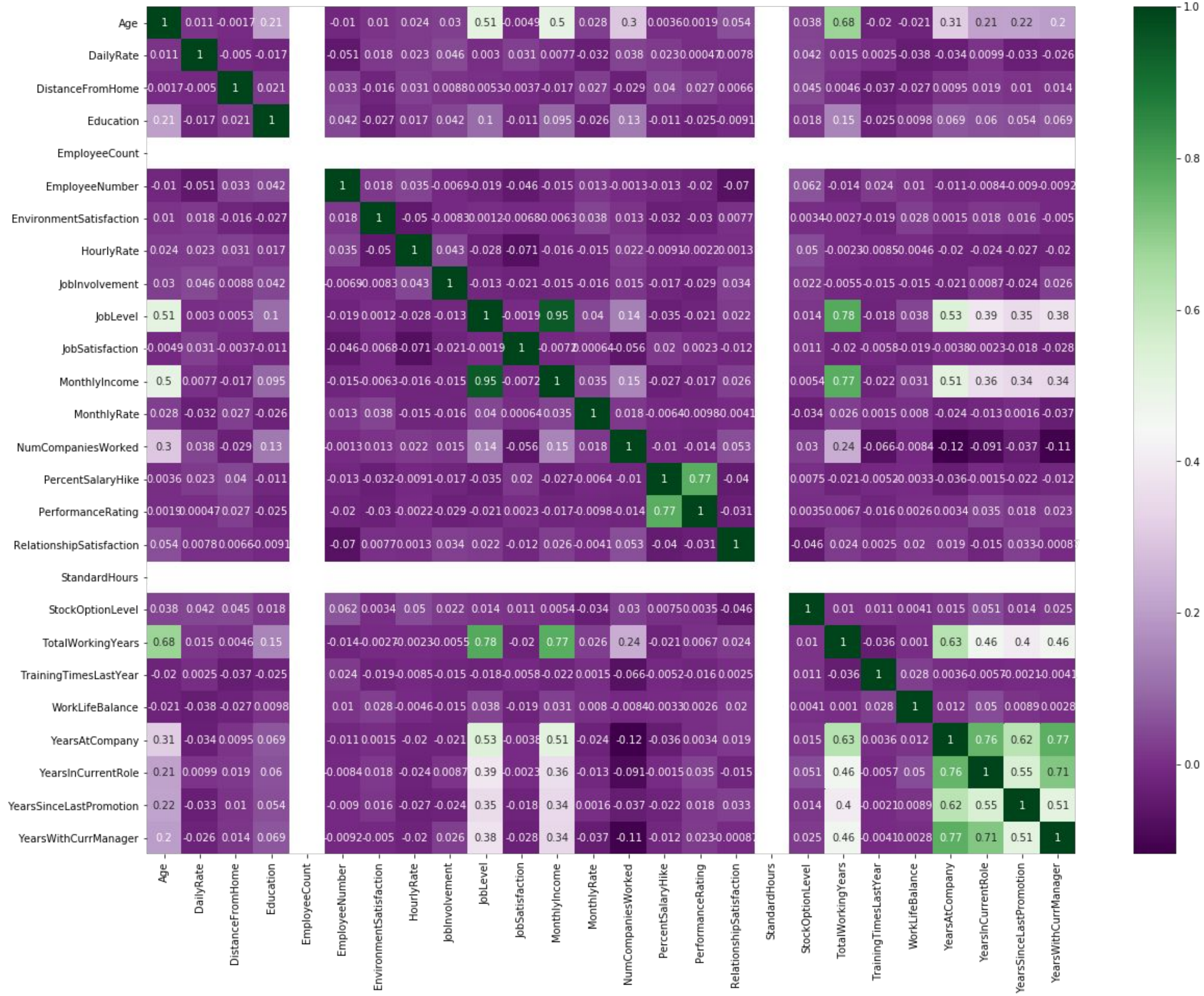

PerformanceRating
RelationshipSatisfaction
StockOptionLevel
TrainingTimesLastYear
WorkLifeBalance
YearsInCurrentRole
YearsSinceLastPromotion
YearsWithCurrManager
No
Yes


No
Yes
Age

No
Yes
DailyRate

No
Yes
HourlyRate

No
Yes
MonthlyIncome


No
Yes
MonthlyRate


No
Yes
TotalWorkingYears


No
Yes
YearsAtCompany


Age
DailyRate
DistanceFromHome
Education
EmployeeCount
EmployeeNumber
EnvironmentSatisfaction
HourlyRate
JobInvolvement
JobLevel
JobSatisfaction
MonthlyIncome
MonthlyRate
NumCompaniesWorked
PercentSalaryHike
PerformanceRating
RelationshipSatisfaction
StandardHours
StockOptionLevel
TotalWorkingYears
TrainingTimesLastYear
WorkLifeBalance
YearsAtCompany
YearsInCurrentRole
YearsSinceLastPromotion
YearsWithCurrManager
1.0
0.8
0.6
0.4
0.2
0.0

Imbalance

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

```
No      1233
Yes      237
Name: Attrition, dtype: int64
```

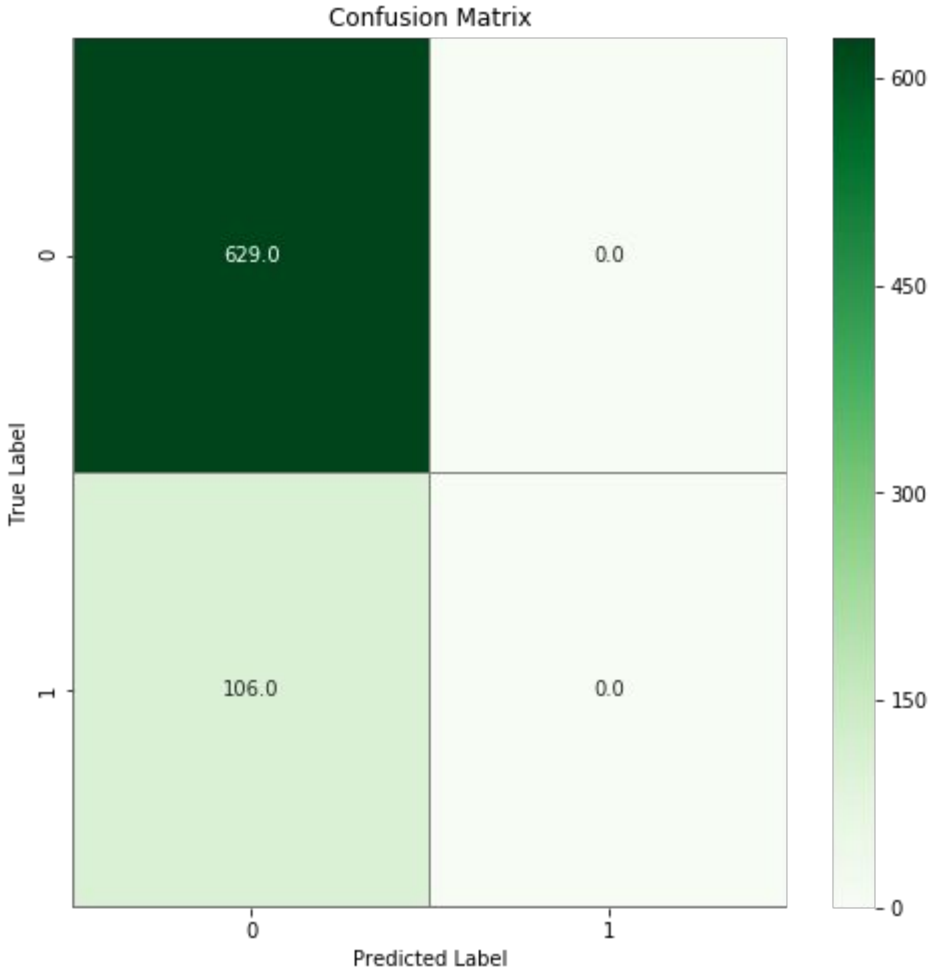
There is a massive imbalance in attrition. About 84% (1,233/1,470) of the employees didn't leave while the other 16% (237/1,470) did leave.

Neural Network

```
def build_model():
  model = keras.Sequential([
      layers.Dense(8, input_shape=(55,)),
      layers.BatchNormalization(),
      layers.Activation('relu'),
      layers.Dropout(0.5),
      layers.Dense(4, activation = 'relu'),
      layers.Dense(2, activation='softmax')
  ])

  model.compile(loss=tf.keras.losses.BinaryCrossentropy(from_logits=True),
                optimizer='adam',
                metrics=['accuracy'])
  return model
```

```
model.evaluate(test_data, test_labels, verbose = 2)
```

```
735/735 - 0s - loss: 0.5753 - accuracy: 0.8558
```

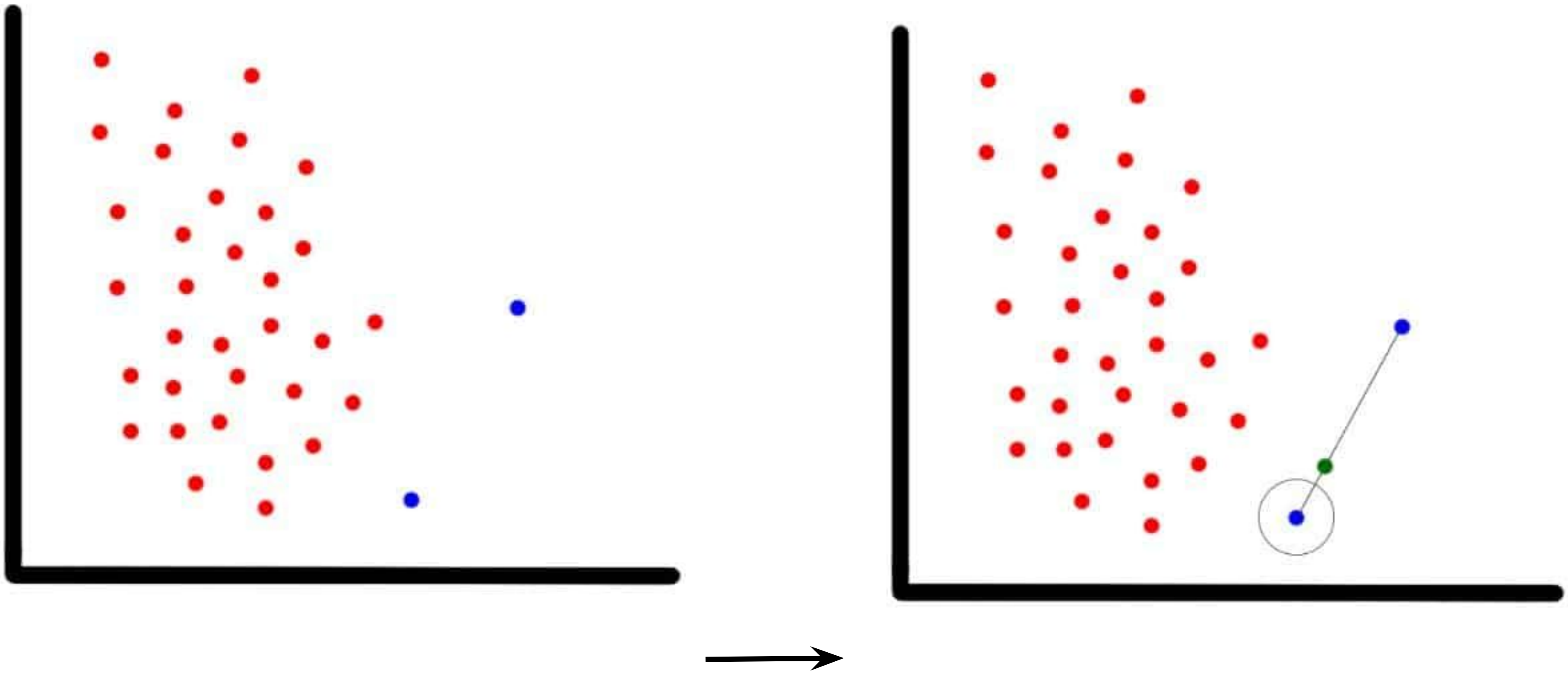


Confusion Matrix
629.0
0.0
106.0
0.0
True Label
Predicted Label
0
1
0
1
600
450
300
150
0

Oversampling

Heavily skewed data that favors one class makes is difficult for a model to effectively learn any patterns or learn the decision boundary. So a way to combat this is to oversample the minority class, in this case Yes for attrition. This is done by duplicating examples from the minority class in the training dataset prior to fitting the model, and it can balance the class distribution while not adding any additional, unnecessary information to the model.

One of the most common ways of synthesizing these new examples is through Synthetic Minority Oversampling Technique, or SMOTE. The way that SMOTE works is that it selects an example of the minority class at random. Then *k*, typically k = 5, of the nearest neighbors for that example are found. From those k-neighbors, a random neighbor is selected and a new example of the minority class is synthesized at a random location between the selected example and its nearest neighbor.



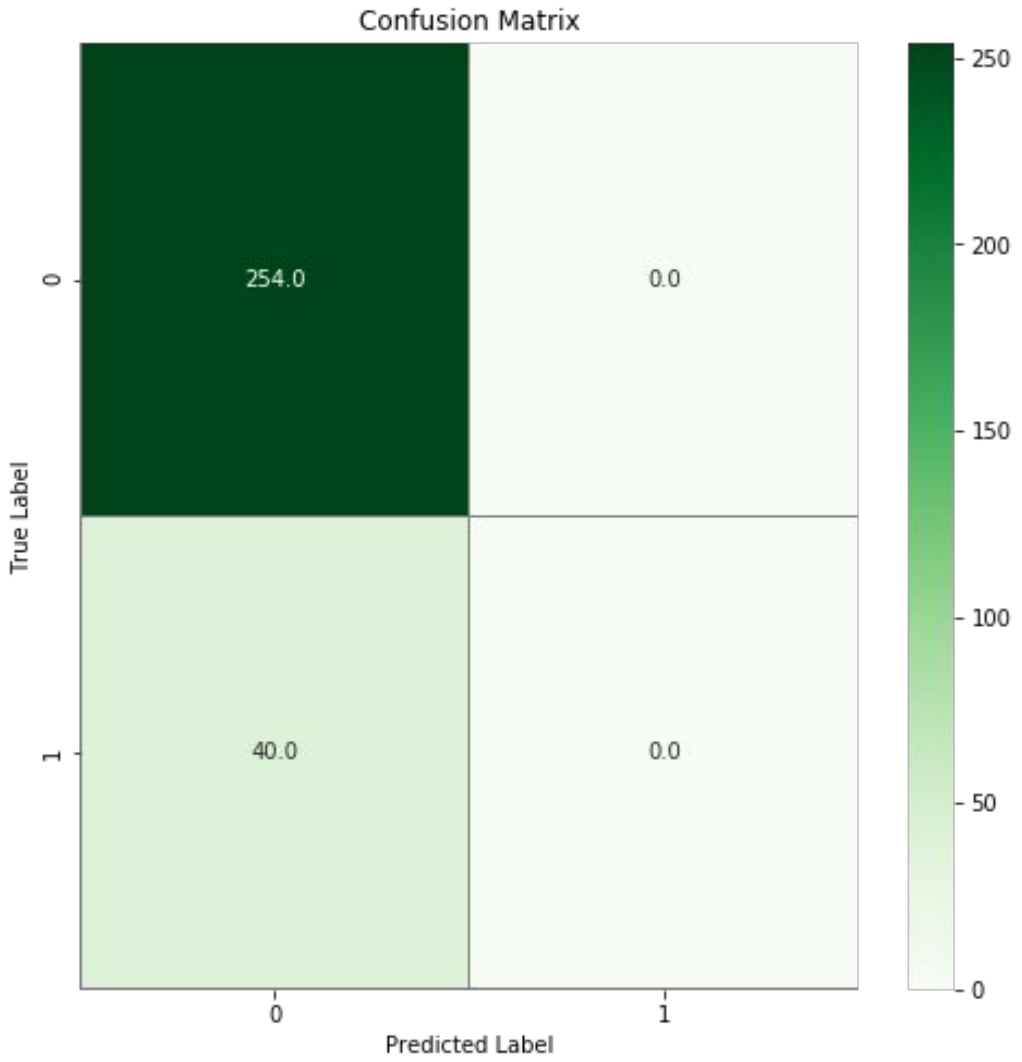
Pictures from:
https://kite.com/blog/python/smote-python-imbalanced-learn-for-oversampling/#smote-work

```
    model = keras.Sequential([
        layers.Dense(8, activation='relu', input_shape=(34,)),
        layers.Dropout(0.5),
        layers.BatchNormalization(),
        layers.Dense(4, activation = 'relu'),
        layers.Dense(1, activation='sigmoid')
    ])

    model.compile(loss=tf.keras.losses.BinaryCrossentropy(from_logits=True),
                  optimizer='adam',
                  metrics=['accuracy'])
    return model
```

```
model.evaluate(X_test, y_test, verbose = 2)
```

```
294/294 - 0s - loss: 0.7040 - accuracy: 0.8639
```

Problem: Just predicted everything to be a 0. So it doesn't seem that oversampling worked here.



Logistic Model

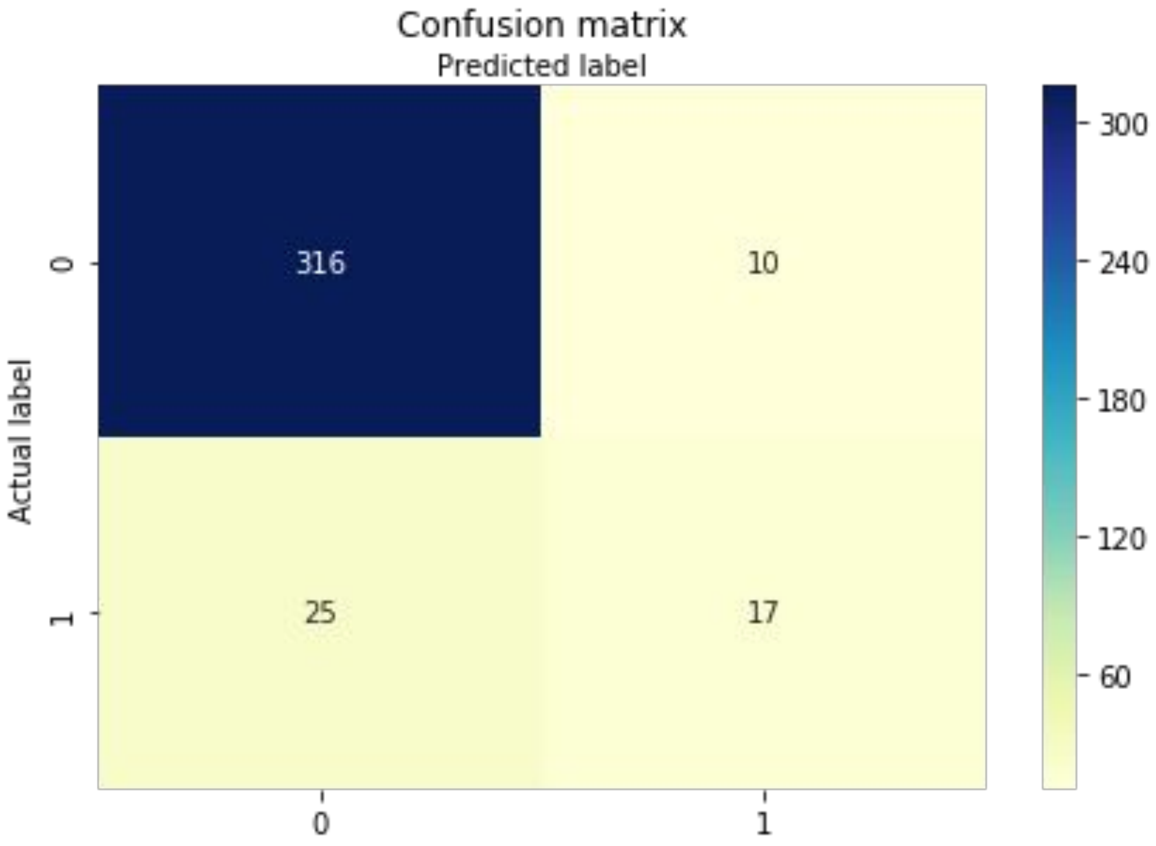
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state=26)
```

```
logreg = LogisticRegression(max_iter = 6000)

logreg.fit(X_train,y_train)

y_pred = logreg.predict(X_test)
```

```
# Training accuracy
logreg.score(X_train, y_train)
```

```
0.8629764065335753
```

```
Accuracy: 0.904891304347826
Precision: 0.6296296296296297
Recall: 0.40476190476190477
```


Confusion matrix
Predicted label
Actual label
316
10
25
17
0
1
0
1
300
240
180
120
60

Precision and Recall

No oversampling done here since it decreased the accuracy.

"Precision-Recall is a useful measure of success of prediction when the classes are very imbalanced. In information retrieval, precision is a measure of result relevancy, while recall is a measure of how many truly relevant results are returned.

High precision relates to a low false positive rate, and high recall relates to a low false negative rate. High scores for both show that the classifier is returning accurate results (high precision), as well as returning a majority of all positive results (high recall).

A system with high recall but low precision returns many results, but most of its predicted labels are incorrect when compared to the training labels.

A system with high precision but low recall is just the opposite, returning very few results, but most of its predicted labels are correct when compared to the training labels.

An ideal system with high precision and high recall will return many results, with all results labeled correctly."

XGBoost

We know that Boosting is an ensemble technique where new models are added to correct the errors made by existing models. Models are added sequentially until no further improvements can be made.

XGBoost stands for Extreme Gradient Boosting, and is an implementation of gradient boosting. Gradient boosting is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction, and is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.

Basic XGBoost:

```
Accuracy = 0.8913043478260869
Precision: 0.7083333333333334
Recall: 0.34
```

XGBoost finding best estimates with GridSearch:

```
Accuracy: 0.8913043478260869
Precision: 0.6785714285714286
Recall: 0.38
```

```
clf = xgb.XGBClassifier()
parameters = {
     "eta"    : [0.05, 0.10, 0.15, 0.20, 0.25, 0.40 ] ,
     "max_depth"        : [ 2, 4, 5, 6, 8, 10, 12, 15],
     "min_child_weight" : [ 1, 3, 5, 7 ],
     "gamma"            : [ 0.0, 0.1, 0.2 , 0.3, 0.4 ],
     "colsample_bytree" : [ 0.3, 0.4, 0.5 , 0.7 ] # perc
                                                   # to
     }
```

```
print(grid.best_params_)
```

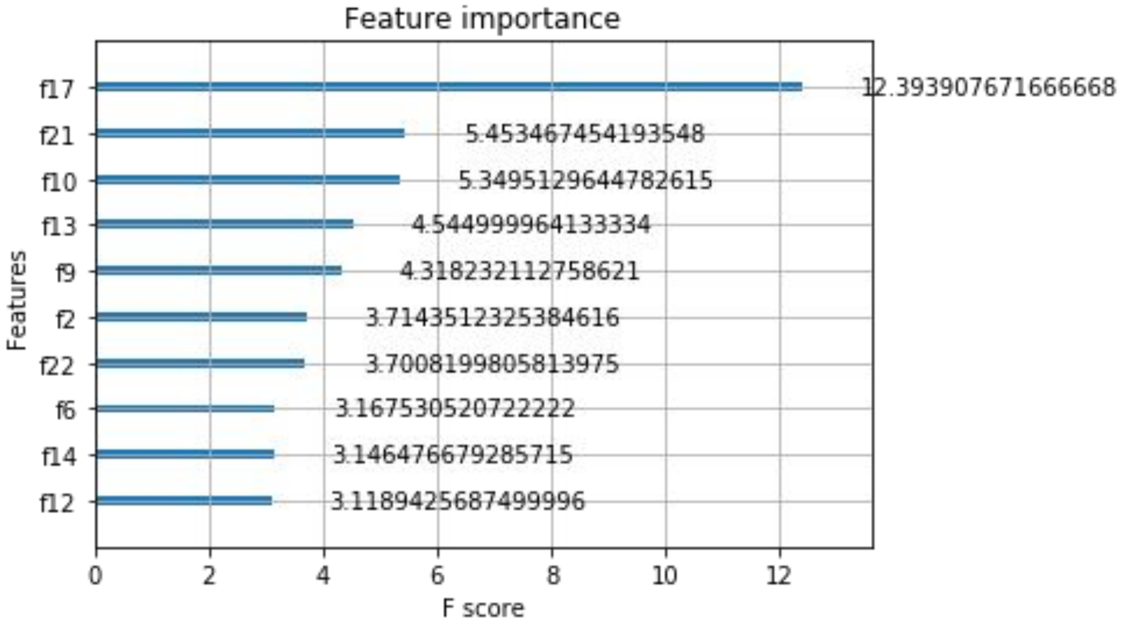
```
{'colsample_bytree': 0.3, 'eta': 0.05, 'gamma': 0.4, 'max_depth': 6, 'min_child_weight': 7}
```

Ran into overfitting when trying to find best estimators with grid search, which, I assume, is why the precision and recall are worse. Running the grid search took upwards of 30 minutes each time, which prevented me from playing with the parameters to fix the overfitting.

Feature Importance - XGBoost

XGBoost comes with a nice feature that allows us to determine the importance of features in the dataset. When computing the F-score, importance type is a parameter that needs set and I used “gain”. Gain is the average gain across all splits the feature is used in.

From researching, “The Gain is the most relevant attribute to interpret the relative importance of each feature. The Gain implies the relative contribution of the corresponding feature to the model calculated by taking each feature's contribution for each tree in the model. A higher value of this metric when compared to another feature implies it is more important for generating a prediction.” - https://datascience.stackexchange.com/questions/12318/how-to-interpret-the-output-of-xgboost-importance



f17: OverTime

f21: StockOptionLevel

f10: JobLevel

f13: MaritalStatus

f9: JobInvolvement

2: Department

f22: TotalWorkingYears

f6: EnvironmentSatisfaction

f14: MonthlyIncome

f12: JobSatisfaction

Other Models

- SVM
- Decision Tree
- Random Forest

SVM

Basic SVM Classifier

```
#Create a svm Classifier
clf = svm.SVC(kernel='linear') # Linear Kernel
```

```
Accuracy: 0.8885869565217391
Precision: 0.84
Recall: 0.3620689655172414
```

Tuning Parameters for SVM Classifier

```
{'C': 0.1, 'gamma': 1, 'kernel': 'rbf'}
```

```
Accuracy: 0.842391304347826
Precision: 0.0
Recall: 0.0
```

Note that the precision and recall is 0. This is because after finding the best estimators for SVM, the model predicted all 0s for the test data and there when trying to calculate the precision and recall score, python passes a warning and sets them to 0.

```
set(y_test) - set(predictions)

{1}
```

This is telling us the label ‘1’ doesn’t appear in predictions

Decision Tree

```
# Create decision tree classifier
clf = DecisionTreeClassifier(criterion = "entropy", max_depth = 2)
```

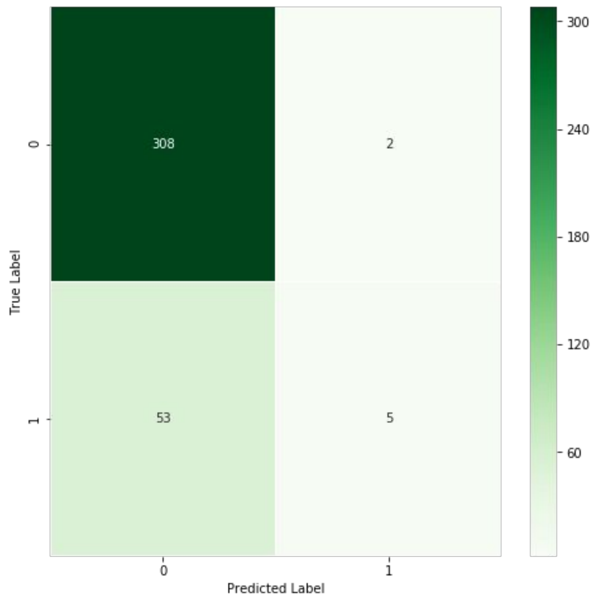
```
Accuracy: 0.8505434782608695
Precision: 0.5454545454545454
Recall: 0.3103448275862069
```

Random Forest

```
model = RandomForestClassifier(n_estimators = 10, criterion = "entropy", max_depth = 5, random_state = 0)
model.fit(X_train, y_train)
```

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='entropy', max_depth=5, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=10,
                       n_jobs=None, oob_score=False, random_state=0, verbose=0,
                       warm_start=False)
```

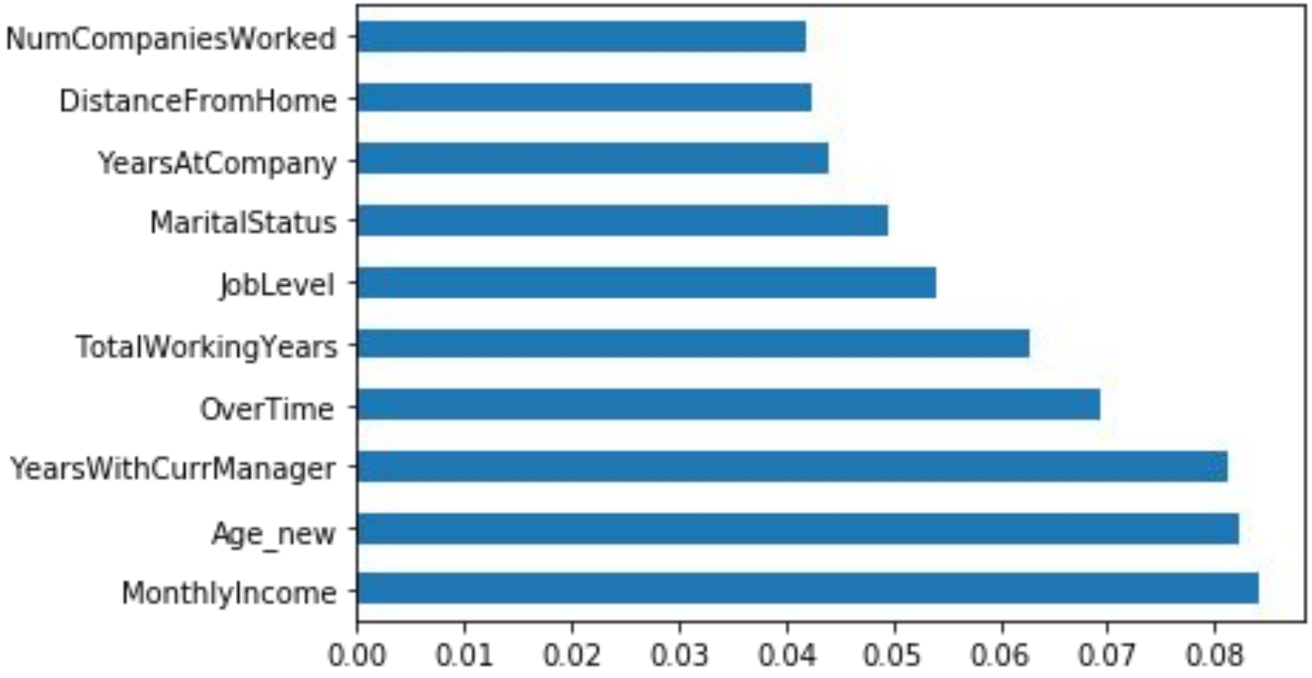
```
Model Testing Accuracy: 0.8505434782608695
Precision: 0.7142857142857143
Recall: 0.08620689655172414
```



Feature Importance - Random Forest

“The most common mechanism to compute feature importances is the mean decrease in impurity (or gini importance) mechanism. The mean decrease in impurity importance of a feature is computed by measuring how effective the feature is at reducing uncertainty (classifiers) or variance (regressors) when creating decision trees within RFs. The problem is that this mechanism, while fast, does not always give an accurate picture of importance” - https://explained.ai/rf-importance/#3

RF Feature Importance



RandomizedSearchCV vs. GridSearchCV

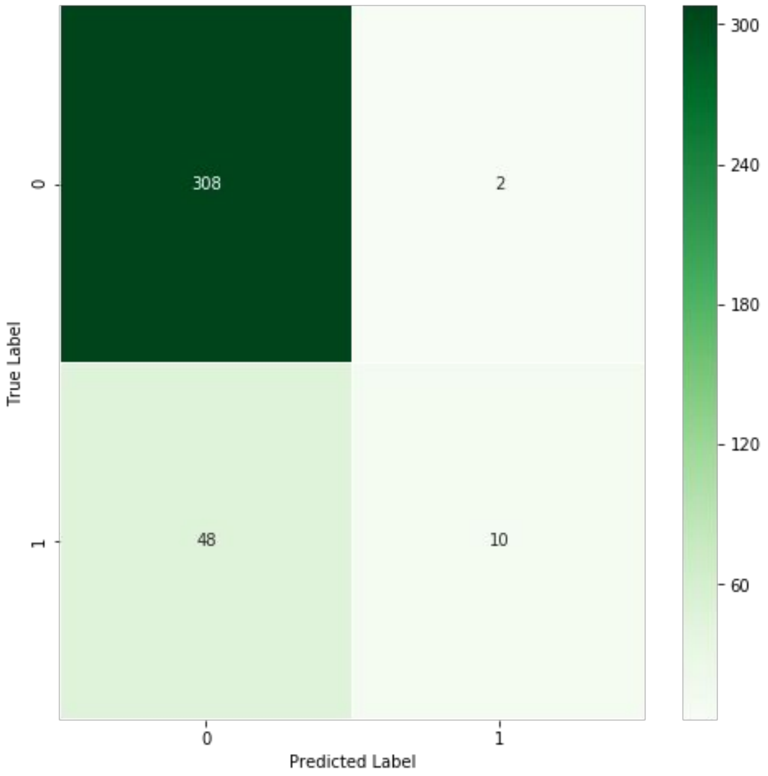
Grid search is an exhaustive search over specified parameter values for an estimator, while with randomized search not all parameter values are tried out, but rather a fixed number of parameter settings is sampled from the specified distributions.

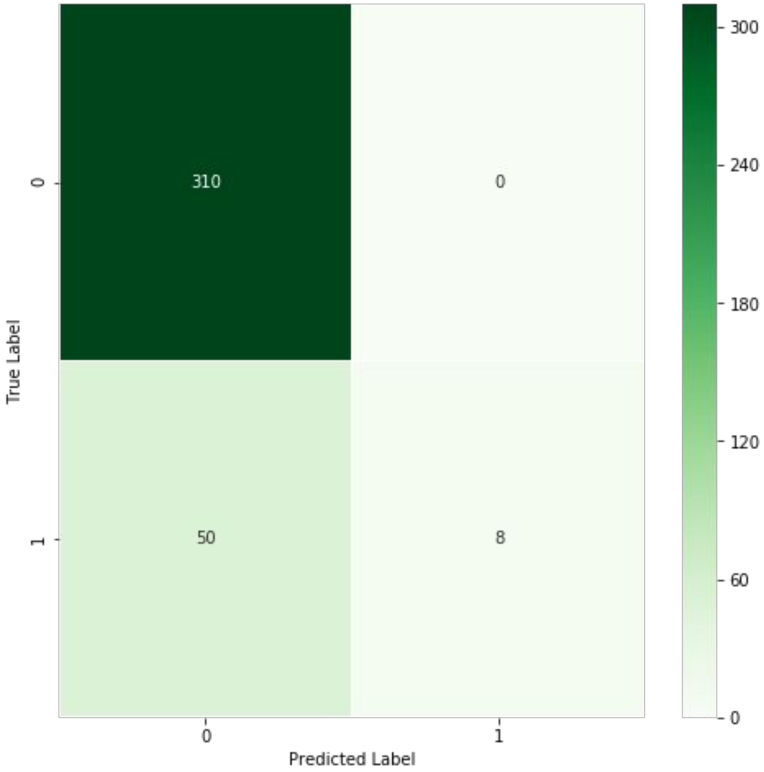
Note: I did run into overfitting with both, but wasn’t able to play around with the parameters to find better values due to each search taking upwards of 30 minutes each.

Accuracy: 0.8641304347826086
Precision: 0.8333333333333334
Recall: 0.1724137931034483

Randomized on left. Grid search on right

Accuracy: 0.8641304347826086
Precision: 1.0
Recall: 0.13793103448275862





Employee Attrition for Employees that Worked 1-2 years

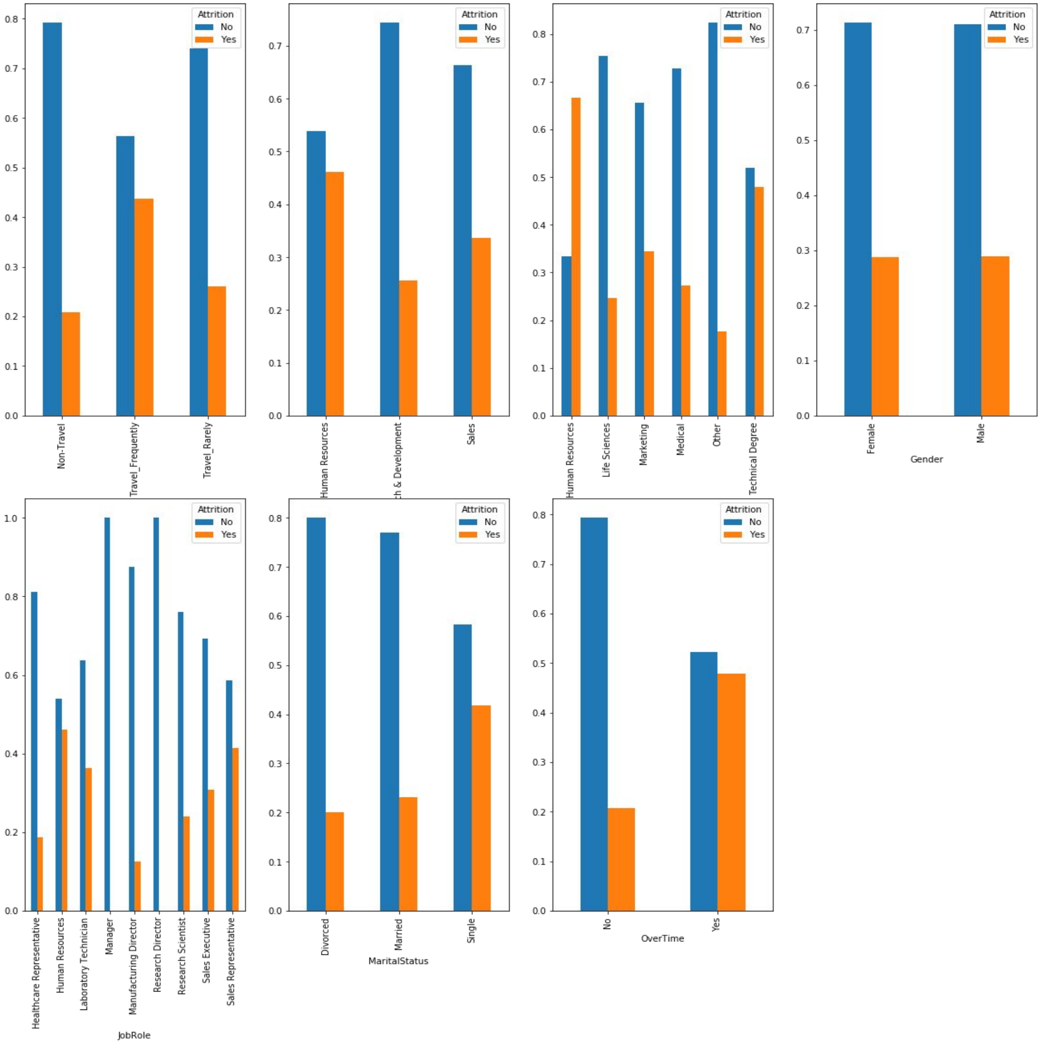
The previous models and EDA were for the entire dataset, but I wanted to know see how the models did on employees that were at the company for 1-2 years and see which features played a bigger role for those that did leave the company.

```
Attrition : ['No' 'Yes']
No      212
Yes      86
```

Still a very large imbalance in Yes and No


```
Age: Min: 19, Max: 60
=====================================================
DailyRate: Min: 104, Max: 1495
=====================================================
HourlyRate: Min: 30, Max: 100
=====================================================
MonthlyIncome: Min: 1009, Max: 19627
=====================================================
MonthlyRate: Min: 2097, Max: 26999
```

Y axis is in percent



Attrition	No	Yes
BusinessTravel		
Non-Travel	0.791667	0.208333
Travel_Frequently	0.563636	0.436364
Travel_Rarely	0.739726	0.260274
All	0.711409	0.288591

==

Attrition	No	Yes
Department		
Human Resources	0.538462	0.461538
Research & Development	0.743719	0.256281
Sales	0.662791	0.337209
All	0.711409	0.288591

==

Attrition	No	Yes
EducationField		
Human Resources	0.333333	0.666667
Life Sciences	0.753846	0.246154
Marketing	0.656250	0.343750
Medical	0.727273	0.272727
Other	0.823529	0.176471
Technical Degree	0.520000	0.480000
All	0.711409	0.288591

Attrition	No	Yes
Gender		
Female	0.712963	0.287037
Male	0.710526	0.289474
All	0.711409	0.288591

==

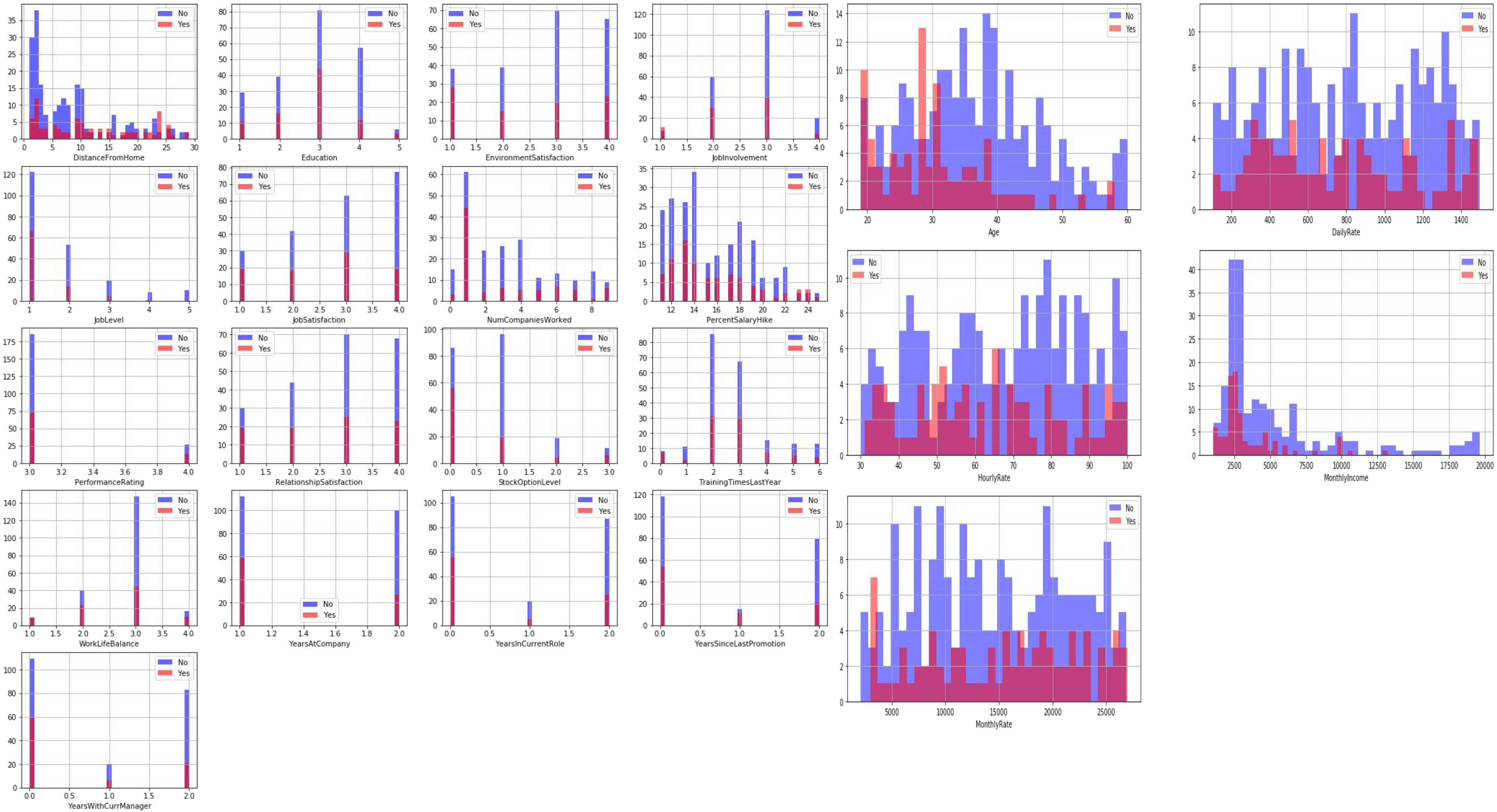
Attrition	No	Yes
JobRole		
Healthcare Representative	0.812500	0.187500
Human Resources	0.538462	0.461538
Laboratory Technician	0.636364	0.363636
Manager	1.000000	0.000000
Manufacturing Director	0.875000	0.125000
Research Director	1.000000	0.000000
Research Scientist	0.760000	0.240000
Sales Executive	0.692308	0.307692
Sales Representative	0.585366	0.414634
All	0.711409	0.288591

==

Attrition	No	Yes
MaritalStatus		
Divorced	0.800000	0.200000
Married	0.769231	0.230769
Single	0.582524	0.417476
All	0.711409	0.288591

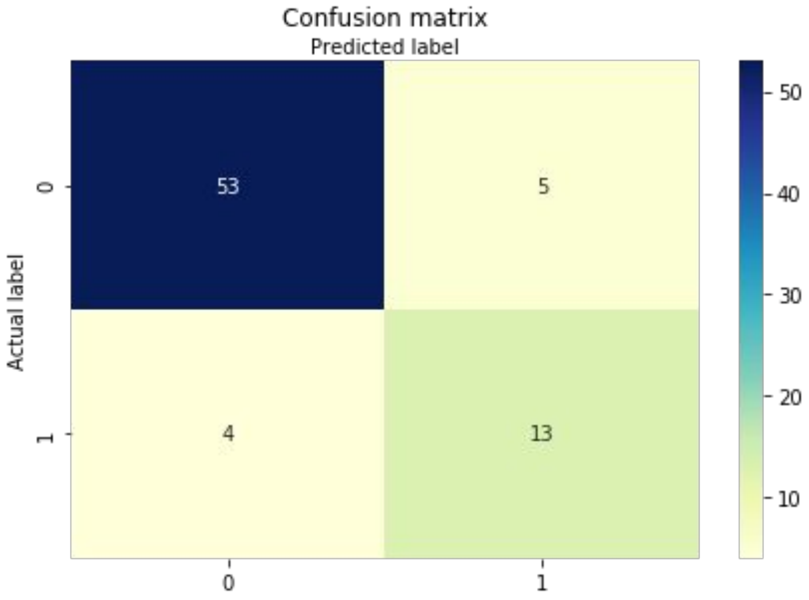
==

Attrition	No	Yes
OverTime		
No	0.793269	0.206731
Yes	0.522222	0.477778
All	0.711409	0.288591


No
Yes
DistanceFromHome
Education
EnvironmentSatisfaction
JobInvolvement
Age
DailyRate
JobLevel
JobSatisfaction
NumCompaniesWorked
PercentSalaryHike
HourlyRate
MonthlyIncome
PerformanceRating
RelationshipSatisfaction
StockOptionLevel
TrainingTimesLastYear
WorkLifeBalance
YearsAtCompany
YearsInCurrentRole
YearsSinceLastPromotion
MonthlyRate
YearsWithCurrManager

Logistic Regression

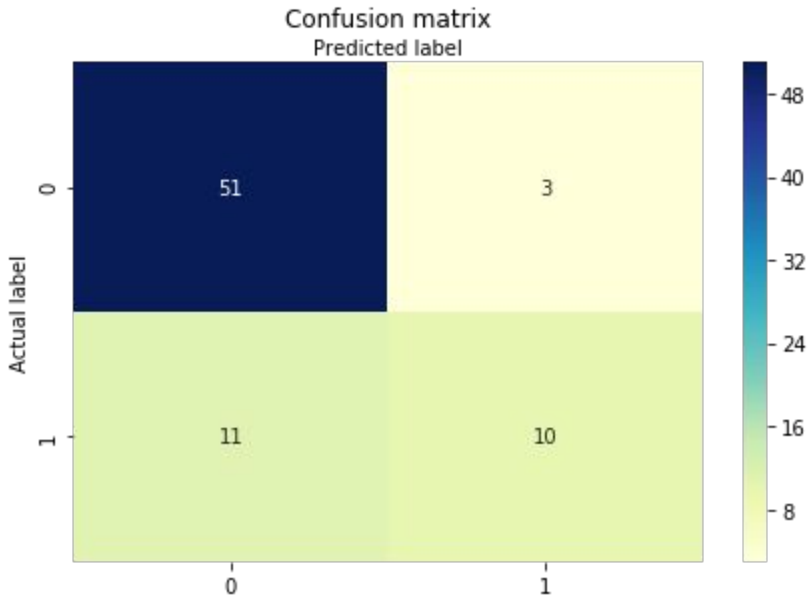
```
Accuracy: 0.88
Precision: 0.7222222222222222
Recall: 0.7647058823529411
```



XGBoost

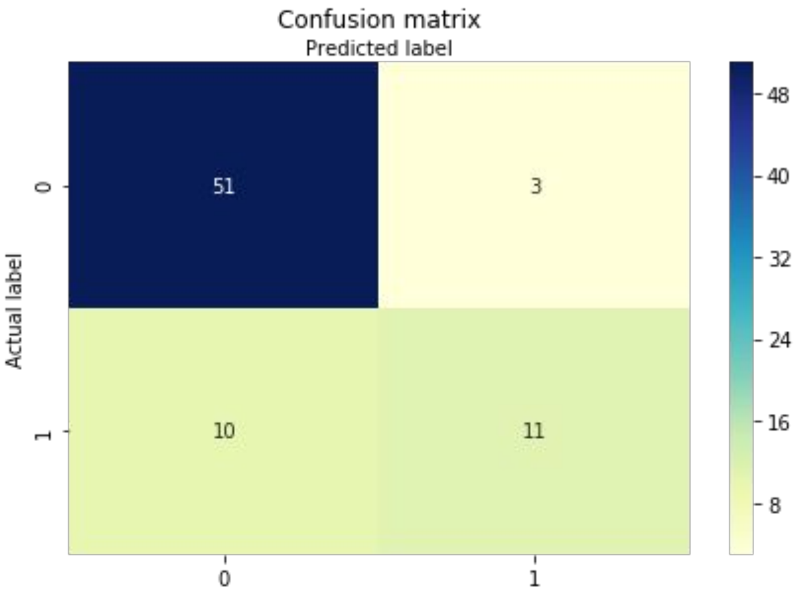
Normal XGBoost:

```
Accuracy: 0.8133333333333334
Precision: 0.7692307692307693
Recall: 0.47619047619047616
```

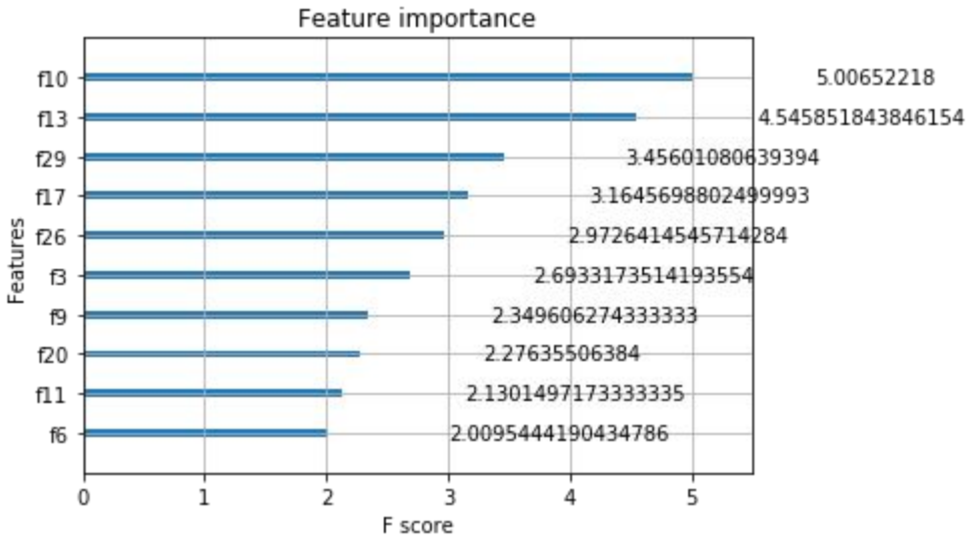


Tuned XGBoost. This is most likely worse due to overfitting.

```
Accuracy: 0.8266666666666667
Precision: 0.7857142857142857
Recall: 0.5238095238095238
```



Feature Importance



f10: JobLevel

f13: MaritalStatus

f29: Age_new

f17: OverTime

f26: YearsInCurrentRole

f3: DistanceFromHome

f9: JobInvolvement

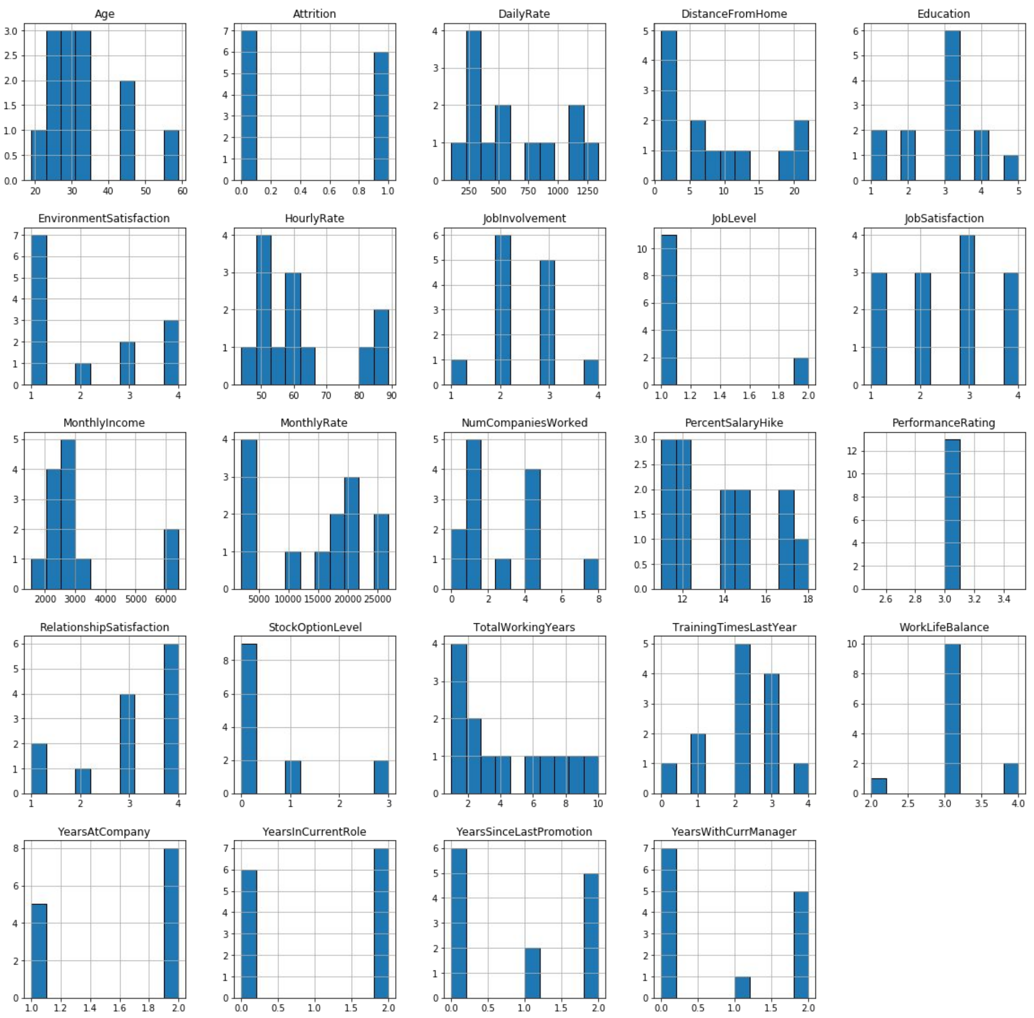
f20: RelationshipSatisfaction

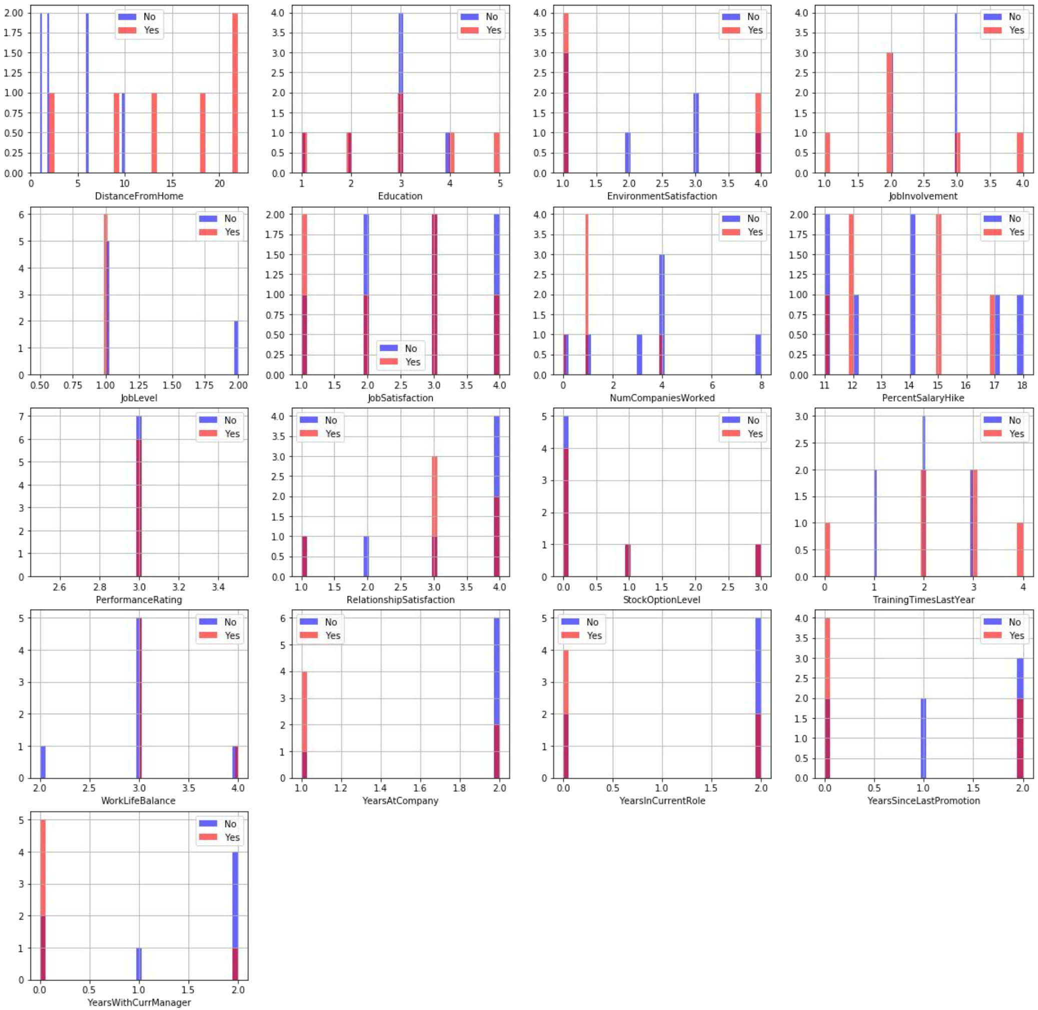
f11: PercentSalaryHike

f6: EnvironmentSatisfaction

Human Resources

1-2 Years (if time allows)


Age
Attrition
DailyRate
DistanceFromHome
Education
EnvironmentSatisfaction
HourlyRate
JobInvolvement
JobLevel
JobSatisfaction
MonthlyIncome
MonthlyRate
NumCompaniesWorked
PercentSalaryHike
PerformanceRating
RelationshipSatisfaction
StockOptionLevel
TotalWorkingYears
TrainingTimesLastYear
WorkLifeBalance
YearsAtCompany
YearsInCurrentRole
YearsSinceLastPromotion
YearsWithCurrManager


No
Yes
DistanceFromHome
Education
EnvironmentSatisfaction
JobInvolvement
JobLevel
JobSatisfaction
NumCompaniesWorked
PercentSalaryHike
PerformanceRating
RelationshipSatisfaction
StockOptionLevel
TrainingTimesLastYear
WorkLifeBalance
YearsAtCompany
YearsInCurrentRole
YearsSinceLastPromotion
YearsWithCurrManager