# Data Analysis on IBM HR Analytics

Jaylon Kiper, [jkiper@bellarmine.edu](mailto:jkiper@bellarmine.edu)

Sheridan Payne, [spayne@bellarmine.edu](mailto:spayne@bellarmine.edu)

**ABSTRACT**

To summarize our project, we worked on a dataset that covered the IBM company’s human resources analytics based on their employees. This seems to be a yearly coverage for HR to cover their employees and keep track of how they are doing within the workplace. The machine learning model used for this data analysis was the K-Nearest Neighbor model which classifies new instances based on a similarity measure. This dataset contains 1471 samples with 32 columns with various data types. Our three experiments were split into models based on training the data, validating it, and testing the final percentage. This includes splits of 80/10/10, 70/15/15, and 50/25/25. To conclude on what was found, using this model gave us messy results with the dataset used. This is most likely due to the dataset being fictional. However, out of the three models created using this dataset, we were able to see the relationship between the previously stated variables used for the models.

1. **INTRODUCTION**

The dataset we worked on covered the IBM company’s human resources analytics based on their employees. The data covers the employees salary, income, department,and other various variables. This seems to be a yearly coverage for HR to cover their employees and keep track of how they are doing within the workplace. The machine learning model used for this data analysis was the K-Nearest Neighbor model which classifies new instances based on a similarity measure.

1. **BACKGROUND**
   1. *Data Set Description*

The dataset we chose to work with was titled “IBM HR Analytics Employee Attrition & Performance,” retrieved from Kaggle. This dataset includes 35 distinct variables describing each surveyed employee’s age, hourly wage, department, education level, job role, and many more categories. We chose this dataset due to the amount of variables to experiment with and seemed fascinating since it was more business-related compared to both of our previous datasets, which were more entertainment-based. The purpose of the development of this fictional dataset was to uncover specific factors that lead to employee attrition and to explore questions such as “show me a breakdown of distance from home by job role and attrition.”

* 1. *Machine Learning Model*

We chose to use the K-nearest neighbor model, because our variables were straightforward and discrete values. Our second choice was the decision tree model based on how each variable is affected by each variable in the dataset. K-nearest neighbor shows the relationship between certain variables in a dataset, and since we have 32 variables to investigate after cleaning the data, we felt this machine learning model was the best for this dataset.

All of the calculations were calculated using the sklearn package in Python. With K-nearest neighbor, we split up the data using 5 variables (DailyRate, HourlyRate, PercentSalaryHike, YearsAtCompany, Attrition), where one of the variables were related to the other four variables, which is Attrition in our case. We had to then fit the variables to the model and predict future values of the Attrition variable using the four previously stated variables. Finally, we had to calculate the accuracy, precision, recall, and f1 score using the metrics package in sklearn.

1. **EXPLORATORY ANALYSIS**

This dataset contains 1471 samples with 32 columns with various data types. There are 32 total data types with this set and they most focus on strings and integers. Also included within this analysis is the summary statistics of the whole dataset, and other graphs to show correlations and counts based on the IBM employees information.

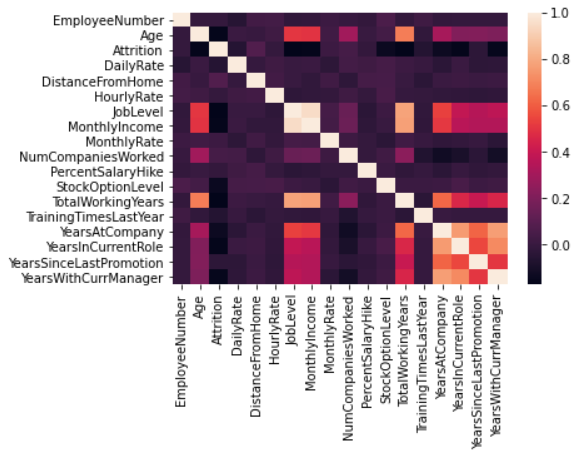
**Table 1: Data Types**

|  |  |
| --- | --- |
| *Variable Name* | *Data Type* |
| Employee Number | int64 |
| Age | int64 |
| Attrition | int64 |
| Business Travel | object |
| Daily Rate | int64 |
| Department | object |
| Distance From Home | int64 |
| Education | object |
| Education Field | object |
| Environment Satisfaction | object |
| Gender | object |
| Hourly Rate | int64 |
| Job Involvement | object |
| Job Level | int64 |
| Job Role | object |
| Job Satisfaction | object |
| Marital Status | object |
| Monthly Income | int64 |
| Monthly Rate | int64 |
| Num Companies Worked | int64 |
| Over Time | object |
| Percent Salary Hike | int64 |
| Performance Rating | object |
| Relationship Satisfaction | object |
| Stock Option Level | int64 |
| Total Working Years | int64 |
| Training Times Last Year | int64 |
| Work Life Balance | object |
| Years at Company | int64 |
| Years in Current Role | int64 |
| Years Since Last Promotion | int64 |
| Years with Curr Manager | int64 |

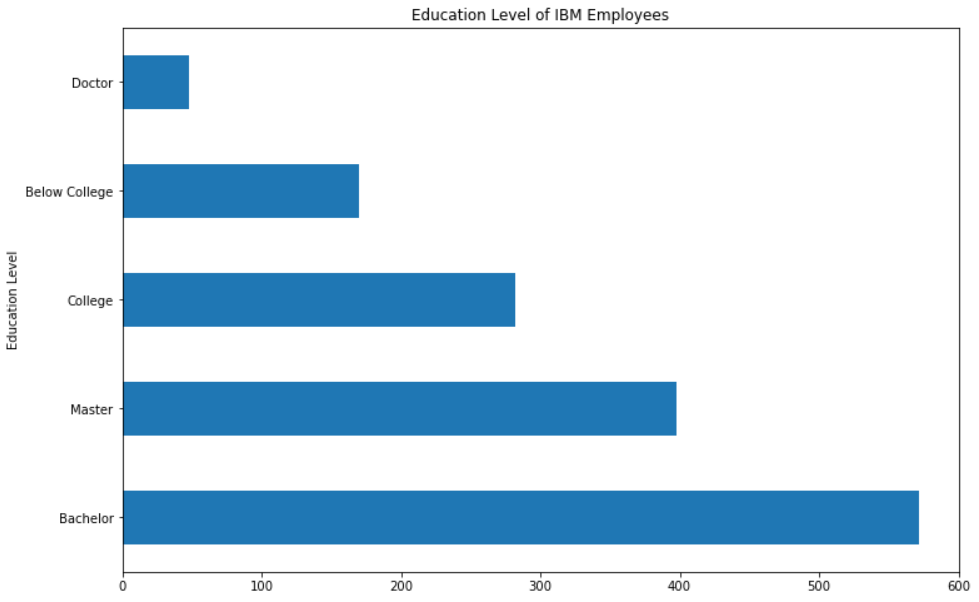
**Table 2: Summary Table**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **count** | **mean** | **std** | **min** | **25%** | **50%** | **75%** | **max** |
| **EmployeeNumber** | 1470.0 | 1024.865306 | 602.024335 | 1.0 | 491.25 | 1020.5 | 1555.75 | 2068.0 |
| **Age** | 1470.0 | 36.923810 | 9.135373 | 18.0 | 30.00 | 36.0 | 43.00 | 60.0 |
| **Attrition** | 1470.0 | 0.161224 | 0.367863 | 0.0 | 0.00 | 0.0 | 0.00 | 1.0 |
| **DailyRate** | 1470.0 | 802.485714 | 403.509100 | 102.0 | 465.00 | 802.0 | 1157.00 | 1499.0 |
| **DistanceFromHome** | 1470.0 | 9.192517 | 8.106864 | 1.0 | 2.00 | 7.0 | 14.00 | 29.0 |
| **HourlyRate** | 1470.0 | 65.891156 | 20.329428 | 30.0 | 48.00 | 66.0 | 83.75 | 100.0 |
| **JobLevel** | 1470.0 | 2.063946 | 1.106940 | 1.0 | 1.00 | 2.0 | 3.00 | 5.0 |
| **MonthlyIncome** | 1470.0 | 6502.931293 | 4707.956783 | 1009.0 | 2911.00 | 4919.0 | 8379.00 | 19999.0 |
| **MonthlyRate** | 1470.0 | 14313.103401 | 7117.786044 | 2094.0 | 8047.00 | 14235.5 | 20461.50 | 26999.0 |
| **NumCompaniesWorked** | 1470.0 | 2.693197 | 2.498009 | 0.0 | 1.00 | 2.0 | 4.00 | 9.0 |
| **PercentSalaryHike** | 1470.0 | 15.209524 | 3.659938 | 11.0 | 12.00 | 14.0 | 18.00 | 25.0 |
| **StockOptionLevel** | 1470.0 | 0.793878 | 0.852077 | 0.0 | 0.00 | 1.0 | 1.00 | 3.0 |
| **TotalWorkingYears** | 1470.0 | 11.279592 | 7.780782 | 0.0 | 6.00 | 10.0 | 15.00 | 40.0 |
| **TrainingTimesLastYear** | 1470.0 | 2.799320 | 1.289271 | 0.0 | 2.00 | 3.0 | 3.00 | 6.0 |
| **YearsAtCompany** | 1470.0 | 7.008163 | 6.126525 | 0.0 | 3.00 | 5.0 | 9.00 | 40.0 |
| **YearsInCurrentRole** | 1470.0 | 4.229252 | 3.623137 | 0.0 | 2.00 | 3.0 | 7.00 | 18.0 |
| **YearsSinceLastPromotion** | 1470.0 | 2.187755 | 3.222430 | 0.0 | 0.00 | 1.0 | 3.00 | 15.0 |
| **YearsWithCurrManager** | 1470.0 | 4.123129 | 3.568136 | 0.0 | 2.00 | 3.0 | 7.00 | 17.0 |

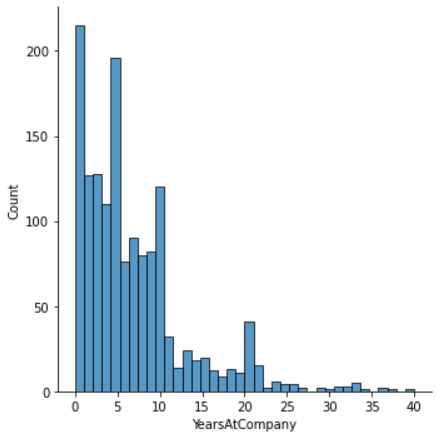
**Figure 1: Correlation Heatmap**



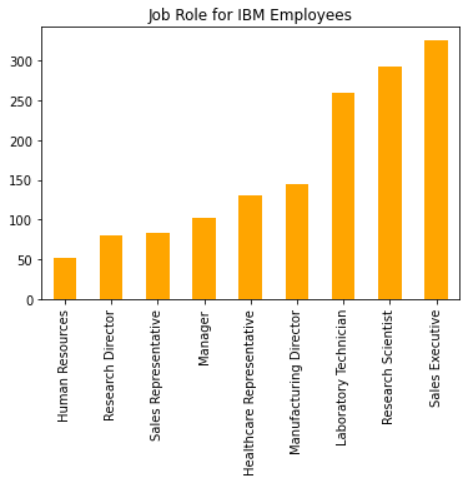
**Figure 2: Count of Each Education Level of IBM Employees**



**Figure 3: Count of IBM Employees by YearsAtCompany Variable**



**Figure 4: Count of Each Job Role of IBM Employees**



1. **METHODS**
   1. *Data Preparation*

Before importing our dataset into Python for analysis, we removed three columns: EmployeeCount, Over18, StandardHours. We removed the EmployeeCount columns since every value in the column was 1. Similarly for the Over18 columns, each value was ‘Yes’. We removed the StandardHours column for a similar reason as the previously two removed columns; every value in the column was 80. The following columns were initially ratings in the dataset that we later replaced with the string that represents each rating value in the respective column: Education, EnvironmentSatisfaction, JobInvolvement, JobSatisfaction, PerformanceRating, RelationshipSatisfaction, and WorkLifeBalance. We also moved the EmployeeNumber column to be the first column in the dataset. We did not normalize the data, since it was already normalized due to the dataset being a fictional one.

* 1. *Experimental Design*

**Table 3: Experiment Parameters**

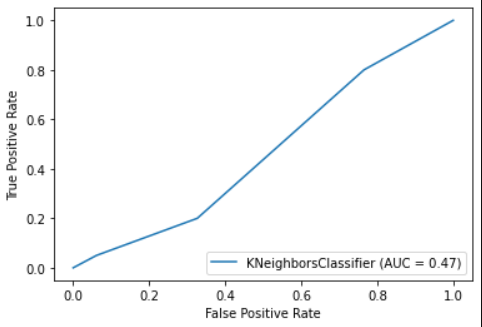
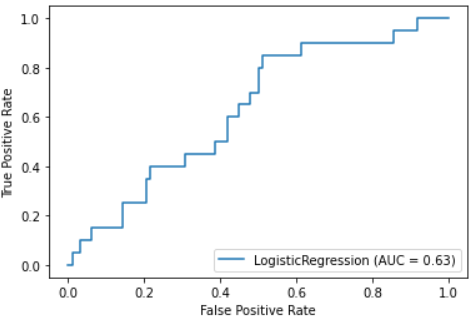
|  |  |
| --- | --- |
| **Experiment Number** | **Parameters** |
| 1 | All four (4) raw / normalized features with 80/10/10 split for train, validate, and test |
| 2 | All four (4) raw / normalized features with 70/15/15 split for train, validate, and test |
| 3 | All four (4) raw / normalized features with 50/25/25 split for train, validate, and test |

* 1. *Tools Used*

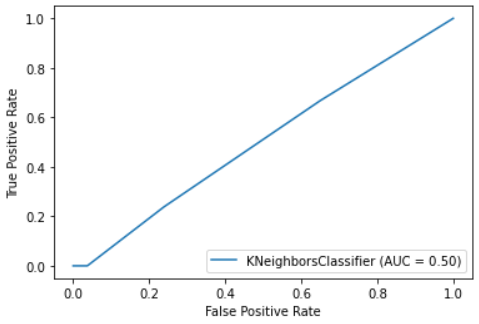
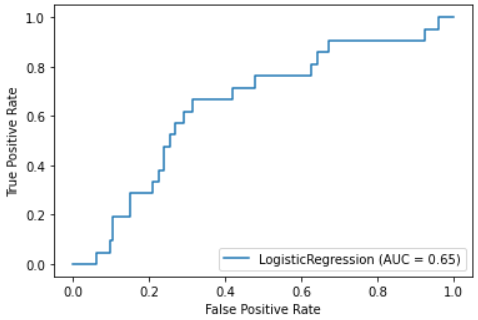
The following tools were used for this analysis: Python v3.5.2 running the Anaconda 4.3.22 environment for Microsoft Windows 10 computer was used for all analysis and implementation. In addition to base Python, the following libraries were also used: Pandas 0.18.1, Numpy 1.11.3, Matplotlib 1.5.3, Seaborn 0.7.1, and SKLearn 0.18.1. Pandas is used for dataframe formations and analysis; Numpy creates arrays to store data; Matplotlib is used to create graphs using data; Seaborn creates more complicated graphs Matplotlib cannot and creates cleaner graphs; SKLearn is necessary for the K-nearest neighbor machine learning model analysis on a dataset.

1. **RESULTS**
   1. *Classification Measures*

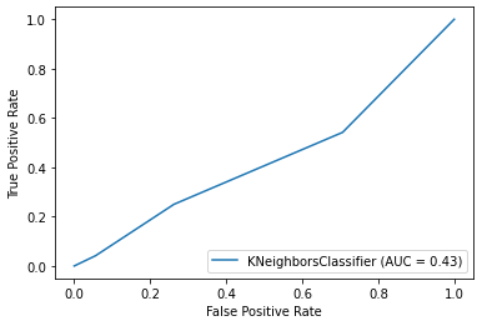
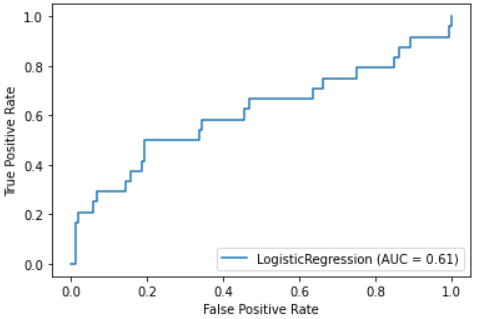
For the first experiment, we measured the selected data by taking 80% as the trained values, and 10% as the validated and test values. Our y prediction model with metrics rounded up had the accuracy at 0.79, precision at 0.14, recall at 0.05, and the F1 at 0.74. By using Logistic Regression, the y prediction model with metrics had the accuracy at 0.83, precision at 0.09375, recall at 0.019 , and the F1 at 0.031. Finally, the cross value scores overall ranged around 0.84 and 0.85.

For the second experiment, we measured the selected data by taking 70% as the trained values, and 15% as the validated and test values. Our y prediction model with metrics rounded up had the accuracy at 0.83, precision at 0.0, recall at 0.0, and the F1 at 0.0. By using Logistic Regression, the y prediction model with metrics had the accuracy at 0.82, precision at 0.19, recall at 0.029 , and the F1 at 0.050. Finally, the cross value scores overall ranged around 0.83 and 0.84.

For the third experiment, we measured the selected data by taking 50% as the trained values, and 25% as the validated and test values. Our y prediction model with metrics rounded up had the accuracy at 0.82, precision at 0.1, recall at 0.042, and the F1 at 0.059. By using Logistic Regression, the y prediction model with metrics had the accuracy at 0.84, precision at 0.1875, recall at 0.0375 , and the F1 at 0.06250. Finally, the cross value scores overall ranged around 0.85.

* 1. *Discussion of Results*

The model which provided the best classification was our 50/25/25 model. This model had the highest accuracy scores for the both testing and validation. Also, the cross value scores matched around 0.85 with the other two previous models had two separate scores. The worst model was our 70/15/15 model. The odd situation with this one includes how the precision, recall and F1 scores all placed at 0.0 with a negative rate on the K-Nearest model. Again, these three scores in the Logistic Regression were below 0.10. This model also had the lowest cross values between the three other experiments.

* 1. *Problems Encountered*

Since we needed to find a dataset with many different variables, with a healthy mix of categorical and continuous variables, finding a dataset took some time. Another problem we encountered was converting the columns involving an employee’s pay from string to integers. We had trouble converting these columns in Python, so we had to fix this issue in the original excel file before importing it back into Python.

* 1. *Limitations of Implementation*

When it comes to limitations, our ROC curve based on both K-Nearest and Logistic Regression created messy line graphs and curves. The K-Nearest graphs had linear data with bends at the beginning of the line and in the middle as well. The Logistic graphs were built like stairs and they had a higher AUC than the K-Nearest graphs. The model we chose gave us a lot of weird results, so I believe the best suited model could be the Decision Tree model.

* 1. *Improvements/Future Work*

We would probably use a different model to see what results we get and compare it to the K-Nearest Model.

1. **CONCLUSION**

The dataset we worked on was based on the IBM company’s human resources analytics based on their employees. The machine learning model used for this data analysis was the K-Nearest Neighbor model which classifies new instances based on a similarity measure. Using this model, we received messy results with the dataset used. This is most likely due to the dataset being fictional. However, out of the three models created using this dataset, we were able to see the relationship between the previously stated variables used for the models.

**REFERENCES**

pavansubhash. (2017). IBM HR Analytics Employee Attrition & Performance. [Data Set]. Retrieved from <https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset>.