Data Mining for Business Intelligence Team Project

Using Tableau and Weka software

Data source: Kaggle

Introduction

- The goal of this project is to develop a classification model to determine whether or not an individual working in a certain Indian company is likely to leave the company within the next 2 years based on a set of attributes including but not limited to gender, age, and domain experience (https://www.kaggle.com/datasets/tejashvi14/employee-future-prediction?resource=download)
- Our dataset consists of 9 attributes and 4500+ rows, with each row containing information on a unique individual, including a class attribute stating whether or not they left their company after 2 years
- We will be visualizing the data in Tableau in order to better understand the impact of certain attributes on the outcome of whether or not an employee would stay with the company
- From a management perspective, HR can use this data to find out which groups of employees to hire for the best retention, and how to improve the company's culture and compensation packages to improve retention as well

Introduction

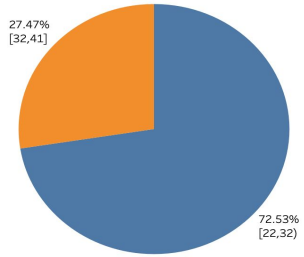
- Attributes:
  - Education: Highest level of education (i.e. Bachelors, Masters, PhD)
  - JoiningYear: The year the employee joined their company
  - City: City where the branch office the employee works in is located (within India)
  - PaymentTier: Ranked 1, 2, or 3 in order from highest to lowest level of income
  - Age: Current age
  - Gender: Gender of employee (i.e. Male of Female)
  - EverBenched: True if employee was every kept out of projects for 1+ month(s), else False
  - ExperienceInCurrentDomain: Years experience in current field
  - LeaveOrNot: Class attribute stating whether or not the employee leaves the company in the next 2 years (0=no, 1=yes)

Preprocessing

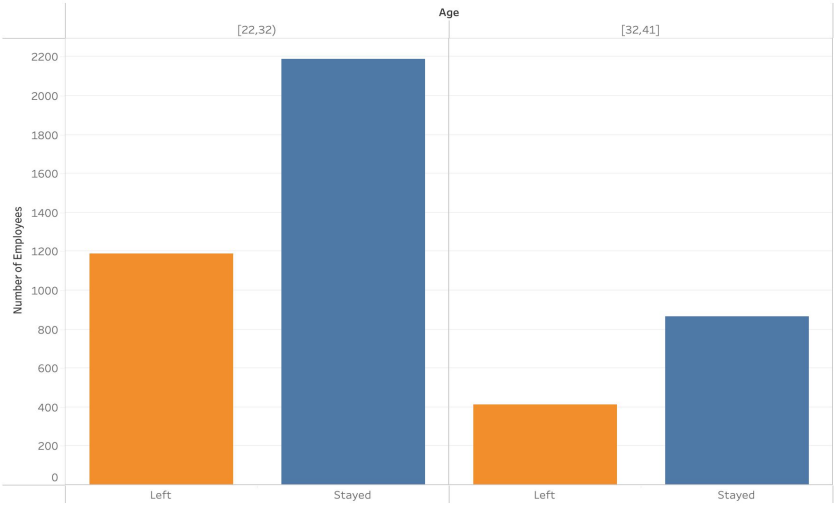
- Discretization
  - We used equal-interval (width) binning to discretize both the Age attribute and ExperienceInCurrentDomain attributes
  - The Age attribute was discretized into 2 bins, each roughly 10 years in width
    - Minimum age - 22, Maximum age - 41
    - Bins: [22, 32), [32, 41]
  - The ExperienceInCurrentDomain attribute was discretized into 3 bins, each roughly 2 years in width
    - Minimum experience - 0 years, Maximum experience - 7 years
    - Bins: [0, 2), [2, 4), [4, 7]
  - By discretizing we reduce model complexity due to the many unique, numeric values in each of these attributes
- Attribute conversion
  - Weka treats the JoiningYear attribute as a continuous attribute, we wish to use it as a nominal attribute, and thus converted it from continuous to nominal using the NumericToNominal filter in Weka and the appropriate attribute index

Tableau Visualization

Employee Breakdown by Age



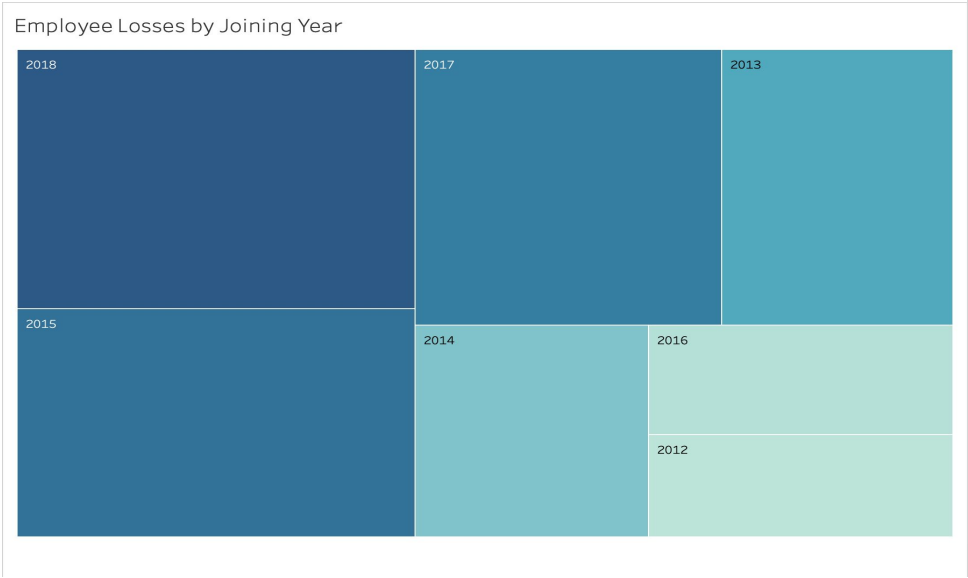
Employee Turnover by Age

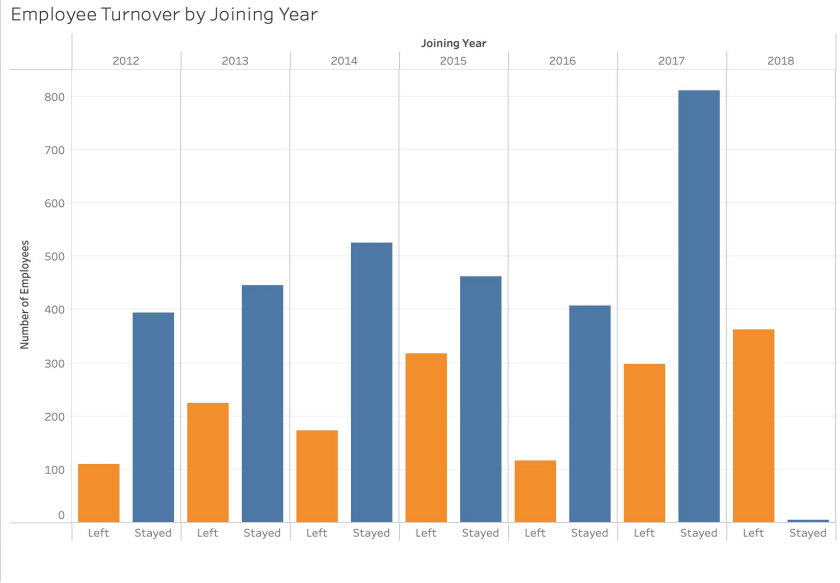


Key Insights:

Nearly 75% of employees fall in the 22-32 age group, and among these employees a larger proportion tend to leave the company (35.2%) compared with the older 32-41 age group (32.2%). This is to be expected as younger employees tend to jump from company to company or go back to school for their master's degrees. This data also shows the company may be relatively young, as they have a low proportion of older employees.

Tableau Visualization

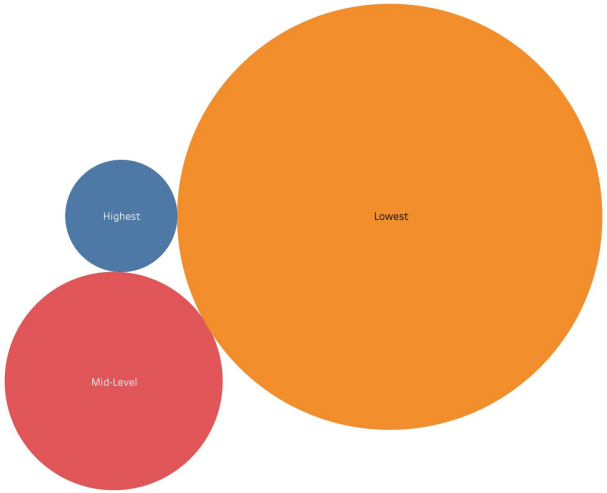




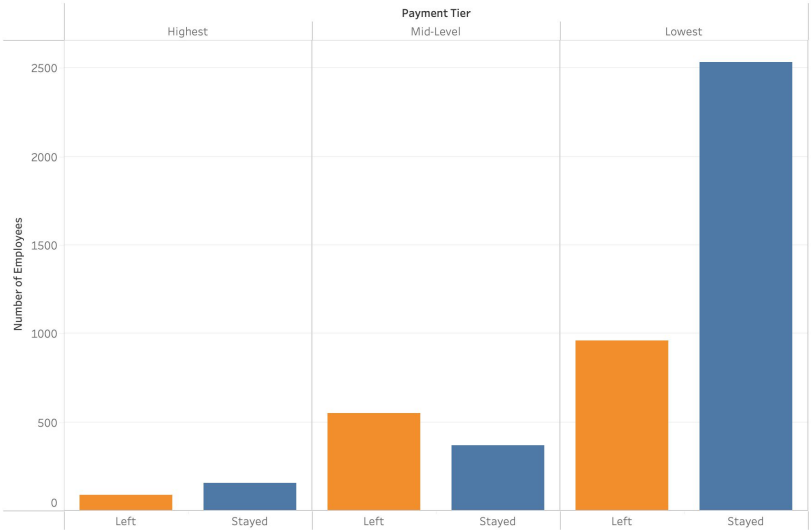
Key Insights:
Among employees who left the company after two years, most had joined the company in 2018. An analysis of new hires by year shows that of employees who joined the company in 2018, nearly all of them left after two years. This could have been due to economic factors, or significant dissatisfaction with the recent direction of the company.

Tableau Visualization

Employee Breakdown by Payment Tier



Employee Turnover by Payment Tier

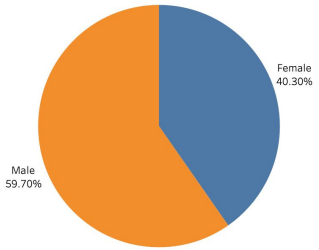


Key Insights:

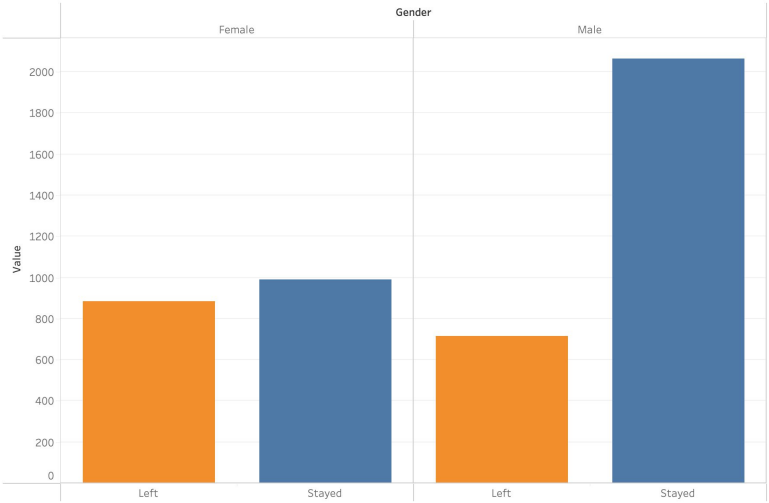
Most employees fall within the lowest payment tier. We would have expected the lowest payment tier to have the highest employee turnover ratio, but the mid-level tier actually had the highest proportion of employees leave after two years (59.9%). This could indicate that employees feel that they are not being adequately compensated for advancing in the company after being promoted.

Tableau Visualization

Breakdown of Employees by Gender



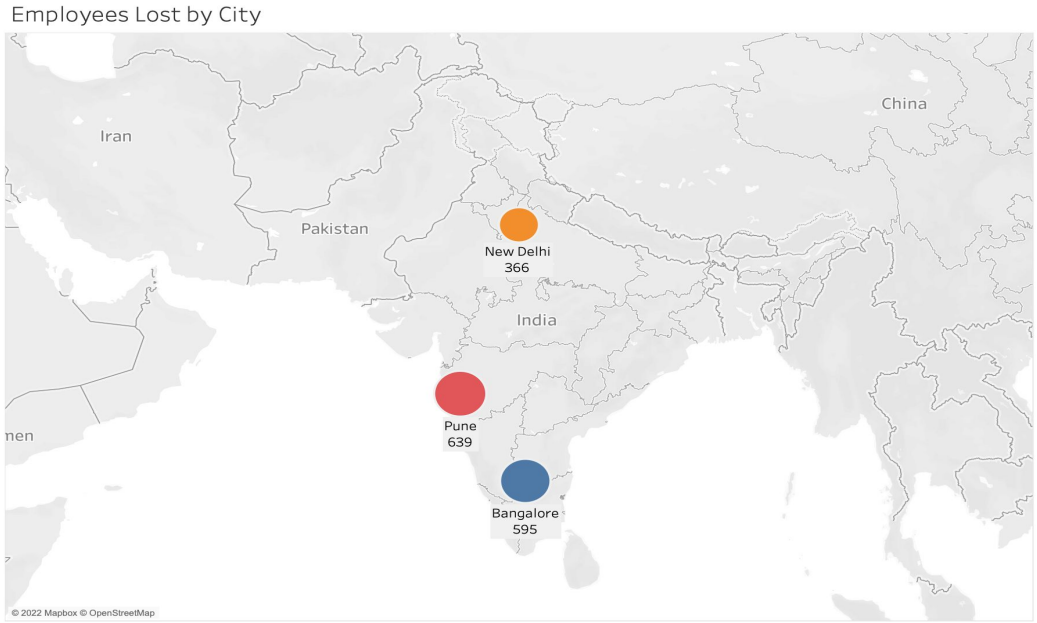
Employee Turnover by Gender



Key Insights:

The company is male dominated, with nearly 60% of employees being male. Additionally, a much higher proportion of females (47.1%) compared to males (25.8%) tend to leave the company after two years. This could indicate problems with workplace harassment and discriminatory practices such as pay inequality.

Tableau Visualization



Key Insights:
The Pune and Bangalore branches have lost significantly more employees over the last two years compared to New Delhi. This could indicate poor local management or subpar living conditions in the cities.

Tree Classifier - J48 Algorithm

- Indirect method of classification, meaning it extracts classification rules based on other classifiers which in this case is a decision tree
- Decision trees made on the principle of entropy reduction and information gain and is thus known as a statistical classifier
- **Advantages:**
  - Decision trees do not require normalization of data and preprocessing is simplified as decision trees do not require scaling of data
  - Decision trees are intuitive and the statistical basis for classification is easy to understand
- **Disadvantages:**
  - Decision trees are computationally costly at every node, meaning that this is especially true for datasets with many attributes.
  - The pruning process is also inefficient and expensive since it occurs after the decision tree is built and the data may be overfitted
- Overfitting occurs when decision trees are split to such a granular degree that every point is learned extremely well and data is model attempts to “perfectly” classify data

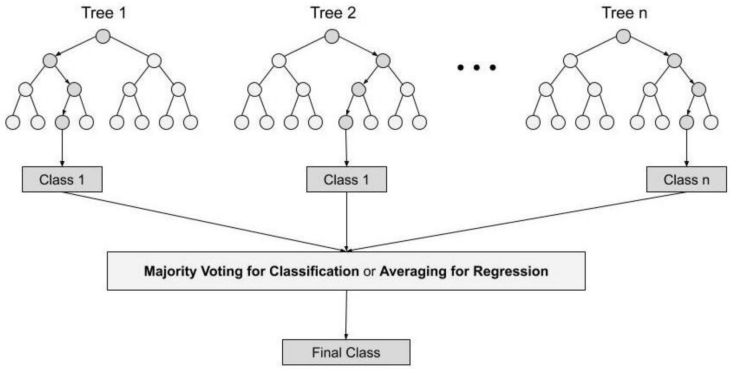
J48 Weka Output

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        3828               82.2695 %
Incorrectly Classified Instances       825               17.7305 %
Kappa statistic                          0.5702
Mean absolute error                      0.282
Root mean squared error                  0.378
Relative absolute error                 62.4873 %
Root relative squared error             79.5695 %
Total Number of Instances             4653

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.959    0.438    0.807      0.959    0.877      0.598    0.783     0.818     No
                 0.562    0.041    0.879      0.562    0.685      0.598    0.783     0.763     Yes
Weighted Avg.    0.823    0.301    0.832      0.823    0.811      0.598    0.783     0.799

=== Confusion Matrix ===

    a    b   <-- classified as
 2929  124 |    a = No
  701  899 |    b = Yes
```

The J48 decision tree algorithm gives us an accuracy rate of 82.3% for the 5 key attributes we identified in the Tableau visualization. It also provided an ROC area of 0.783 (closer to 1 the better)

Tree Classifier - Random Forest

- Random forest tree used to overcome overfitting as it builds multiple trees on subsets of the data
  - Builds decision trees on different samples and takes the majority vote for classification
- **Advantages:**
  - Each tree is created independently out of different data and attributes. This gives it diversity and utilizes all the data properly.
  - There is stability since it is based on majority voting and averaging
- **Disadvantages:**
  - Since each tree does not consider every single attribute, the random selection may be skewed. Not all features are considered while making a tree in random forest
  - Slower than other methods such as decision tree



Random Forest Weka Output

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        3830               82.3125 %
Incorrectly Classified Instances       823               17.6875 %
Kappa statistic                          0.5755
Mean absolute error                      0.2717
Root mean squared error                  0.3764
Relative absolute error                 60.2154 %
Root relative squared error             79.2331 %
Total Number of Instances             4653

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.950    0.418    0.813      0.950    0.876      0.597    0.804     0.847     No
                 0.582    0.050    0.858      0.582    0.693      0.597    0.804     0.778     Yes
Weighted Avg.    0.823    0.292    0.828      0.823    0.813      0.597    0.804     0.823

=== Confusion Matrix ===

    a    b   <-- classified as
 2899  154 |    a = No
  669  931 |    b = Yes
```

Random Forest gives us an accuracy rate of 82.3% and an ROC area of 0.804

Ensemble Classifier - AdaBoost

- AdaBoost is an ensemble learning method that also classifies under “meta-learning” which was created to increase the efficiency of binary classifiers. One of the benefits of AdaBoost is that it uses an iterative approach which allows the model to learn from the mistakes the weak classifiers, and turn it into strong classifiers.
- AdaBoost can be applied on any classifier and learn about the data to propose a more accurate model.
- AdaBoost is best used as a technique to build off other classification algorithms as opposed to being just a classifier it self.
- **Advantages**:
    - An advantage of AdaBoost is that it is less prone to overfitting the data as the input parameters that are used are not jointly optimized
    - AdaBoost also improves the accuracy of weak classifiers
        - Because of this AdaBoost has become more popular in being used to classify text and images rather than binary classification problems.
- **Disadvantages**:
    - The biggest disadvantage of using AdaBoost is that it requires a quality dataset. Datasets that include many outliers or other noisy data is an extreme detriment and would need to be cleaned out before using. (**In our dataset there are no outliers, thus it would be considered an optimal dataset use an AdaBoost algorithm**.)

AdaBoost Weka Output

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        3643               78.2936 %
Incorrectly Classified Instances      1010               21.7064 %
Kappa statistic                          0.4624
Mean absolute error                      0.3433
Root mean squared error                  0.4051
Relative absolute error                 76.0838 %
Root relative squared error             85.2763 %
Total Number of Instances             4653

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.951    0.537    0.772      0.951    0.852      0.499    0.770     0.808     No
                 0.463    0.049    0.831      0.463    0.595      0.499    0.770     0.724     Yes
Weighted Avg.    0.783    0.369    0.792      0.783    0.763      0.499    0.770     0.779

=== Confusion Matrix ===

    a    b   <-- classified as
 2902  151 |    a = No
  859  741 |    b = Yes
```

AdaBoost gives us an accuracy rate of 78.3% and an ROC area of 0.770

Bayesian Classifier - Naive Bayes

- The Naive Bayes classifier is a probability based machine learning model used for classification. The basis of this model is derived from the Bayes theorem
- The dataset is divided into two parts, the feature matrix and response vector. It is called Naive because the X's are independent of each other
- The feature matrix is what contains the attributes in the dataset
- The response vector contains the value of the class variable (prediction or output). In our dataset, the response vector is the value in LeaveOrNot that displays whether the employee chooses to leave the company based on the other attributes.
- Naive Bayes assumes that each feature makes an independent and equal contribution to the outcome.
    - We assume that no pair of features are dependent. For example, the tier of education has no effect on the joining year or the age of the employee has no effect on the city of employment. These features are considered to be independent of one another.
    - Each feature in this dataset must also have the same weight and impact as each other. For example, the Age of an employee must have the same importance as the Gender. All attributes must be contributing equally to the overall outcome.

Bayesian Classifier - Naive Bayes (continued)

Advantages:

- Algorithm works quickly and saves a lot of time
- Suitable for multi-class prediction problems
- If assumption of independence of features holds true, then it performs better than other models and requires much less training data
- Better suited for categorical input variables rather than numerical variables

Disadvantages:

- Assumes that all predictors are independent which rarely happens in real life, limiting the applicability of this algorithm
- Has a zero-frequency problem where zero probability is assigned to a categorical variable where the category from the test data set has no available data
    - Better to use a smoothing technique to overcome this issue
- The estimations of this model may be incorrect in some cases so the outputs are more for a generalization and should not be interpreted for accuracy

Naive Bayes Weka Output

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        3699               79.4971 %
Incorrectly Classified Instances       954               20.5029 %
Kappa statistic                          0.5175
Mean absolute error                      0.3081
Root mean squared error                  0.3943
Relative absolute error                 68.277  %
Root relative squared error             83.006  %
Total Number of Instances             4653

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.908    0.421    0.805      0.908    0.853      0.528    0.783     0.824     No
                 0.579    0.092    0.767      0.579    0.660      0.528    0.783     0.751     Yes
Weighted Avg.    0.795    0.308    0.792      0.795    0.787      0.528    0.783     0.799

=== Confusion Matrix ===

    a    b   <-- classified as
 2772  281 |    a = No
  673  927 |    b = Yes
```

Naive Bayes gives us an accuracy rate of 79.5% and an ROC area of 0.783

Instance Based Classifier - K-nearest Neighbor (IBk)

- K-nearest neighbor is an instance based classifier which uses training records to predict the class of unseen cases
- It works by using the class labels of the k-closest neighbors to a particular instance (measured using, for example, Euclidean distance) to classify the instance by taking the majority vote
- Requires three components - the set of training records, the value of k, the distance metric for computing distance between records
- **Advantages**
  - Simple to implement and intuitive to understand
  - Variety of distance measures to choose from
- **Disadvantages**
  - Lazy learner model, so classifying unknown records is relatively expensive
  - Sensitive to noise and outliers
  - Choosing the value of k can be difficult
    - If k is too small - classifier is sensitive to noise points
    - If k is too large - neighborhood of an instance may include points from many different classes

K-nearest Neighbor Weka Output

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        3826               82.2265 %
Incorrectly Classified Instances       827               17.7735 %
Kappa statistic                          0.5718
Mean absolute error                      0.2734
Root mean squared error                  0.3776
Relative absolute error                 60.5859 %
Root relative squared error             79.4986 %
Total Number of Instances             4653

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.953    0.427    0.810      0.953    0.876      0.595    0.802     0.845     No
                 0.573    0.047    0.864      0.573    0.689      0.595    0.802     0.777     Yes
Weighted Avg.    0.822    0.296    0.829      0.822    0.811      0.595    0.802     0.822

=== Confusion Matrix ===

    a    b   <-- classified as
 2909  144 |    a = No
  683  917 |    b = Yes
```

The IBk algorithm gives us an accuracy rate of 82.2% and an ROC area of 0.802

Model Summary

Model	Accuracy	TP Rate	FP Rate	Precision	Recall	ROC Area
J48	82.2695%	0.823	0.301	0.832	0.823	0.783
RandomForest	**82.3125%**	**0.823**	**0.292**	**0.828**	**0.823**	**0.804**
AdaBoost	78.2963%	0.783	0.369	0.792	0.783	0.770
Naive Bayes	79.4971%	0.795	0.308	0.792	0.795	0.783
K-NN (IBk)	82.2265%	0.822	0.296	0.829	0.822	0.802

Conclusions

- The Random Forest model performed the best out of all the classification models we selected for this dataset based on the measures on the previous slide, although all performed reasonably well
    - Different combinations of attributes should be used to see if accuracy can be further improved
- Based on the insights we collected from the Tableau visualization, an employee has a higher chance of leaving the company if they are younger, in the mid-level payment tier, female, and working out of the Pune or Bangalore offices
- The company should look into local management at the two branches with the highest turnover, evaluate its compensation packages when promoting employees to mid-level positions, and investigate the issues around female employees in the workplace
- Doing all of these things will allow them to better retain employees in the future