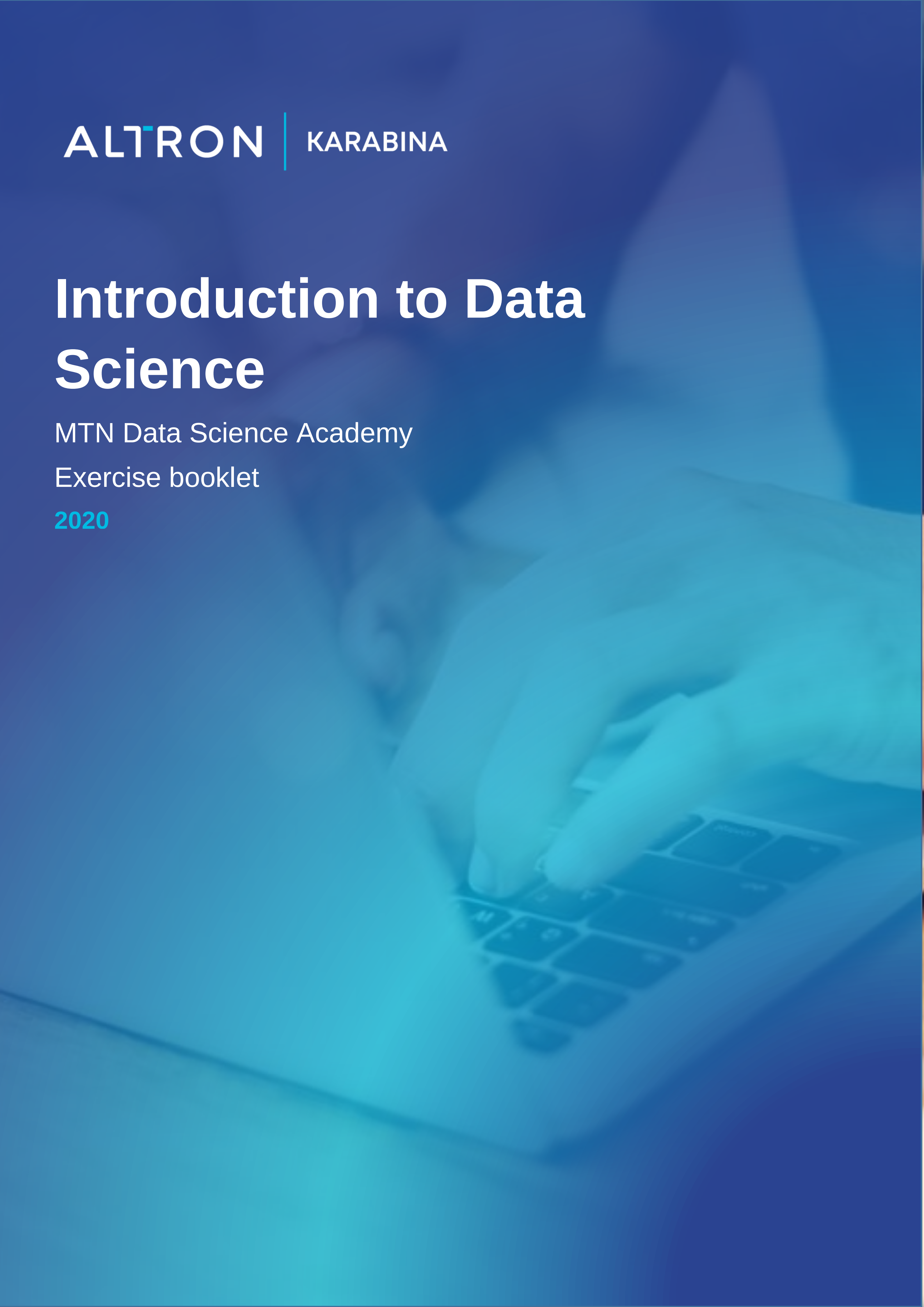


ALTRON | KARABINA

# Introduction to Data Science

MTN Data Science Academy

Exercise booklet

2020

## Table of Contents

## Agenda

- Welcome and sign in
  - Course Overview
- Session 1 : Introduction to data science
  - Basic statistics and machine learning
- Session 2: The data science process
  - Overview of the Data Science process
  - Business understanding/problem understanding
- Session 3 : Ingest and explore data
  - Using Excel to explore & summarize data
  - Using Azure Machine Learning Studio to ingest, explore & summarize data
- Session 4 : Prepare the data
  - Using Azure Machine Learning Studio to clean and prepare data
- Session 5 : Train & test a model
  - Use Azure Machine Learning to build a logistic regression model
  - Use Azure Machine Learning  to test a model
- Session 6: Evaluate model
  - Evaluate a machine learning model in Azure Machine Learning
- Session 7: Business cases
  - HR Churn Model
  - Predictive Maintenance
  - Detecting insurance fraud

## Overview

In this course you will train and evaluate a classification model. This is a supervised machine learning technique in which a set of data with known labels is used to train and test a model. Classification is a type of algorithm used to predict a certain defined outcome; is the result A, or B? Is the e-mail spam or not spam? Is a tumour cancerous or not cancerous? In this course you will use the data set provided to predict employee attrition/churn.

# 1. The Dataset

Throughout this course you will work with a dataset that contains anonymised HR data. The dataset contains information about employees and a target column that indicates whether an employee left the company or not. Employee attrition is a big problem for many companies. Acquiring a new employee can be costly in both time and money, and for this reason companies are going to great lengths to try keep their employees satisfied and happy. One of the ways to keep employees from leaving is to analyse why people who left the company decided to do so, so that we can predict who could be at risk of leaving next and try to take pre-emptive action. The dataset is relatively small containing 1,470 rows (samples) and 35 columns (features).

Please note that the dataset will show an outcome that is specific to that dataset. The predictors of employee churn in this particular dataset may not be the predictors of churn for a different HR dataset.

## 1.1. Dataset Features

The dataset contains the following features:

- Age: The employee age
- Attrition (Target Column): Indicator of whether the employee left the company or not
- BussinessTravel: How frequently the employee travelled on behalf of the business
- DailyRate: Amount the employee costs the company as an hourly rate
- Department: In which department the employee works
- DistanceFromHome: The distance from the employee’s home to the office
- Education: 1 ‘Below College’ 2 ‘College’ 3 ‘Bachelor’ 4 ‘Master’ 5 ‘Doctor’
- EducationField: In which industry the employee has an education
- EmployeeCount: A count of the employees
- EmployeeNumber: A unique ID number assigned to each employee
- EnvironmentSatisfaction: An indication of how happy an employee is with their environment on a scale ranking from 1 to 4 with 1 being ‘Low’, 2 ‘Medium’, 3 ‘High’ and 4 ‘Very High’.
- Gender: The employee’s gender
- HourlyRate: Amount an employee earns as an hourly rate
- JobInvolvement: How involved an employee is in their job on a scale ranking from 1 to 5. Where 1 is very uninvolved and 5 is very involved.
- JobLevel: An indication of how highly an employee’s job ranks, ranging from 1 to 5. With 1 being a low-ranking job (e.g. a junior position) and 5 being a high-ranking job level (e.g. a CEO)
- JobRole: The employee’s job title
- JobSatisfaction: An indication of how satisfied an employee is with their job on a scale ranging from 1 to 4 with 1 being ‘Low’, 2 ‘Medium’, 3 ‘High’ and 4 ‘Very High’.
- MaritalStatus: The employee’s relationship status
- MonthlyIncome: How much the employee earns on a monthly basis
- MonthlyRate: How much the employee costs the company as a monthly rate
- NumCompaniesWorked: The number of companies the employee worked for previously
- Over18: Whether the employee is older than 18 years or not
- OverTime: Whether the employee worked overtime or not
- PercentSalaryHike: The percentage increase the employee has had in their salary
- PerformanceRating: 1 ‘Low’ 2 ‘Good’ 3 ‘Excellent’ 4 ‘Outstanding’
- RelationshipSatisfaction: 1 ‘Low’ 2 ‘Medium’ 3 ‘High’ 4 ‘Very High’
- StandardHours: The number of hours the employee ordinarily works over 2 weeks
- StockOptionLevel: What level of stock option the employee has in the company
- TotalWorkingYears: How many years the employee has been working
- TrainingTimesLastYear: How long the employee spent in training last year
- WorkLifeBalance: An indicator of how healthy the employee’s work life balance is with 1 being ‘Bad’, 2 ‘Good’, 3 ‘Excellent’ and 4 ‘Very High’.
- YearsAtCompany: How many years the employee has been with the company
- YearsInCurrentRole: How many years the employee has been in their current role
- YearsSinceLastPromotion: How many years it has been since the employees last promotion
- YearsWithCurrManager: How many years the employee has been with their current manager.

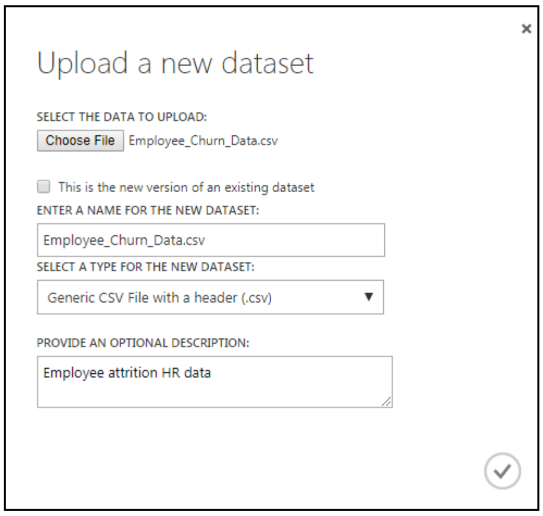
# 2. Ingest and Explore the Data

## 2.1. Using Azure Machine Learning Studio to upload, explore & summarize data

### 2.1.1. Upload Data

Follow these steps to upload the data:

1. If you have not already done so, open a browser and browse to https://studio.azureml.net. Then sign in using the Microsoft account associated with your Azure ML account.
2. Create a new blank experiment and give it the title **Employee Churn Classification.**
3. With the **Employee Churn Classification** experiment open, at the bottom left, click **NEW**. Then in the **NEW** dialog box, click the **DATASET** tab.
4. Click **FROM LOCAL FILE.** Then in the **Upload a new dataset** dialog box, browse to select the **Employee_Churn_Data.csv** file from the folder where you extracted the course files on your local computer and enter the following details as shown below, and then click the **OK** icon:
    - **This is a new version of an existing dataset:** Unselected
    - **Enter a name for the new dataset:** Employee_Churn_Data.csv
    - **Select a type for the new dataset:** Generic CSV file with a header (.csv)
    - **Provide an optional description:** HR data for employee churn


Upload a new dataset

SELECT THE DATA TO UPLOAD:
Choose File Employee_Churn_Data.csv

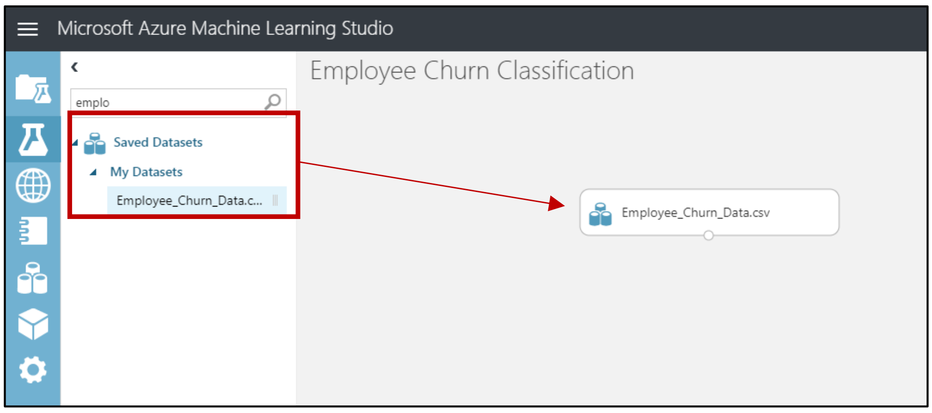
☐ This is the new version of an existing dataset

ENTER A NAME FOR THE NEW DATASET:
Employee_Churn_Data.csv

SELECT A TYPE FOR THE NEW DATASET:
Generic CSV File with a header (.csv)

PROVIDE AN OPTIONAL DESCRIPTION:
Employee attrition HR data

5. Wait for the upload of the dataset to be completed, and then on the experiment items pane, expand **Saved Datasets** and **My Datasets** and drag the **Employee_Churn_Data.csv** dataset onto the canvas.



## 2.1.2. Explore the Data

The Microsoft Azure Machine Learning Studio offers many different ways to visualize and explore data. We are going to be exploring three of these methods:

### 2.1.2.1.1. Visualize the data with Azure ML Studio

1. Right-click the output of the **Employee_Churn_Data.csv** dataset and click **Visualize** to view the data.

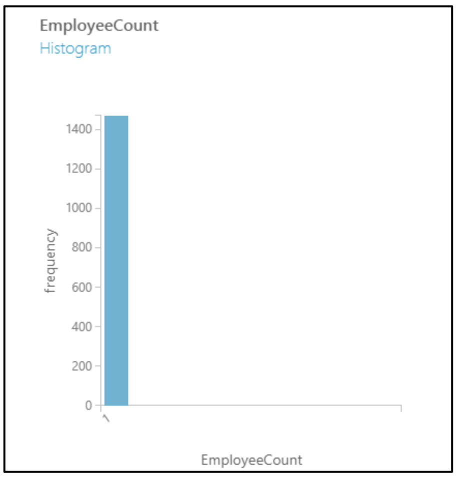
   Note that it contains a number of fields, including an employee number, some demographic information about the employee, some employee survey information such as the employee's level of job satisfaction, some HR data about each employee such as their monthly income and a column named **Attrition** that indicates whether the employee left the company or not.

   Additionally, note how much rich insight we can obtain from our data just by exploring it in this environment offered by Azure ML Studio.

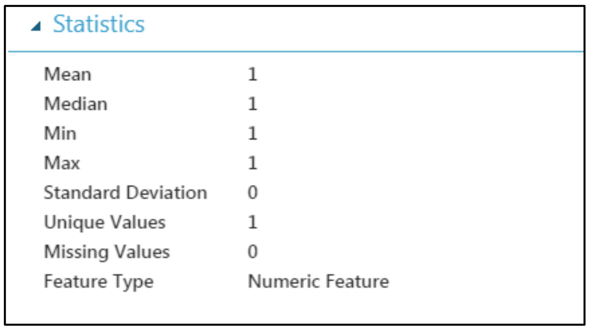
2. Click on the column labelled **Attrition**:

   Note the histogram that is created, showing us that this dataset is skewed. There are far more cases of employees who did not leave the company than there are cases of employees who did leave the company. The skewness of the dataset may mean that we can run into some issues when training the model because of the under representation of cases where employees do actually leave the company. The target variable is under represented and this means that when we train the model it sees more cases of employees not leaving than it does of employees leaving. This may mean that the model is more likely to predict an employee will not leave even when the employee does leave. It is best practice to have a dataset that is as normally distributed as possible.

Click on the column labelled **EmployeeCount**:



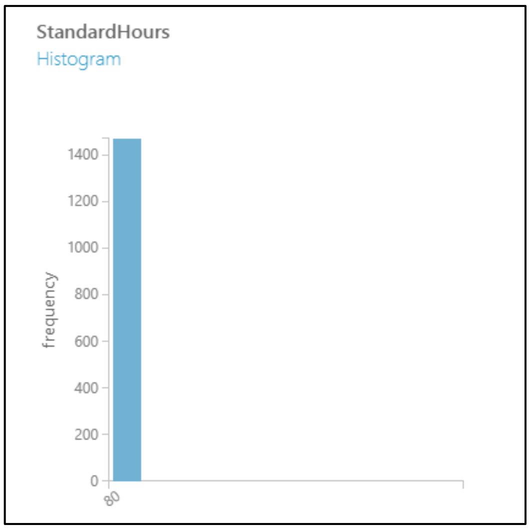
Note the histogram as well as the summary statistics generated for us:

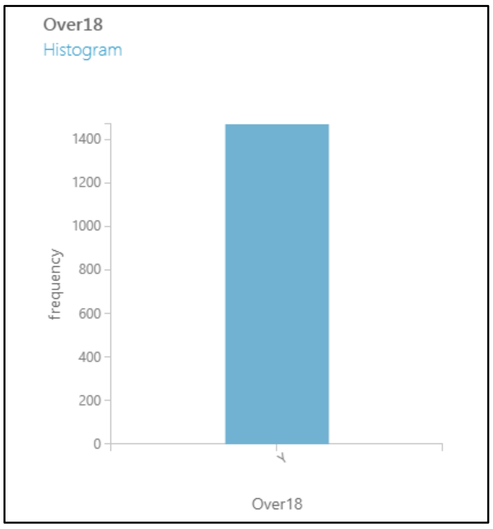


Statistics

| | |
|---|---|
| Mean | 1 |
| Median | 1 |
| Min | 1 |
| Max | 1 |
| Standard Deviation | 0 |
| Unique Values | 1 |
| Missing Values | 0 |
| Feature Type | Numeric Feature |

It is clear that there is only one unique value in this column, and that each employee has the same value for **EmployeeCount**. That value is 1. A variable like this is not very useful to us in a machine learning problem as it creates no distinction between different employees. Each employee has the same **EmployeeCount** value. This column should be removed from the dataset before any model training is done to reduce noise, as the feature adds no value. For the same reason we should remove the **Over18** and the **StandardHours** features:

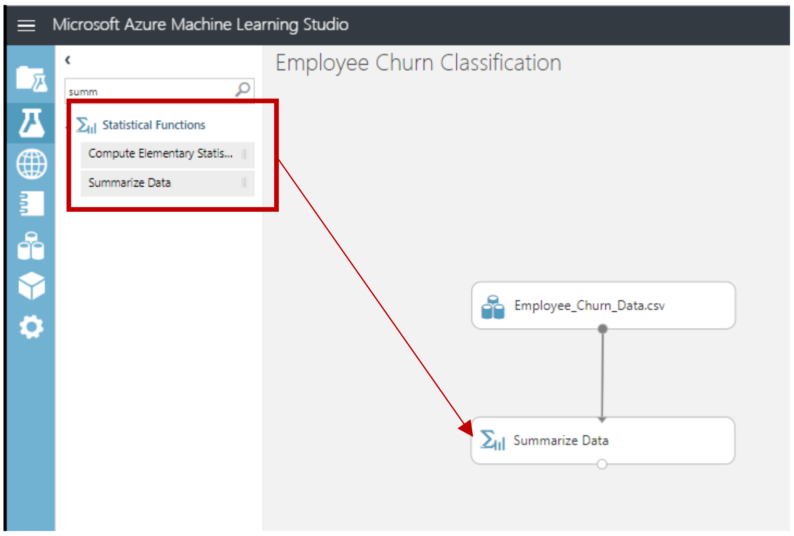




Click on the **EmployeeNumber** column. Note that this column is essentially an ID column. Each employee is given a unique number starting from 1. We can see by scrolling through the preview of this column that no duplicates exist in this column. Again, this column does not tell us anything about the employee and is useful only for admin and security purposes. The ID column will be of no use to us in a machine learning model as it tells us nothing about the difference between employees who left and employees who stayed. This column should also be removed from the dataset before continuing on to training a model to reduce noise.

### 2.1.2.2. Summarize the data with the Summarize Data module

We have had a quick look at each of the columns in our dataset, through visualization and some basic summary statistics. Now we would like to see some summary statistics for our dataset, to give us an overall idea of the layout of our dataset. This will also give us a better idea of how much data cleaning and preparation we will need to do before we can put this data into a model.


Microsoft Azure Machine Learning Studio
Employee Churn Classification
summ
Statistical Functions
Compute Elementary Statis...
Summarize Data
Employee_Churn_Data.csv
Summarize Data

1. Under the **Statistical Functions** tab, drag and drop the **Summarize Data** module onto your experiment:
2. Connect the output of **Employee_Churn_Data.csv** to the input of the **Summarize Data** module.
3. Save and run your experiment
4. Once your experiment has run, Right click on the **Summarize Data** module output and click **Visualize** to view the output.

Notice that we get a wide range of summary statistics that give us a good idea of what our data looks like. The first statistic we are interested in is the **Count** and **Missing Value Count**.

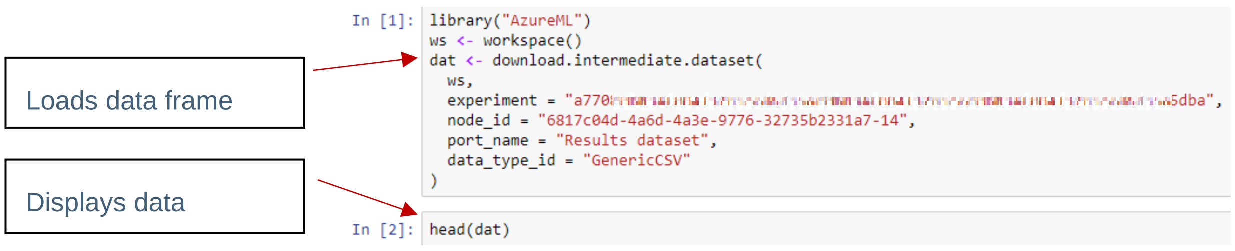
- The **Count** and **Missing Value Count** in combination give us a good idea of which columns have missing data. In real world situations, large amounts of missing data is a problem. In our case it seems the only column containing some missing data is the **DistanceFromHome** feature. All other features have a **Count** of 1470 and a **Missing Value Count** of 0. The **DistanceFromHome** feature has a **Count** of 1454 and a **Missing Value Count** of 16. This is only a small number of missing values, and they all fall in the same column and so we can just remove those rows containing missing values. In datasets containing a greater number of missing values other approaches may need to be taken.
- Note the **Mode** for **EmployeeNumber**. Because each employee has a unique **EmployeeNumber** every single number is listed in the mode, as they are all listed once. This confirms that this column may not provide any value in helping us distinguish between employees who left and employees who stayed.
- Note the **Mean MonthlyIncome** is 6502.931293, but the **Mode** salary is far lower at 2342 and much closer to the **Min** salary which is 1009. This tells us that the company is paying a large number of its employees a lower end salary. This may be something important to keep in mind when we try to determine which features have the greatest effect on whether our employees leave or stay.

### 2.1.2.3. Visualizing the data with R

At this point you should have a good idea of what the data looks like, what data cleaning and preparation is needed, and you may be starting to form some ideas on which features may be affecting whether our employees will stay with or leave the company. Using R, we will create Histograms and boxplots that show the relationship between each numerical feature in the dataset and whether the employee left or stayed, and we will create bar graphs that show the relationship between each categorical feature in the dataset and whether the employee left or stayed.

This data set has both numeric and categorical (string) features. The label column is named **Attrition** and it can have two values “Yes” (Indicating the employee left the company) and “No” (Indicating the employee did not leave the company). Having two values in the label makes this a two-class or binary classification problem. These visualizations will help identify features which will help separate the two classes. These features will exhibit different values when conditioned by the two classes of the label. Other features will show little or no difference when conditioned by the label. Yet, other classes may be degenerate with nearly all values the same regardless of label value.

1. Add a **Convert to CSV** module to the experiment and connect the output of the **Edit Metadata** module to its input. Then run the experiment.
2. Right-click the output of the **Convert to CSV** module and in the **Open in a new workbook** submenu, click **R.**
3. In the new browser tab that opens, at the top of the page, rename the new workbook **Employee Churn Visualization.**
4. Review the code that has been generated automatically. The code in the first cell loads a data frame from your experiment. The code in the second cell displays the data.



```
In [1]: library("AzureML")
        ws <- workspace()
        dat <- download.intermediate.dataset(
          ws,
          experiment = "a770[redacted]5dba",
          node_id = "6817c04d-4a6d-4a3e-9776-32735b2331a7-14",
          port_name = "Results dataset",
          data_type_id = "GenericCSV"
        )

In [2]: head(dat)
```

5. On the **Cell** menu, click **Run All** to run all the cells in the notebook, and then view the output.
6. On the **Insert** menu, click **Insert Cell Below** to add a new cell to the notebook.

In the new cell, enter the following code (Which you can copy from the text file named **R Code Book 1**):

```
install.packages("ggplot2")

library(ggplot2)


bar.plot <- function(x, cut = 200) {

    require(ggplot2)

    if (is.factor(employee[, x]) | is.character(employee[, x]) & (x != 'Attrition') & x != 'Attrition')
{
        colList = c('Attrition', x)
        print(paste('*** The col name = ', x))
        employee[, colList] = lapply(employee[, colList], as.factor)
        sums <- summary(employee[, x], counts = n())
        msk <- names(sums[which(sums > cut)])
        tmp <- employee[employee[, x] %in% msk, colList]
        capture.output(
      if (strsplit(x, '[-]')[[1]][1] == x) {
        g <- ggplot(tmp, aes_string(x)) +
          geom_bar() +
          facet_grid(. ~ Attrition) +
          ggtitle(paste('Attrition by', x))
        print(g)
      }
    )
    }
}

box.plot <- function(x) {
    require(ggplot2)
    if (is.numeric(employee[, x])) {
        ggplot(employee, aes_string('Attrition', x)) +
        geom_boxplot() +
        ggtitle(paste('Attrition by', x))
    }
}

hist.plot <- function(x) {
    require(ggplot2)
    if (is.numeric(employee[, x])) {
        capture.output(
      ggplot(employee, aes_string(x)) +
        geom_histogram() +
        facet_grid(Attrition ~ .) +
        ggtitle(paste('Attrition by', x))
    )
  }
}
```

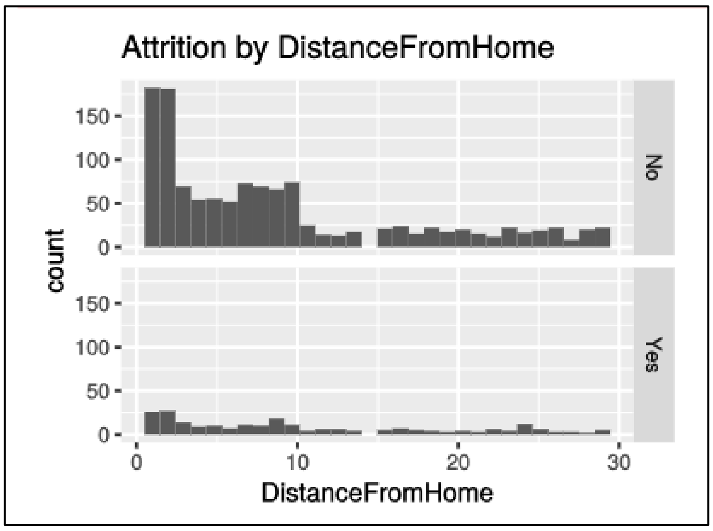
7. Ensure that the cursor is in the cell containing the function definitions above, and then on the **Cell** menu, click **Run and Select Below** (or click the ▶ button on the toolbar).
8. After the code has finished running, in the new empty cell at the bottom of the notebook, enter the following code, and then run the new cell and wait for the code to finish running (Which can be found in **R Code Book 2**):

```
employee <- dat
col.names = names(employee)
col.names = c(col.names, names(employee))
```

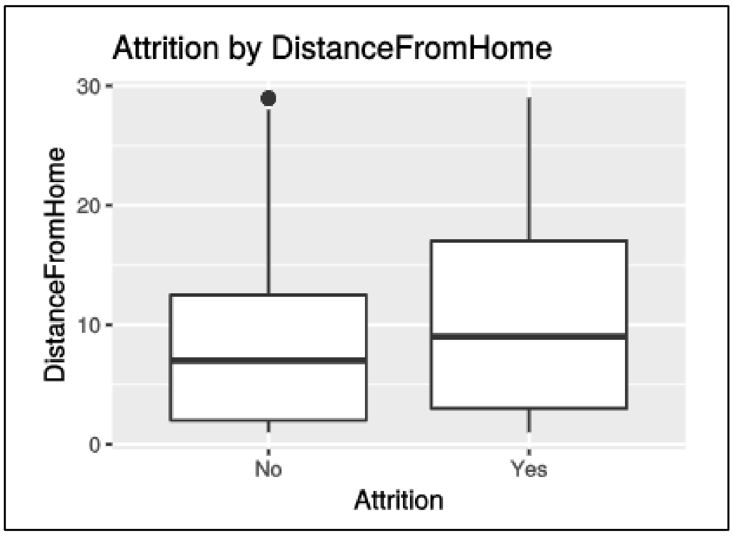
```
lapply(col.names, bar.plot)
lapply(col.names, box.plot)
lapply(col.names, hist.plot)
```

9. View the plots that are produced (You can ignore the error message that is displayed).

You will see a histogram as well as a boxplot for each of the numerical features. Examine these plots to develop an understanding of the relationship between these numerical features and the label values. Note the histogram and boxplot of the **DistanceFromHome** feature, as shown below:

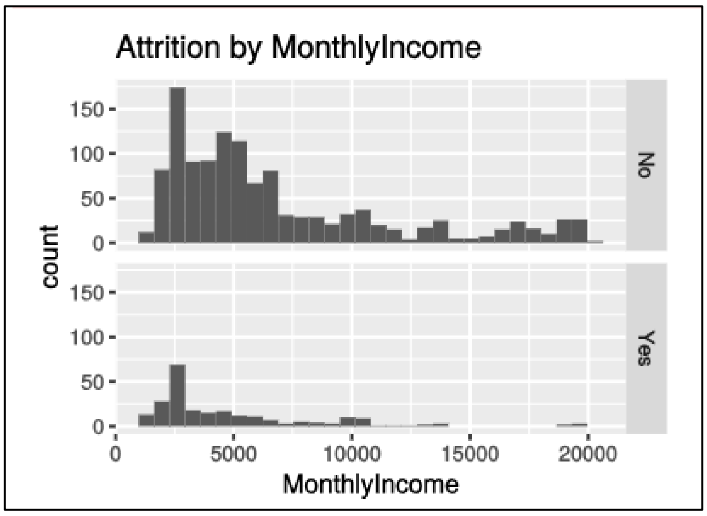


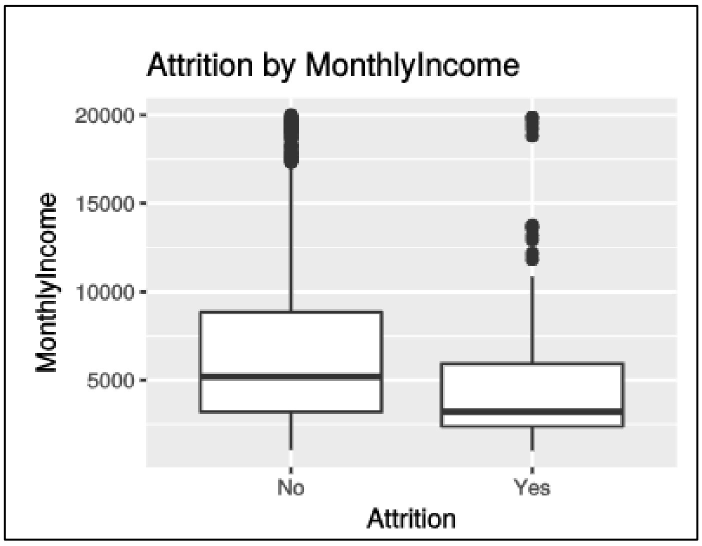
Note with the vertical (frequency) scale set to be identical for each of the value labels (“Yes”, “No”), there are more cases of employee’s not leaving the company than there are of employees leaving the company, indicating that the dataset is slightly skewed. In this case the boxplot may be better to help us view the differences between our employees who leave and those who stay:



Here we see that employees who leave the company tend to have to travel further to get to work than those employees who do not leave. This is represented nicely by the boxplot which shows that the mean distance travelled by employees who leave the company is around 9km whereas the mean distance travelled by employee’s who did not leave the company is closer to 7km. Additionally, quartile 3 for employees who left the company is around 17km and for employees who did not leave the company it is around 13km. This indicates that in general, employees who stayed on at the company appear to live a little closer to the office and may be a factor in why they are not interested in leaving the company as compared to those employees that live further from the office and have to travel a little further to work every day.

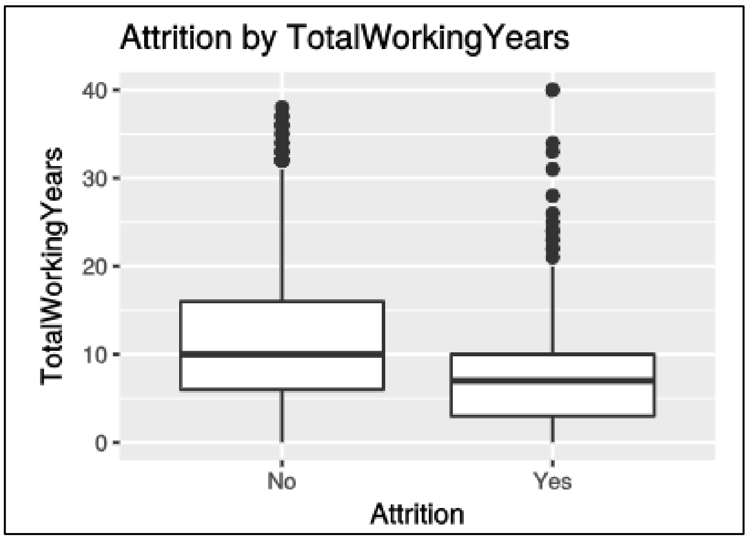
10. Locate the histogram and boxplot for the **MonthlyIncome** feature, as shown below:





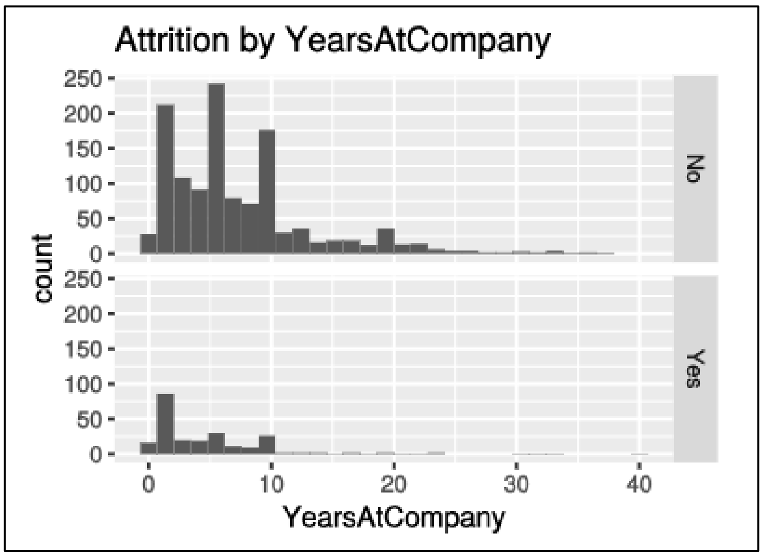
Examine these plots and note the differences between the values for employees who left the company and employees who did not leave the company. The difference is most noticeable in the boxplot where it becomes clear that employees who left the company tend to have a lower monthly income. The mean monthly income for employees who did not leave the company is noticeably higher. It follows, logically, that employees being paid a lower salary may leave the company in pursuit of a job that offers a higher monthly income.

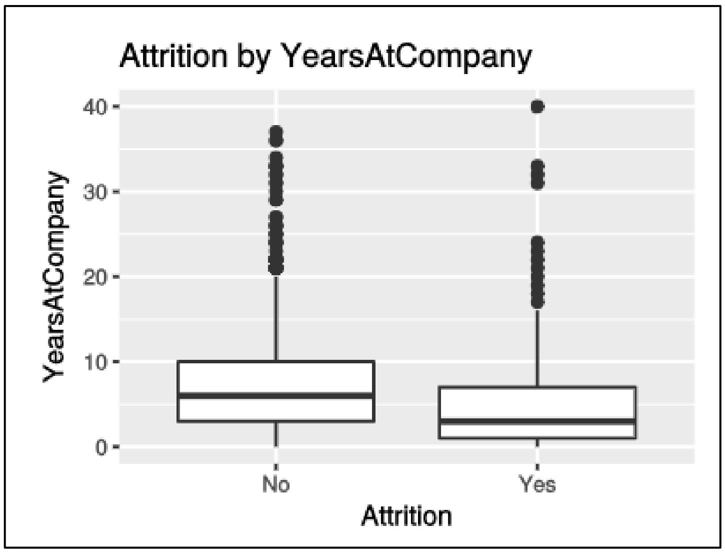
11. Locate the boxplot for the **TotalWorkingYears** feature, as shown below:



Employees who did not leave the company appear to in general have more years of working experience than those employees who leave. Employees that are new to the job market appear to be more likely to leave the company. Something similar can be seen in the **YearsAtCompany** feature.

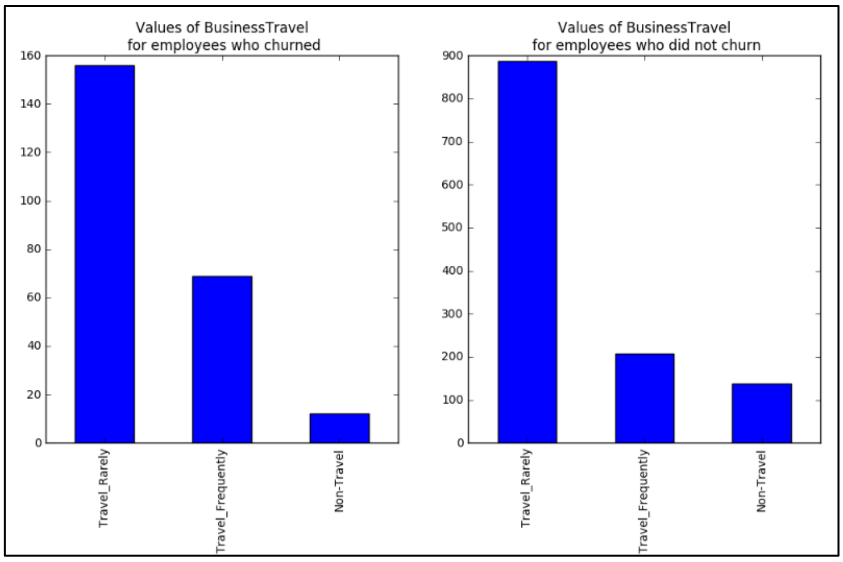
12. Locate the Histogram and Boxplot for the **YearsAtCompany** feature, as shown below:

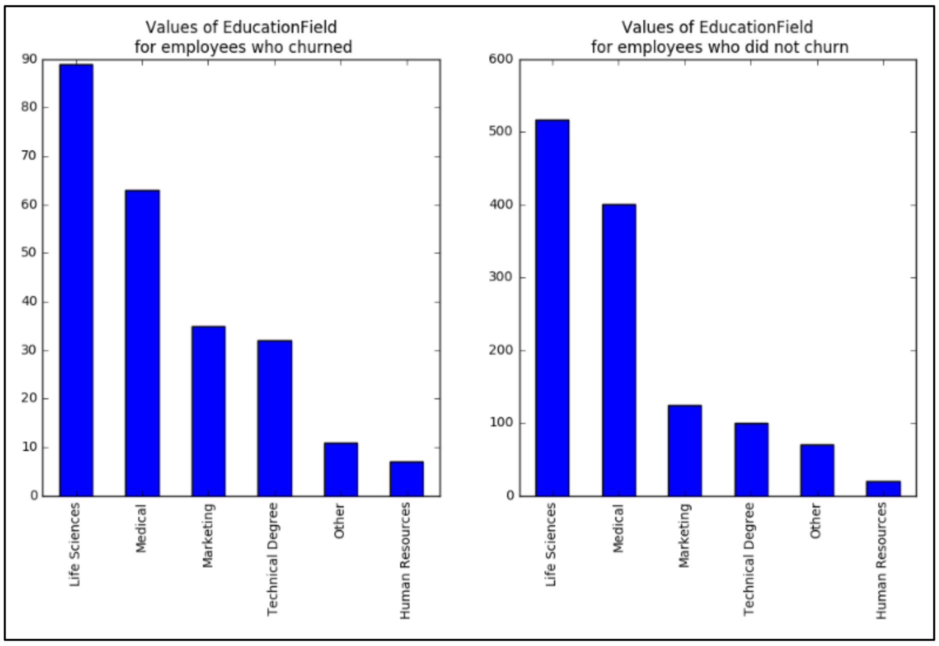


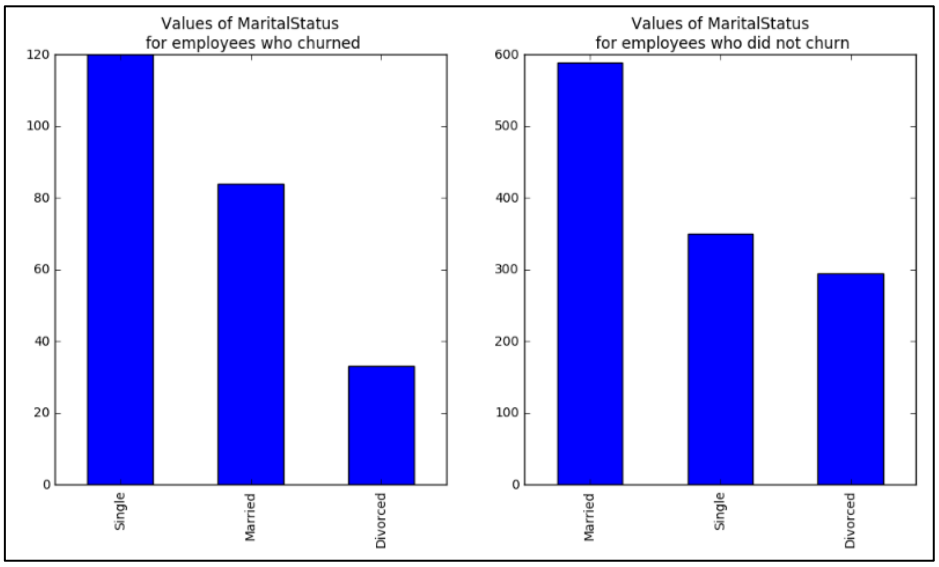


Here again it is highlighted that employees who are new to the company or who have only been working for the company for a few years are more likely to leave the company, than those employees who have been with the company for many years. This also follows somewhat logically, as employees who have been with the company for many years may already be in higher ranking positions, earning larger salaries or may even be in line for more senior ranking positions.

13. Let us now turn our attention to some of the categorical features and the relationship they have with employee attrition, which are visualized in the bar graphs. Locate the bar graph for the **BusinessTravel, Education and MaritalStatus** features, as shown below:





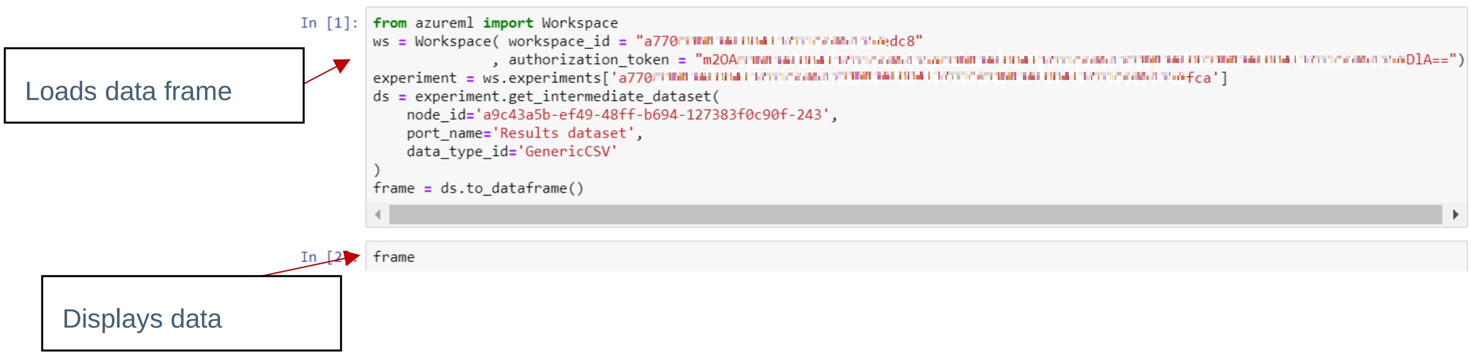


It becomes clear when we look at these bar graphs, showing the relationships between the target feature and some of the categorical features, that in these features there is not much distinction between employees who stayed and those employees who left the company. For example, when looking at the bar graph for the **BusinessTravel** feature, it seems that how much an employee travels for business has no real effect on whether the employee will stay with or leave the company.

## 2.1.2.4. Visualize the data with Python

In this exercise you will create custom Python code to visualize the data set and examine the relationships. This data set has both numeric and categorical (string) features. The label column is named **Attrition** and it can have two values "Yes" (Indicating the employee left the company) and "No" (Indicating the employee did not leave the company). Having two values in the label makes this a two-class or binary classification problem. These visualizations will help identify features which will help separate the two classes. These features will exhibit different values when conditioned by the two classes of the label. Other features will show little or no difference when conditioned by the label. Yet, other classes may be degenerate with nearly all values the same regardless of label value.

14. Add a **Convert to CSV** module to the experiment and connect the output of the **Edit Metadata** module to its input. Then run the experiment.
15. Right-click the output of the **Convert to CSV** module and in the **Open in a new workbook** submenu, click **Python 3.**
16. In the new browser tab that opens, at the top of the page, rename the new workbook **Employee Churn Visualization.**
17. Review the code that has been generated automatically. The code in the first cell loads a data frame from your experiment. The code in the second cell displays the data.



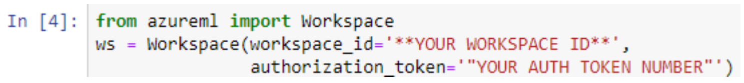
```
In [1]: from azureml import Workspace
        ws = Workspace( workspace_id = "a770[redacted]dc8"
                      , authorization_token = "m2OA[redacted]DlA==")
        experiment = ws.experiments['a770[redacted]fca']
        ds = experiment.get_intermediate_dataset(
            node_id='a9c43a5b-ef49-48ff-b694-127383f0c90f-243',
            port_name='Results dataset',
            data_type_id='GenericCSV'
        )
        frame = ds.to_dataframe()

In [2]: frame
```

18. On the **Cell** menu, click **Run All** to run all of the cells in the notebook, and then view the output.

    If you get the following error while running the first cell:

```
ValueError: workspace_id not provided and not available via config
```

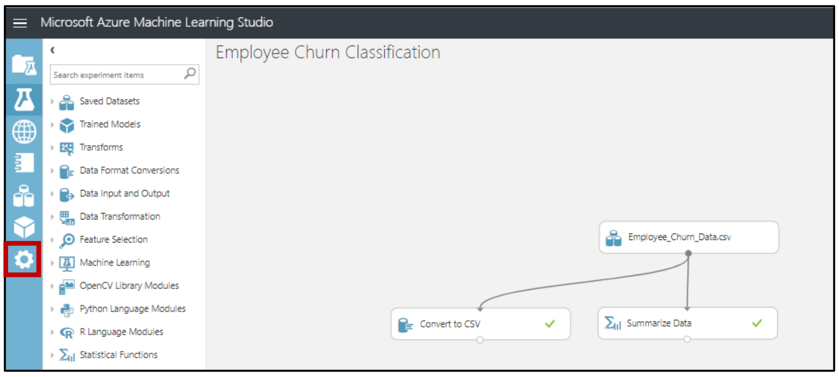
    You will have to manually enter your workspace details. Edit your code so that it looks similar to the code below:



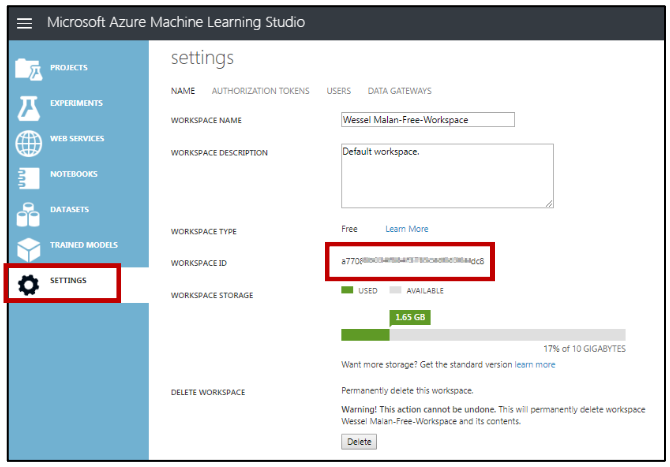
```
In [4]: from azureml import Workspace
        ws = Workspace(workspace_id='**YOUR WORKSPACE ID**',
                       authorization_token='"YOUR AUTH TOKEN NUMBER"')
```

    To find your workspace id:

- Click on **SETTINGS:**



- Under the **Name** tab you will see your **WORKSPACE ID:**



settings

NAME AUTHORIZATION TOKENS USERS DATA GATEWAYS

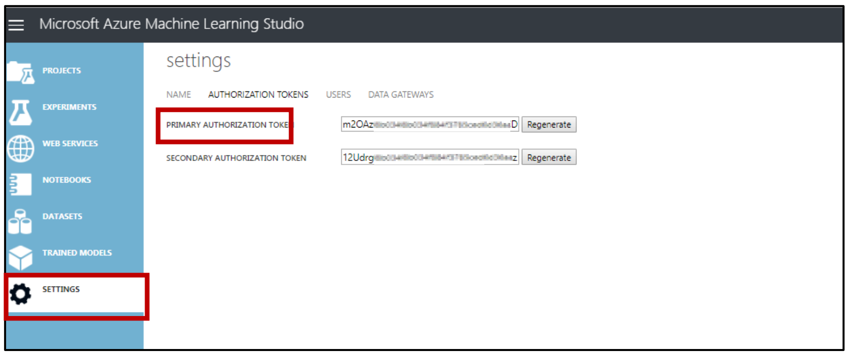
| | |
|---|---|
| WORKSPACE NAME | Wessel Malan-Free-Workspace |
| WORKSPACE DESCRIPTION | Default workspace. |
| WORKSPACE TYPE | Free Learn More |
| WORKSPACE ID | a770[redacted]dc8 |
| WORKSPACE STORAGE | 🟩 USED ▨ AVAILABLE<br>1.65 GB<br>17% of 10 GIGABYTES<br>Want more storage? Get the standard version learn more |
| DELETE WORKSPACE | Permanently delete this workspace.<br>Warning! This action cannot be undone. This will permanently delete workspace Wessel Malan-Free-Workspace and its contents.<br>Delete |

- Copy your workspace ID and paste it into your code block:

```
In [4]: from azureml import Workspace
        ws = Workspace(workspace_id='a770[redacted]dc8',
```

To find your authorization token:'

- In **SETTINGS** click on the tab named **AUTHORIZATION TOKENS:**

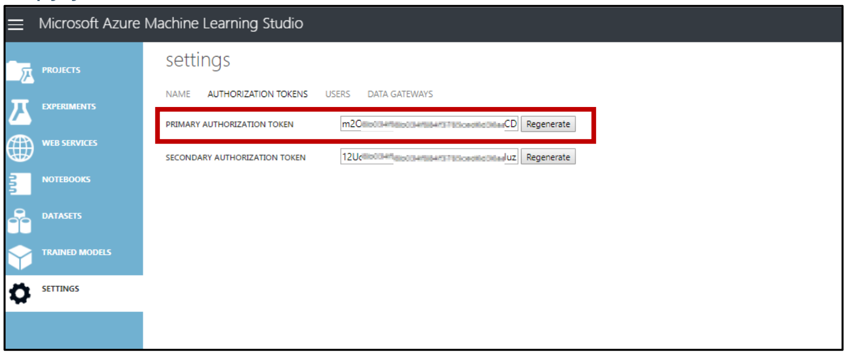


settings

NAME AUTHORIZATION TOKENS USERS DATA GATEWAYS

| | | |
|---|---|---|
| PRIMARY AUTHORIZATION TOKEN | m2OAz[redacted]D | Regenerate |
| SECONDARY AUTHORIZATION TOKEN | 12Udrg[redacted]z | Regenerate |

- Copy your **PRIMARY AUTHORIZATION TOKEN:**



settings

NAME AUTHORIZATION TOKENS USERS DATA GATEWAYS

| | | |
|---|---|---|
| PRIMARY AUTHORIZATION TOKEN | m2C[redacted]CD | Regenerate |
| SECONDARY AUTHORIZATION TOKEN | 12Uc[redacted]uz | Regenerate |

- Paste your authorization token in your code block:

```
In [4]: from azureml import Workspace
        ws = Workspace(workspace_id='a770[redacted]dc8',
                       authorization_token='m2OAz[redacted]sqhDlA==')
```

- Run all code again

19. On the **Insert** menu, click **Insert Cell Below** to add a new cell to the notebook.
20. In the new cell, enter the following code (which you can copy and paste from name_where_code_stored***):

```
def employee_bar(df):
    import matplotlib
    matplotlib.use('agg')
    import numpy as np
    import matplotlib.pyplot as plt

    ## Create a series of bar plots for the various levels of the
    ## string columns in the data frame by readmi_class
    names = df.columns.tolist()
    for col in names:
        if(df[col]. dtype not in [np.int64, np.int32, np.float64]):
            temp1 = df.ix[df.Attrition == 'Yes', col].value_counts()
            temp0 = df.ix[df.Attrition == 'No', col].value_counts()

            fig = plt.figure(figsize = (12,6))
            fig.clf()
            ax1 = fig.add_subplot(1, 2, 1)
            ax0 = fig.add_subplot(1, 2, 2)
            temp1.plot(kind = 'bar', ax = ax1)
            ax1.set_title('Values of ' + col + '\n for employees who churned')
            temp0.plot(kind = 'bar', ax = ax0)
            ax0.set_title('Values of ' + col + '\n for employees whop did not churn')
            fig.savefig('bar_' + col + '.png')

    return 'Done'

def employee_box(df):
    import matplotlib
    matplotlib.use('agg')
    import numpy as np
    import matplotlib.pyplot as plt

    ## Now make box plots of the columns with numerical values
    names = df.columns.tolist()
    for col in names:
        if(df[col].dtype in [np.int64, np.int32, np.float64]):
            temp1 = df.ix[df.Attrition == 'Yes', col]
            temp0 = df.ix[df.Attrition == 'No', col]

            fig = plt.figure(figsize = (12,6))
            fig.clf()
            ax1 = fig.add_subplot(1, 2, 1)
            ax0 = fig.add_subplot(1, 2, 2)
            ax1.boxplot(temp1.as_matrix())
            ax1.set_title('Box plot of ' + col + '\n for employees who churned')
            ax0.boxplot(temp0.as_matrix())
            ax0.set_title('Box plot of ' + col + '\n for employees who did not churn')
            fig.savefig('box_' + col + '.png')
    return 'Done'

def employee_hist(df):
    import matplotlib
    matplotlib.use('agg')
    import numpy as np
    import matplotlib.pyplot as plt

    ## Now make histograms of the columns with numerical values
    names = df.columns.tolist()
    for col in names:
        if(df[col].dtype in [np.int64, np.int32, np.float64]):
            temp1 = df.ix[df.Attrition == 'Yes', col]
            temp0 = df.ix[df.Attrition == 'No', col]

            fig = plt.figure(figsize = (12,6))
            fig.clf()
            ax1 = fig.add_subplot(1, 2, 1)
```

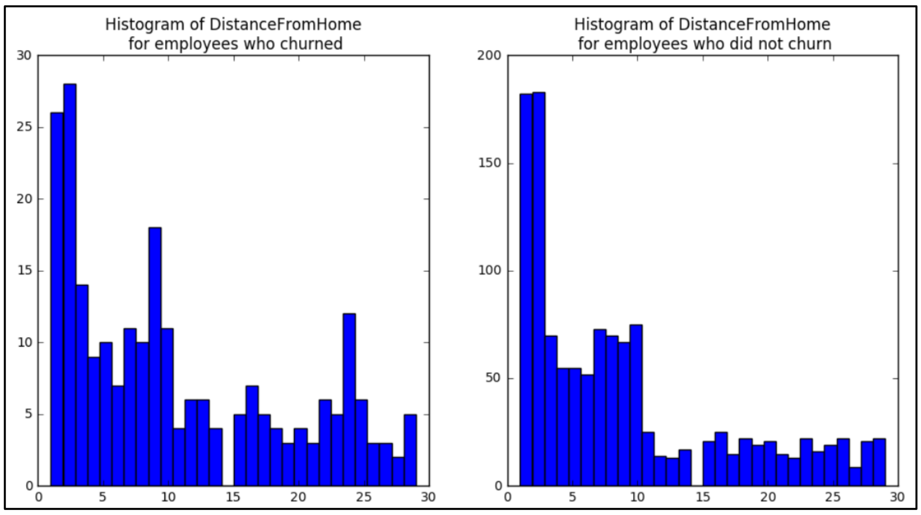
```
                ax0 = fig.add_subplot(1, 2, 2)
                ax1.hist(temp1.as_matrix(), bins = 30)
                ax1.set_title('Histogram of ' + col + '\n for employees who churned')
                ax0.hist(temp0.as_matrix(), bins = 30)
                ax0.set_title('Histogram of ' + col + '\n for employees who did not churn')
                fig.savefig('hist_' + col + '.png')
        return 'Done'
```

21. Ensure that the cursor is in the cell containing the function definitions above, and then on the **Cell** menu, click **Run and Select Below** (or click the ▶ button on the toolbar).
22. After the code has finished running, in the new empty cell at the bottom of the notebook, enter the following code, and then run the new cell and wait for the code to finish running:

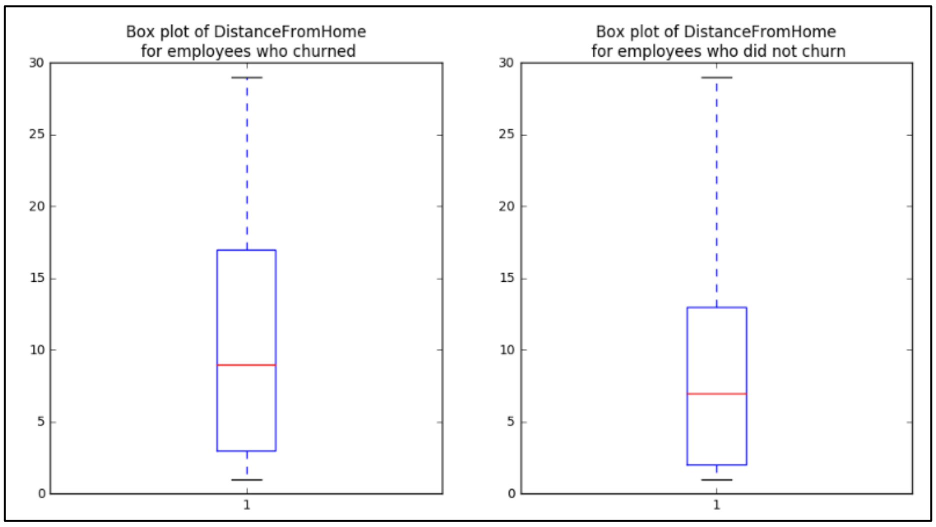
```
%matplotlib inline
employee_hist(frame)
employee_bar(frame)
employee_box(frame)
```

23. View the plots that are produced. You can ignore the warner message that may be displayed.

You will see a histogram as well as a boxplot for each of the numerical features. Examine these plots to develop an understanding of the relationship between these numerical features and the label values. Note the histogram and boxplot of the **DistanceFromHome** feature, as shown below:

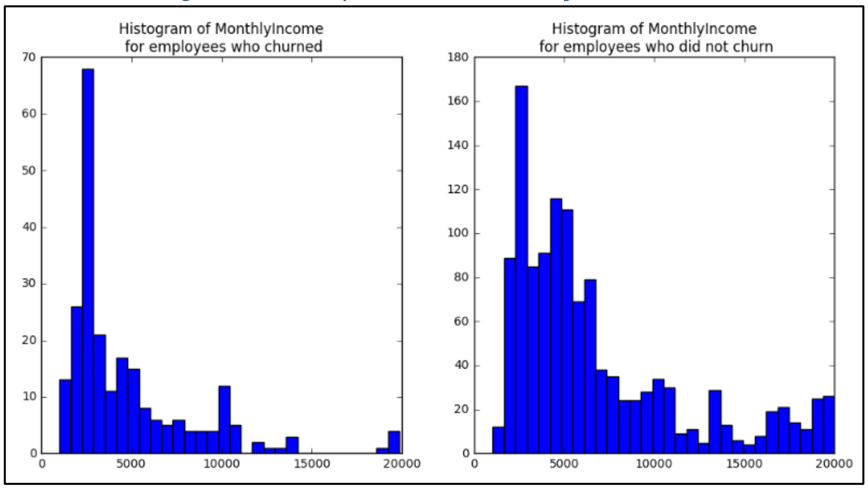


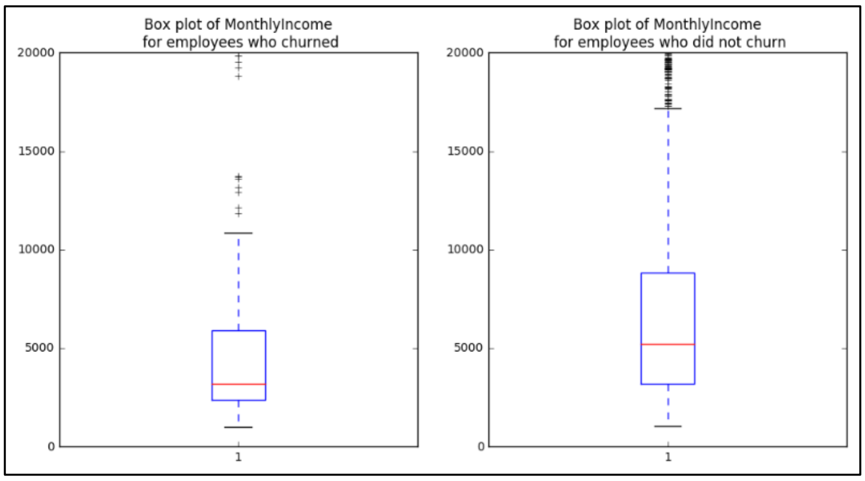
Note, the vertical (frequency) scale is not identical for each of the value labels ("Yes", "No"). There are more cases of employee's not leaving the company than there are of employees leaving the company, indicating that the dataset is slightly skewed. In this case the boxplot may be better to help us view the difference's between our employees who leave and those who stay:



Here we see that employees who leave the company tend to have to travel further to get to work than those employees who do not leave. This is represented nicely by the boxplot which shows that the mean distance travelled by employees who leave the company is around 9km whereas the mean distance travelled by employee's who did not leave the company is closer to 7km. Additionally Q3 for employees who left the company is around 17km and for employees who did not leave the company it is around 13km.

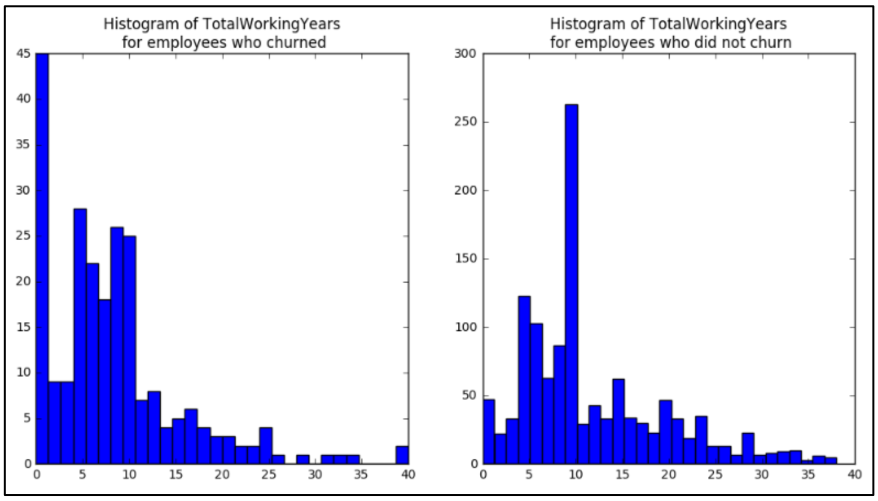
24. Locate the histogram and boxplot for the **MonthlyIncome** feature, as shown below:

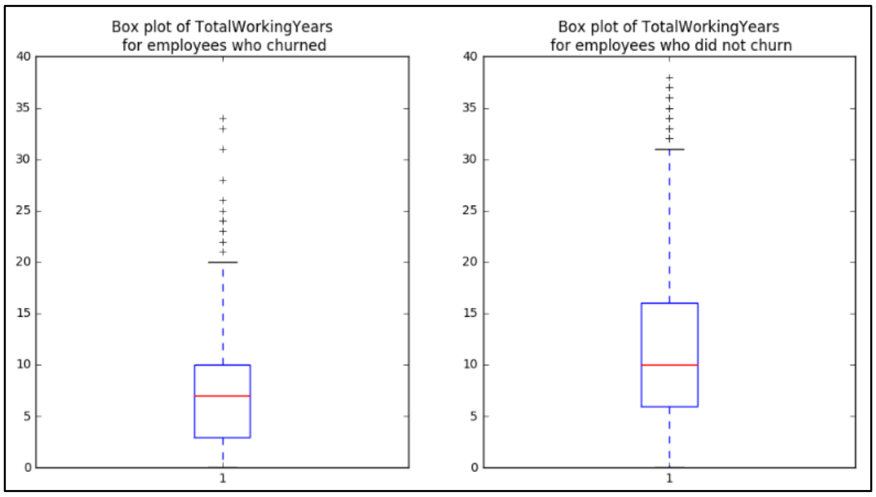




Examine these plots and note the differences between the values for employees who left the company and employees who did not leave the company. The difference is most noticeable in the boxplot where it becomes clear that employees who left the company tend to have a lower monthly income. The mean monthly income for employees who did not leave the company is noticeably higher. It follows logically that employees being paid a lower salary may leave the company in pursuit of a job that offers a higher monthly income.

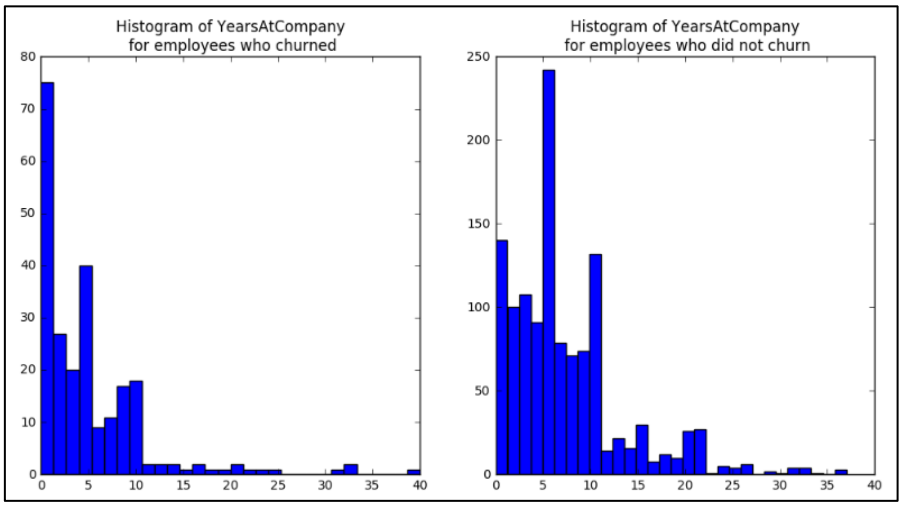
25. Locate the histogram and boxplot for the **TotalWorkingYears** feature, as shown below:

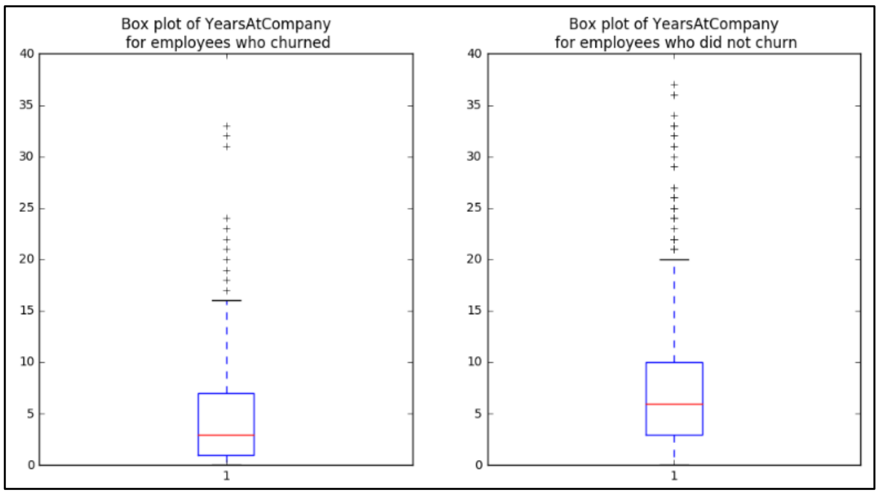




Employees who did not leave the company appear to in general have more years of working experience than those employees who leave. Employees that are new to the job market appear to have a better chance of leaving the company. Something similar can be seen in the **YearsAtCompany** feature.

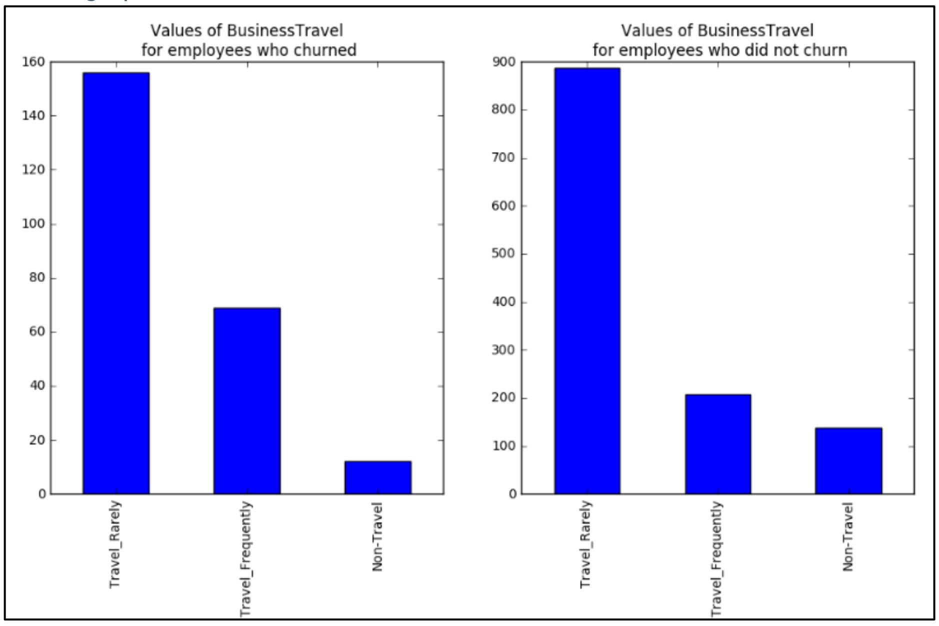
26. Locate the **YearsAtCompany** feature, as shown below:

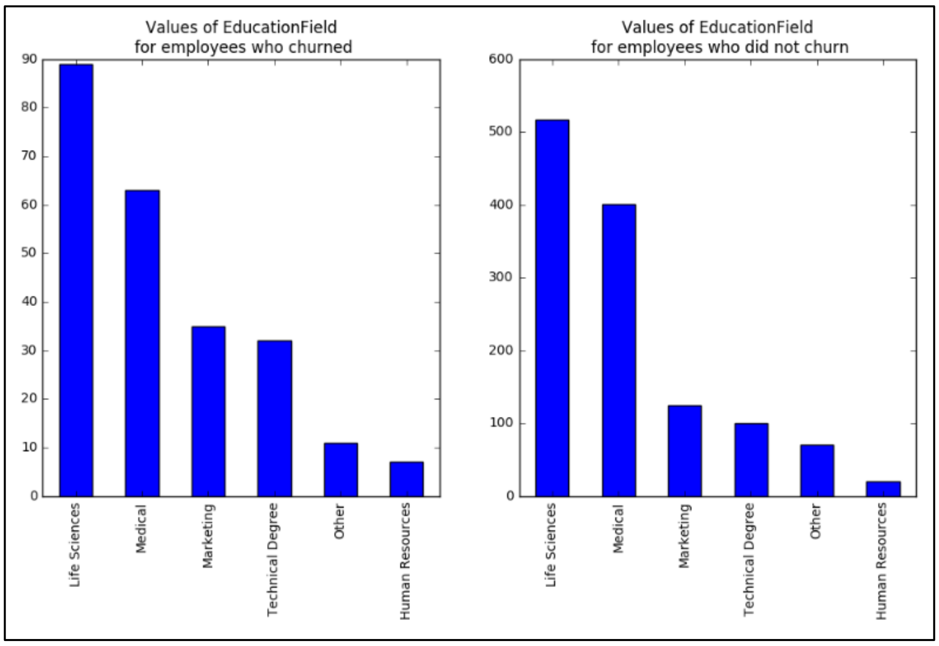


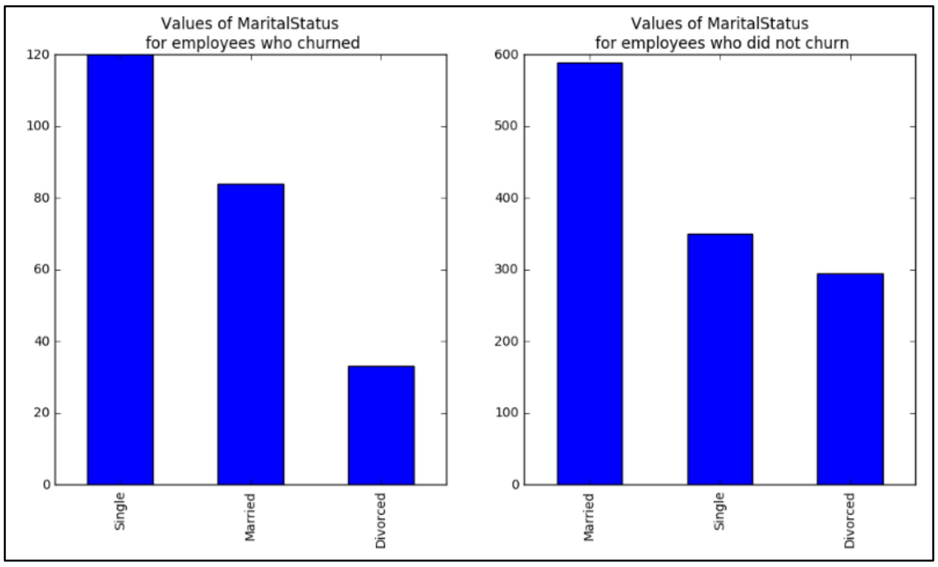


Here again it is highlighted that employees who are new to the company or who have only been working for the company for a few years are more likely to leave the company, than those employees who have been with the company for many years. This also follows somewhat logically, as employees who have been with the company for many years may already be in higher ranking positions, earning larger salaries or may even be in line for more senior ranking positons.

27. Locate the bar graph for the **BusinessTravel, Education and MaritalStatus** features, as shown below:







It becomes clear when we look at these bar graphs, showing the relationships between the target feature and some of the categorical features, that there is not much distinction between employees who stayed and those employees who left the company. For example, when looking at the bar graph for the **BusinessTravel** feature, it seems that how much an employee travels for business has no real effect on whether the employee will stay with or leave the company.

However to be truly sure of this we may need to visualize this data slightly differently. It may be better to view these particular features using a different visualization that gives us a better idea of the interaction between each categorical feature and the target feature. For this visualization we will use Power BI.

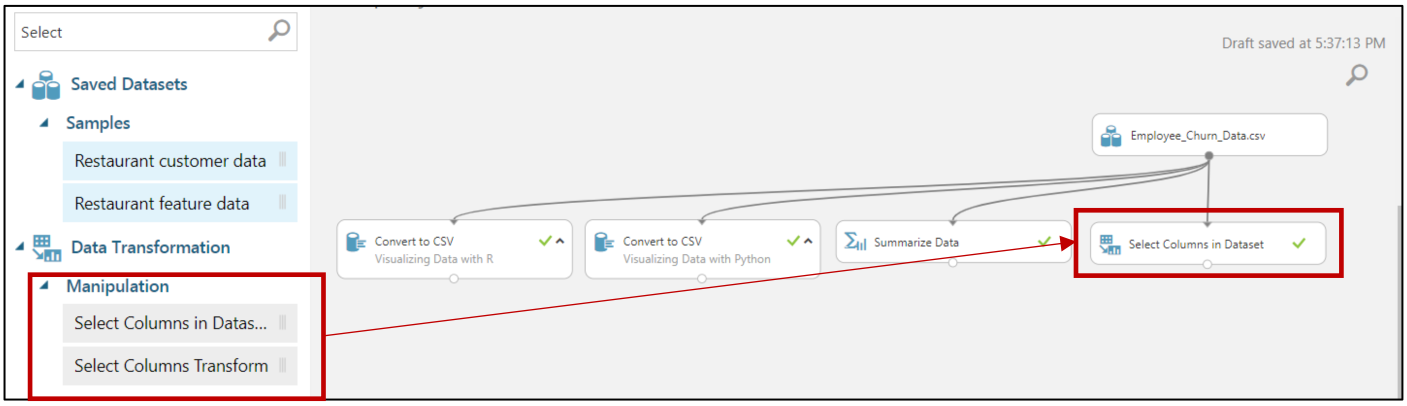
# 3. Prepare the Data

## 3.1. Using Azure Machine Learning Studio to clean and prepare data

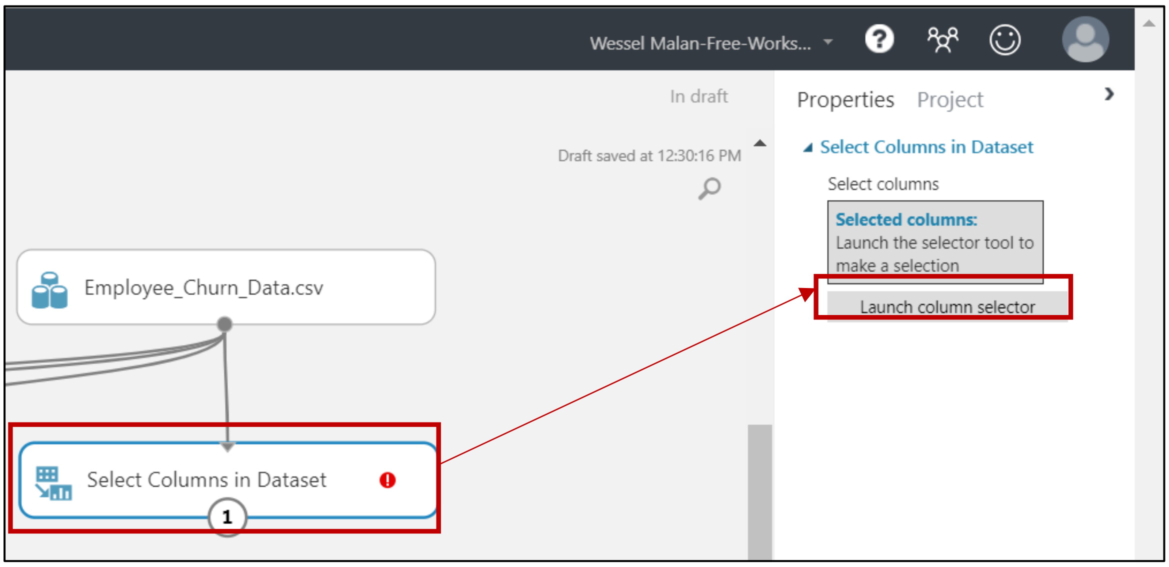
### 3.1.1. Remove columns

While exploring and visualizing the data we came across some columns that seem to have no relationship with the target variable and as a result cause noise in the machine learning model.

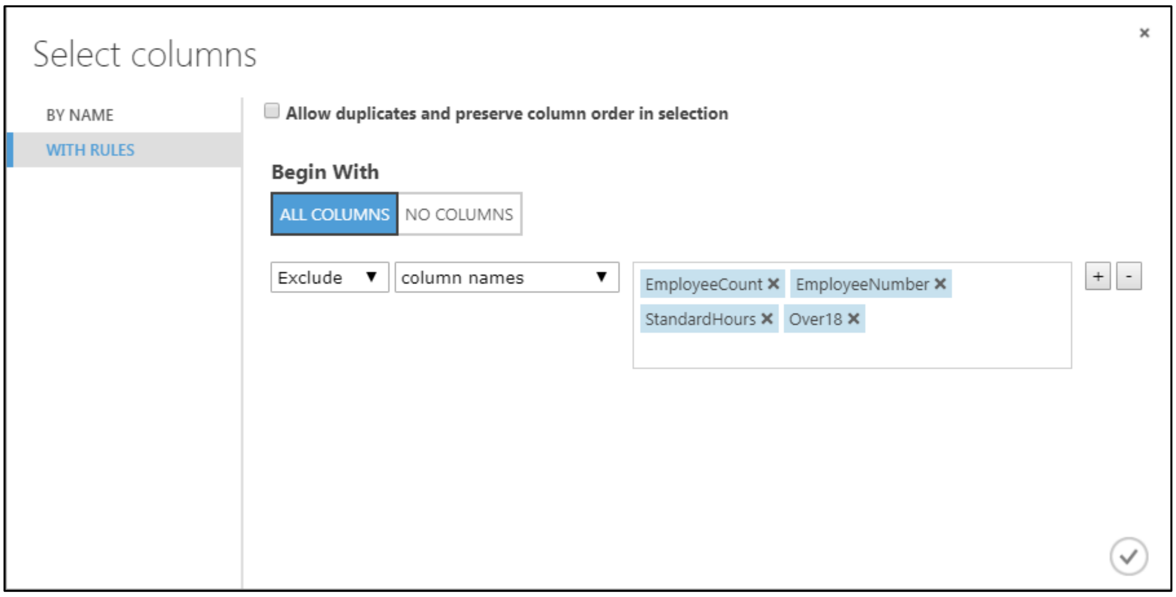
1. To remove these columns from the dataset add a **Select Columns in Dataset** module to your experiment and connect the output from **Employee_Churn_data.csv** to the **Select Columns in Dataset** modules input:



2. With **Select Columns in Dataset** selected, click on **Launch column selector** in the properties pane to the right:


Wessel Malan-Free-Works...
In draft
Draft saved at 12:30:16 PM
Employee_Churn_Data.csv
Select Columns in Dataset
1
Properties
Project
Select Columns in Dataset
Select columns
Selected columns:
Launch the selector tool to make a selection
Launch column selector

3. Choose to select columns **WITH RULES** and **EXCLUDE** the following columns:
   - EmployeeCount
   - EmployeeNumber
   - StandardHours
   - Over18



Select columns

BY NAME

WITH RULES

☐ Allow duplicates and preserve column order in selection

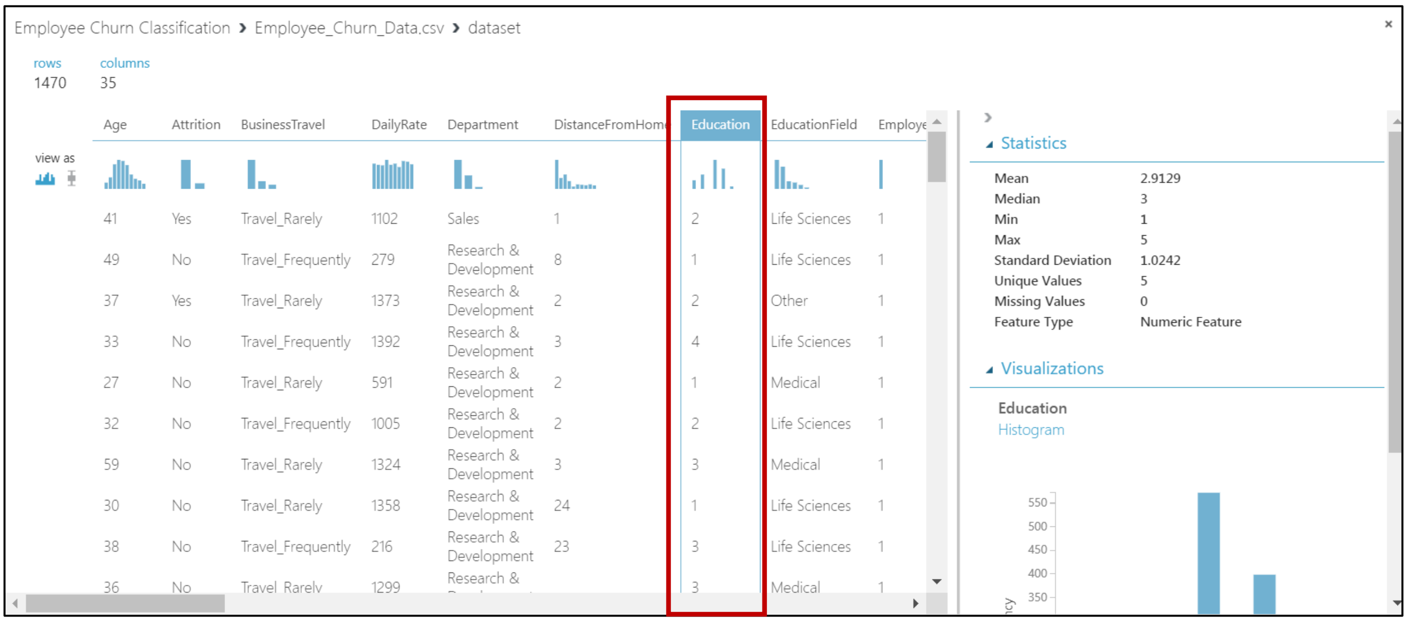
Begin With

ALL COLUMNS | NO COLUMNS

Exclude ▼ | column names ▼ | EmployeeCount ✖ EmployeeNumber ✖ StandardHours ✖ Over18 ✖

## 3.1.2. Correct data types

During data exploration and visualization we also noticed some numerical columns which would be better represented as categorical features, such as the **Education** and **EnvironmentSatisfaction** features:



Employee Churn Classification › Employee_Churn_Data.csv › dataset

| rows | columns |
|---|---|
| 1470 | 35 |

| Age | Attrition | BusinessTravel | DailyRate | Department | DistanceFromHome | Education | EducationField | Employe |
|---|---|---|---|---|---|---|---|---|
| 41 | Yes | Travel_Rarely | 1102 | Sales | 1 | 2 | Life Sciences | 1 |
| 49 | No | Travel_Frequently | 279 | Research & Development | 8 | 1 | Life Sciences | 1 |
| 37 | Yes | Travel_Rarely | 1373 | Research & Development | 2 | 2 | Other | 1 |
| 33 | No | Travel_Frequently | 1392 | Research & Development | 3 | 4 | Life Sciences | 1 |
| 27 | No | Travel_Rarely | 591 | Research & Development | 2 | 1 | Medical | 1 |
| 32 | No | Travel_Frequently | 1005 | Research & Development | 2 | 2 | Life Sciences | 1 |
| 59 | No | Travel_Rarely | 1324 | Research & Development | 3 | 3 | Medical | 1 |
| 30 | No | Travel_Rarely | 1358 | Research & Development | 24 | 1 | Life Sciences | 1 |
| 38 | No | Travel_Frequently | 216 | Research & Development | 23 | 3 | Life Sciences | 1 |
| 36 | No | Travel_Rarely | 1299 | Research & | | 3 | Medical | 1 |

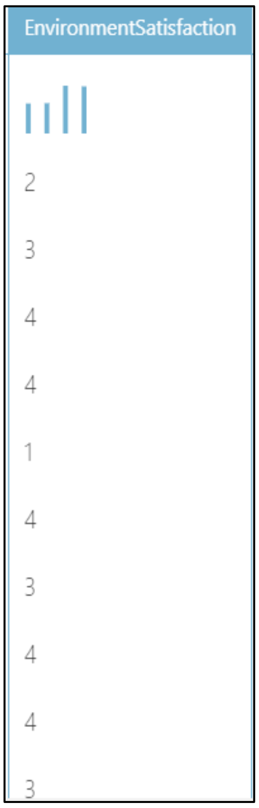
Statistics

| | |
|---|---|
| Mean | 2.9129 |
| Median | 3 |
| Min | 1 |
| Max | 5 |
| Standard Deviation | 1.0242 |
| Unique Values | 5 |
| Missing Values | 0 |
| Feature Type | Numeric Feature |

Visualizations

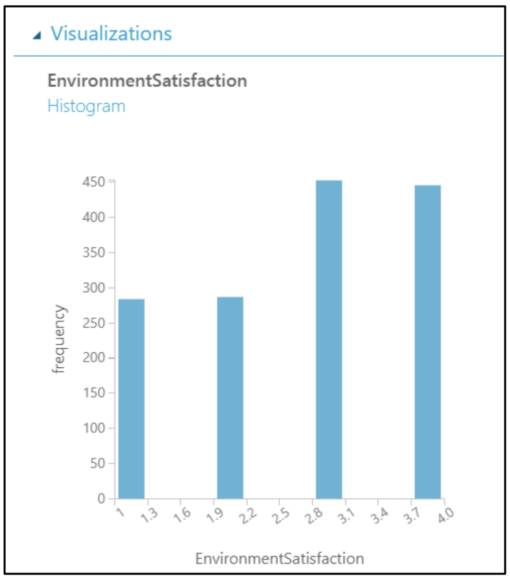
Education

Histogram



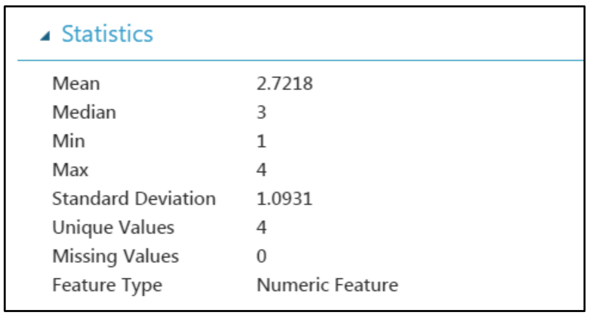
EnvironmentSatisfaction

2
3
4
4
1
4
3
4
4
3



Visualizations

EnvironmentSatisfaction
Histogram

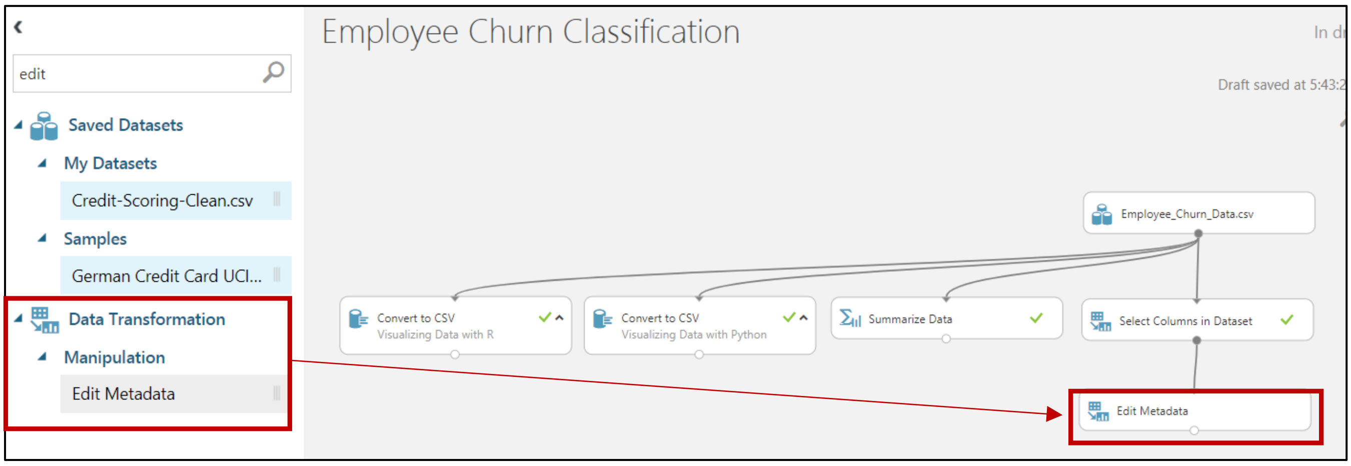


Statistics

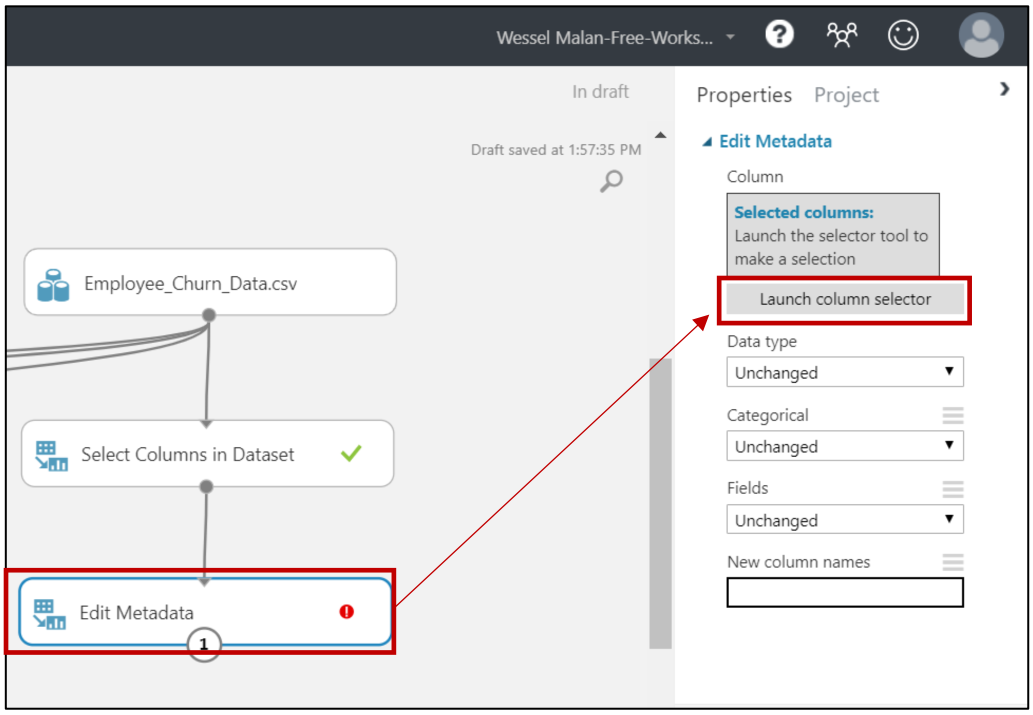
| | |
|---|---|
| Mean | 2.7218 |
| Median | 3 |
| Min | 1 |
| Max | 4 |
| Standard Deviation | 1.0931 |
| Unique Values | 4 |
| Missing Values | 0 |
| Feature Type | Numeric Feature |

Note the feature type of each of the variables above as well as the visualization. The feature type is numeric and the visualization shows that the number is being stored as a floating point numeric variable. Because we know the data, we know that these types of features would be better represented as categorical features as they are essentially points on a scale ranging from 1 to 4 as opposed to truly numeric features.

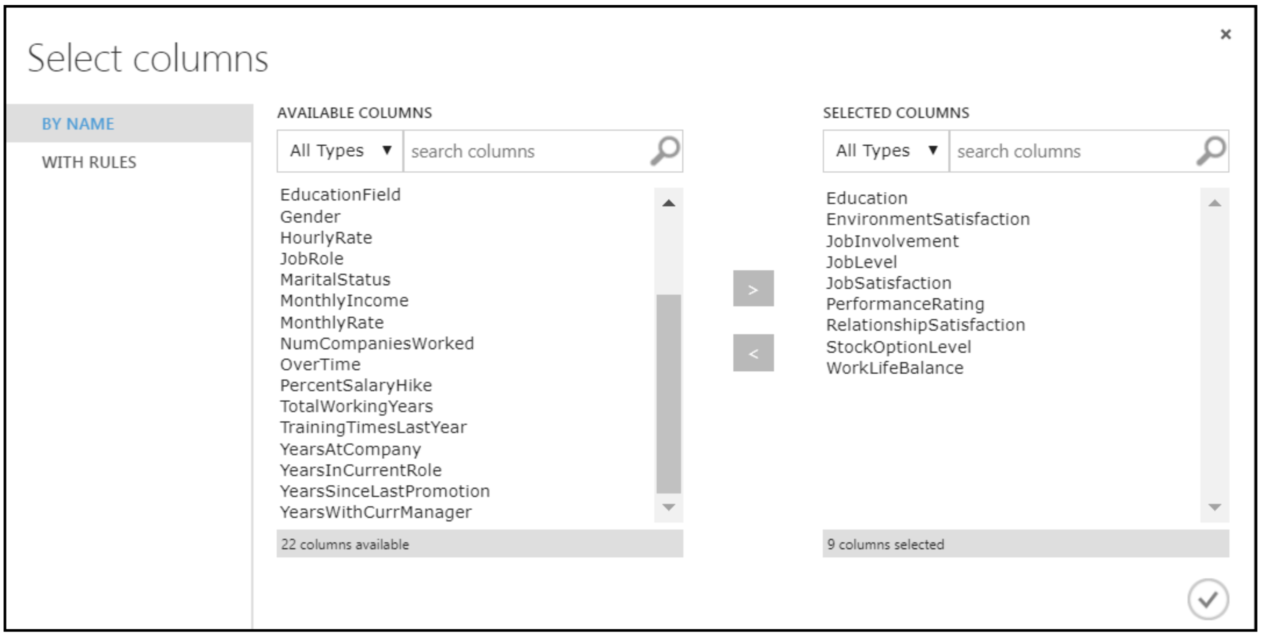
1. Add an **Edit Metadata** module to your experiment and connect the output from **Select Columns in Dataset** to the **Edit Metdata** modules input:



2. With **Edit Metadata** selected, click on **Launch column selector** in the properties pane to the right:



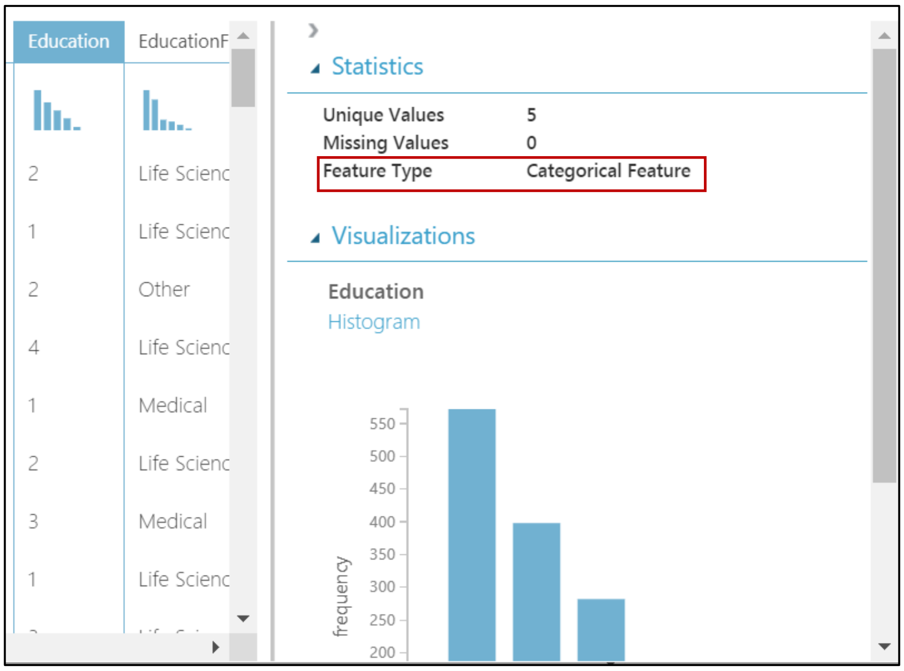
4. Choose to select columns **BY NAME** and select the following columns as shown below:
   - Education
   - EnvironmentSatisfaction
   - JobInvolvement
   - JobLevel
   - JobSatisfaction
   - PerformanceRating
   - RelationshipSatisfaction
   - StockOptionLevel
   - WorkLifeBalance



```
EducationField
Gender
HourlyRate
JobRole
MaritalStatus
MonthlyIncome
MonthlyRate
NumCompaniesWorked
OverTime
PercentSalaryHike
TotalWorkingYears
TrainingTimesLastYear
YearsAtCompany
YearsInCurrentRole
YearsSinceLastPromotion
YearsWithCurrManager
22 columns available
```

```
Education
EnvironmentSatisfaction
JobInvolvement
JobLevel
JobSatisfaction
PerformanceRating
RelationshipSatisfaction
StockOptionLevel
WorkLifeBalance
9 columns selected
```

5. Set the remaining properties of your **Edit Metadata** to match the following details:
   - **Data type:** Unchanged
   - **Categorical:** Make categorical
   - **Fields:** Unchanged
   - **New column names:** *Leave blank*
6. Save and run your experiment
7. Once your experiment has run, right click on the output of your **Edit Metadata** output and **Visualize** the results.
8. Select some of the features that you have just edited and note that they are now stored as categorical features:



| Education | EducationF |
|---|---|
| 2 | Life Scienc |
| 1 | Life Scienc |
| 2 | Other |
| 4 | Life Scienc |
| 1 | Medical |
| 2 | Life Scienc |
| 3 | Medical |
| 1 | Life Scienc |

Statistics

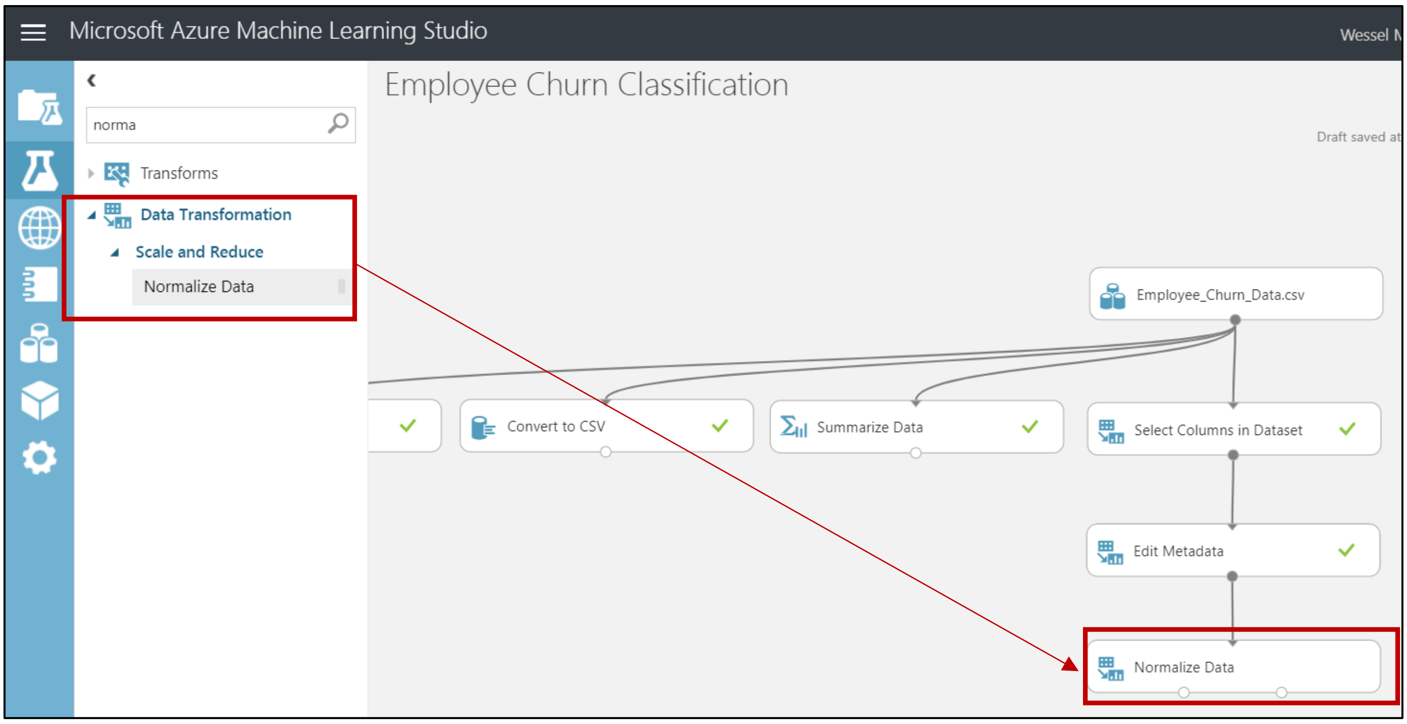
| | |
|---|---|
| Unique Values | 5 |
| Missing Values | 0 |
| Feature Type | Categorical Feature |

Visualizations

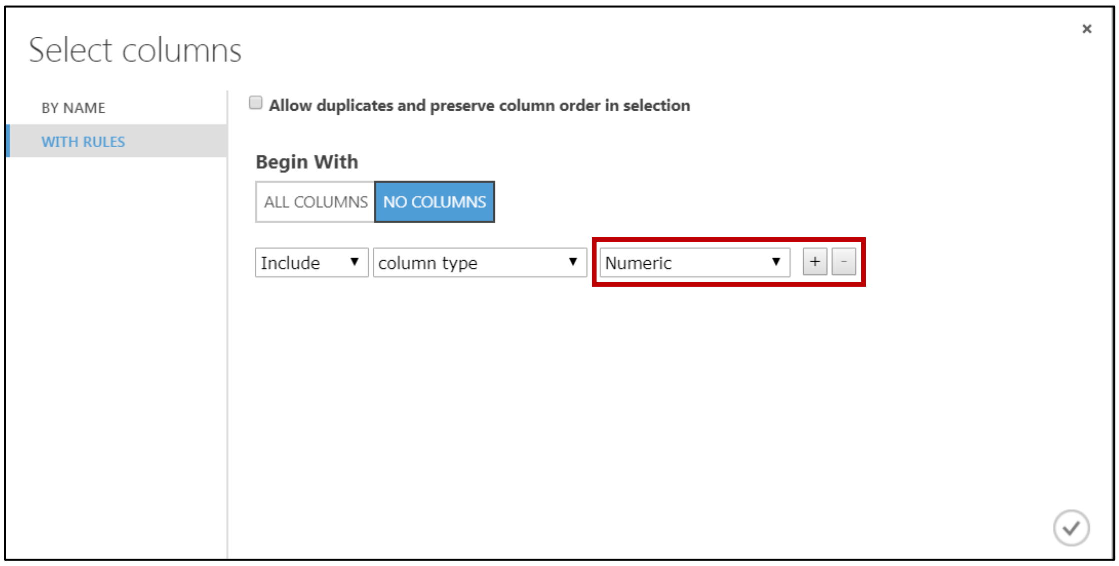
Education
Histogram

## 3.1.3. Normalize data

1. Add a **Normalize Data** module to your experiment and connect the output from the **Edit Metadata** module to the input of the **Normalize Data** module:



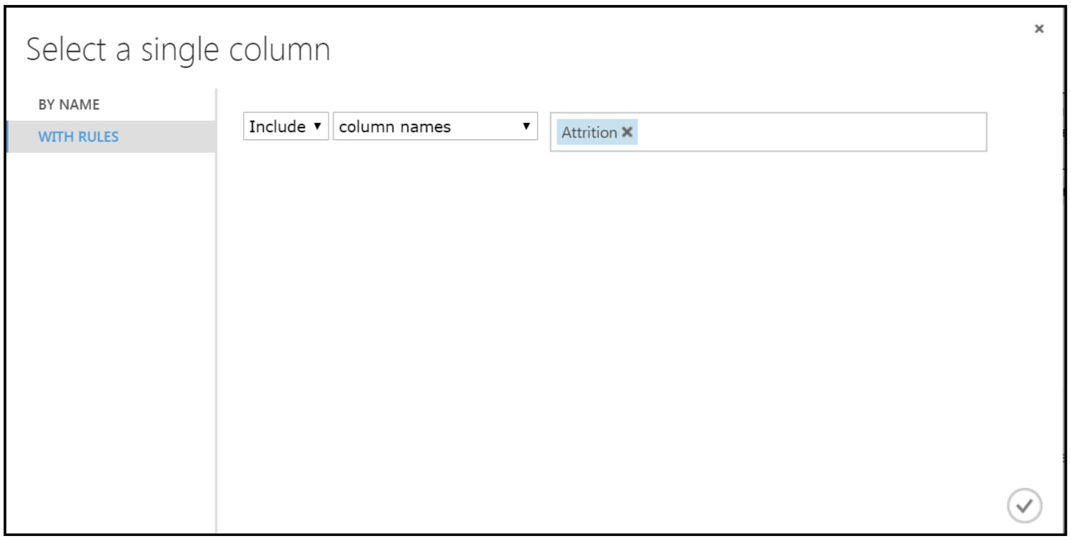
2. With **Normalize Data** selected, click on **Launch column selector** in the properties pane to the right.
3. In the column selector, choose to select columns **WITH RULES** and include all **Numeric** columns:



## 4. Building a Classification Model

1. Add a **Split Data** module to the experiment, connect the output of the **Normalize Data** module to its input port.

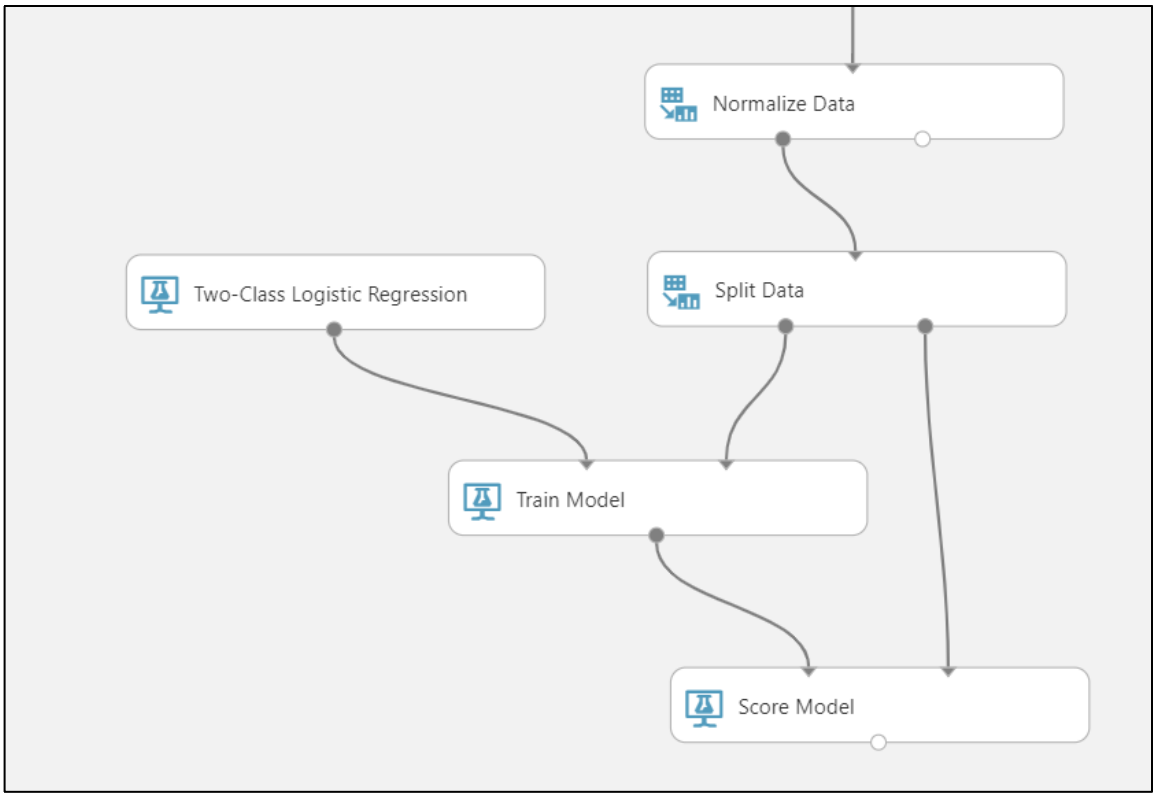
2. Set the **Properties** of the **Split Data** module as follows:
   - **Splitting mode:** Split Rows
   - **Fraction of rows in the first output:** 0.6
   - **Randomized split:** Checked
   - **Random seed:** 123
   - **Stratified split:** False
3. Add a **Two Class Logistic Regression** module to the experiment and set its properties as follows:
   - **Create trainer mode:** Single Parameter
   - **Optimization tolerance:** 1E-07
   - **L1 regularization weight:** 0.001
   - **L2 regularization weight:** 0.001
   - **Memory size for L-BFGS:** 20
   - **Random number seed:** 1234
   - **Allow unknown categorical levels:** checked
4. Add a **Train Model** module to the experiment, connect the **Untrained Model** output port of the **Two Class Logistic Regression** module to the **Untrained Model** (left) input port, and connect the **Results dataset1** (left) output port of the **Split Data** module to the **Dataset** (right) input port of the **Train model** module. Then configure its properties to select the **Attrition** column as the label column.


Select a single column

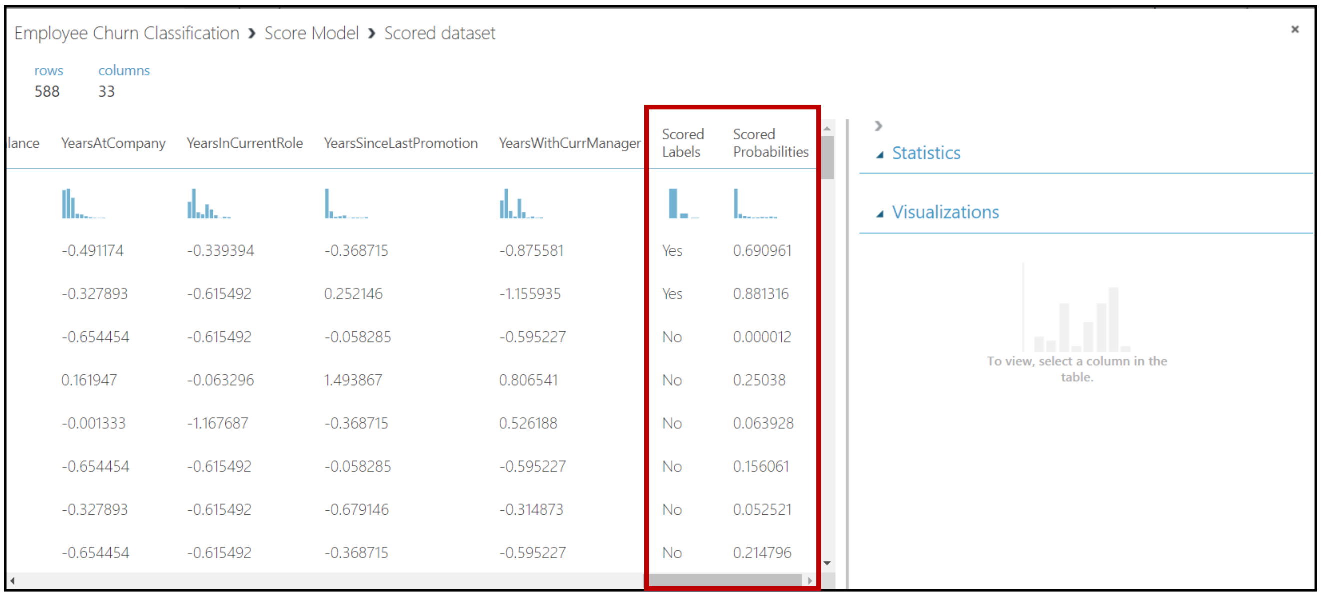
Include ▾ column names ▾ Attrition ✕

5. Add a **Score Model** module to the experiment. Then connect the output port of the of the **Train Model** module to its **Trained Model** (left) input port, and connect the **Results dataset2** (right) output port of the **Split Data** module to its **Dataset** (right) input port.

6. Verify that the bottom half of your experiment looks like this:



7. Save and run the experiment
8. When the experiment has finished running, visualize the output of the **Score Model** module and view the **Scored Labels** column. Note that this contains the values predicted by the model – the actual values from the test data are in the **Attrition** column. Additionally a **Scored Probabilities** column is generated. This is the true output of the classification model, a score indicating the probability that the employee either left or did not leave the company. The threshold is set to 0.5. This means that any score above 0.5 would mean the employee is likely to leave and thus has the scored label **"Yes"** where as any score below 0.5 would indicate that an employee is unlikely to leave the company and thus has a the scored label **"No"**.



Employee Churn Classification › Score Model › Scored dataset

rows 588 columns 33

| lance | YearsAtCompany | YearsInCurrentRole | YearsSinceLastPromotion | YearsWithCurrManager | Scored Labels | Scored Probabilities |
|---|---|---|---|---|---|---|
| | -0.491174 | -0.339394 | -0.368715 | -0.875581 | Yes | 0.690961 |
| | -0.327893 | -0.615492 | 0.252146 | -1.155935 | Yes | 0.881316 |
| | -0.654454 | -0.615492 | -0.058285 | -0.595227 | No | 0.000012 |
| | 0.161947 | -0.063296 | 1.493867 | 0.806541 | No | 0.25038 |
| | -0.001333 | -1.167687 | -0.368715 | 0.526188 | No | 0.063928 |
| | -0.654454 | -0.615492 | -0.058285 | -0.595227 | No | 0.156061 |
| | -0.327893 | -0.615492 | -0.679146 | -0.314873 | No | 0.052521 |
| | -0.654454 | -0.615492 | -0.368715 | -0.595227 | No | 0.214796 |

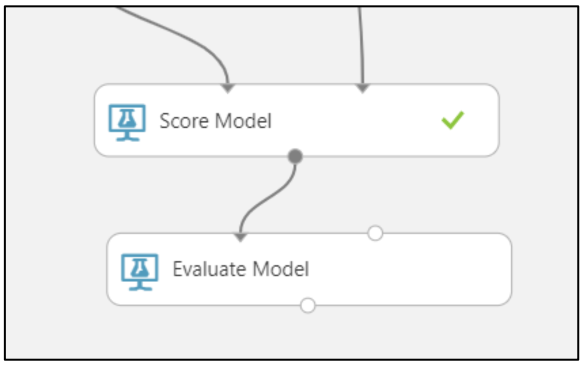
Statistics

Visualizations

To view, select a column in the table.

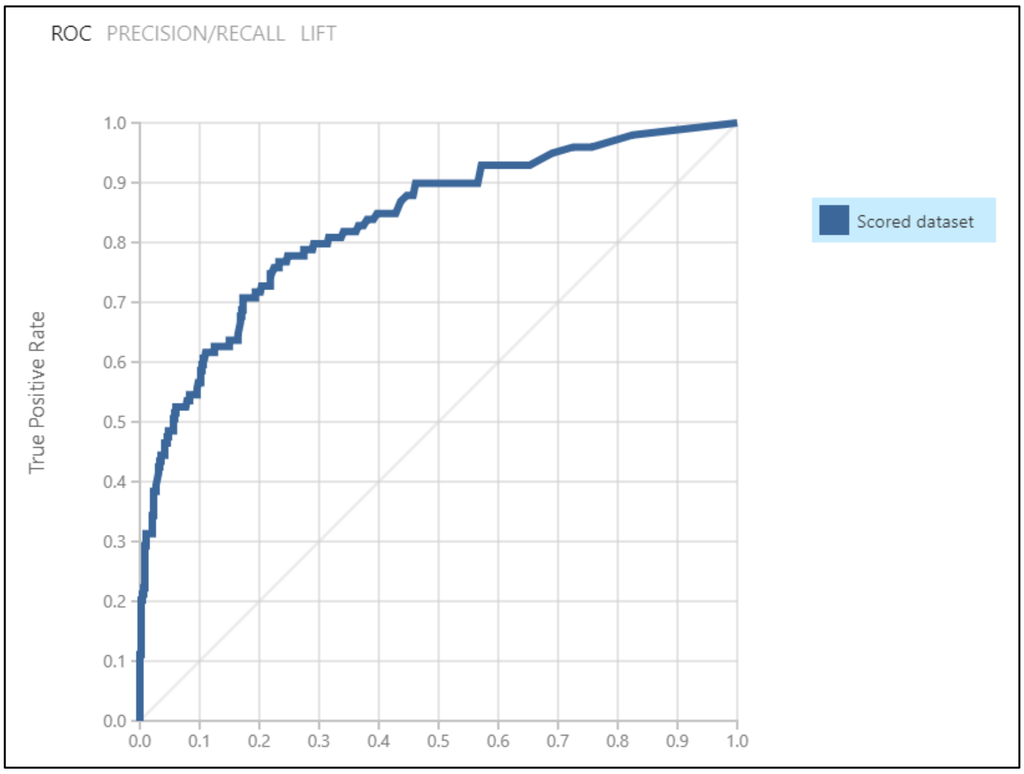
## 4.1. Evaluate the Model

1. Add an **Evaluate Model** module to the experiment. Then connect the output port of the **Score Model** module to the **Scored dataset** (left) input port of the **Evaluate Model** module.



2. Save and run the experiment.
3. When the experiment has finished, visualize the **Evaluation Result** port of the **Evaluate Model** module and scroll down to find the ROC curve as shown below.

   Remember that the goal of this classification problem is to prevent employees from leaving the company. Correctly identifying which employees are likely to leave allows the company to act and attempt to retain the employee, and as a result reduce their costs. Employees with a false negative will not be properly classified as unlikely to leave the company, and will in reality leave the company. Therefore, the recall statistic is important in this problem since maximizing recall minimizes the number of false negative scores.



Examine the ROC curve. Notice that the bold blue line is well above the diagonal grey line, indicating the model is performing better that random guessing.

4. Next, examine the statistics for the model evaluation, noting the area under the curve (AUC), the number of true positives, false positives, false negatives, and true negatives in the confusion matrix, and the accuracy, recall, precision, and F1 score.

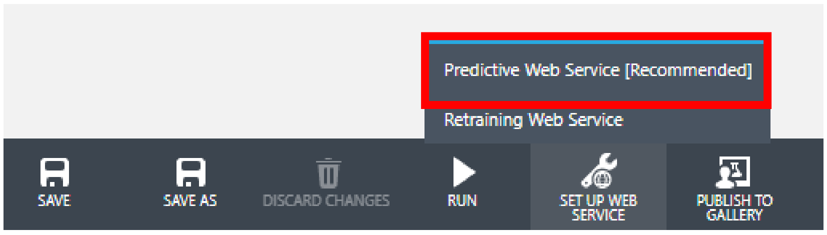
   Notice the following:

   - Correctly classified values (**True Positive + True Negative)** values should outnumber the errors (**False Negative + False Positive**).
   - An **Accuracy** of greater than 0.5 indicates that the scores are correct more often than not.
   - There are nearly equal numbers of **True Positive** and **False Negative** scores.
   - The relatively low **Recall** value results from a high number of **False Negative** values.
   - The AUC metric quantifies the area under the curve – this should be greater than 0.5 in a model that performs better than random guessing.
5. Close the evaluation results.

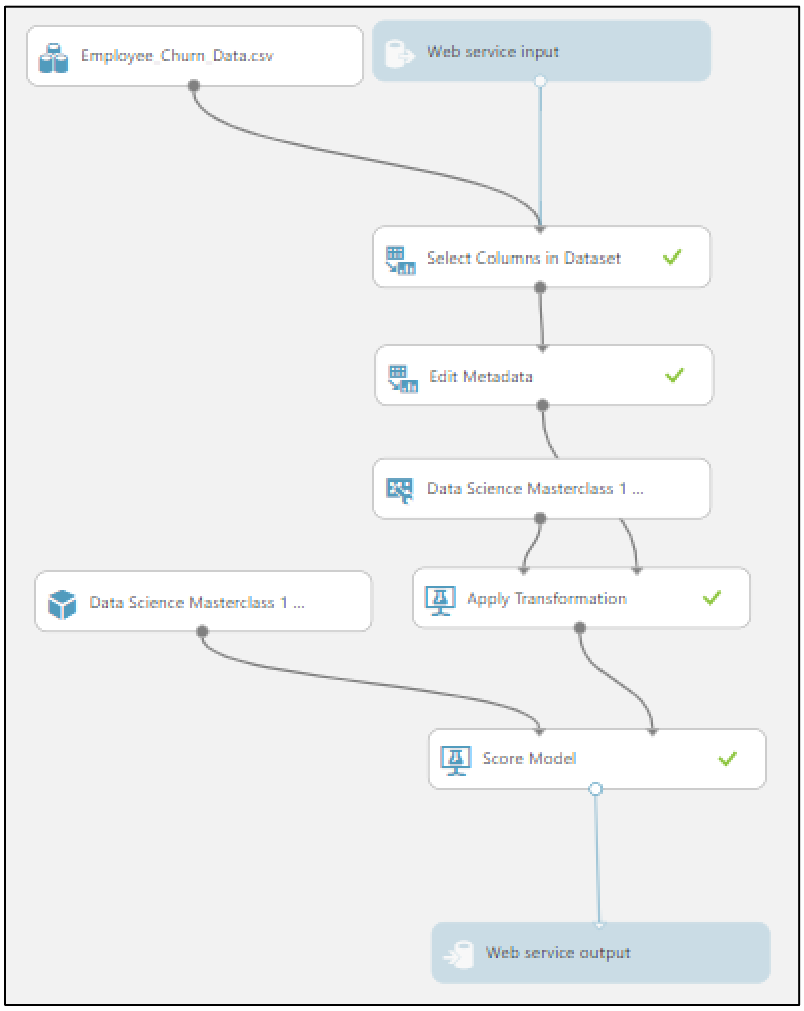
## 5. Deploying the Model

Once you have a predictive model built, and tested, the next step is to the deploy the model into an environment where end users can interact with the model and consume the results. To we will deploy a model to excel online, where users can enter the required employee features into a table and get back a predicted result, indicating whether this newly specified employee is likely to leave or stay.

To begin deploying of your model make sure you have saved and run your model. Click on **SET UP WEB SERVICE** and select **Predictive Web Service [Recommended].** Once your predictive experiment is created.



After the experiment has run, the Predictive Experiment should look like the following:



Delete the link between the **Score Model** module and the Web service output. From the menu pane, find and insert the **Select columns in dataset** module and drag it into the experiment. Connect the output from the **Score Model** module to the input of the **Select columns in dataset**. Connect the output of the **Select columns in dataset** to the input of the **Web service out** module.

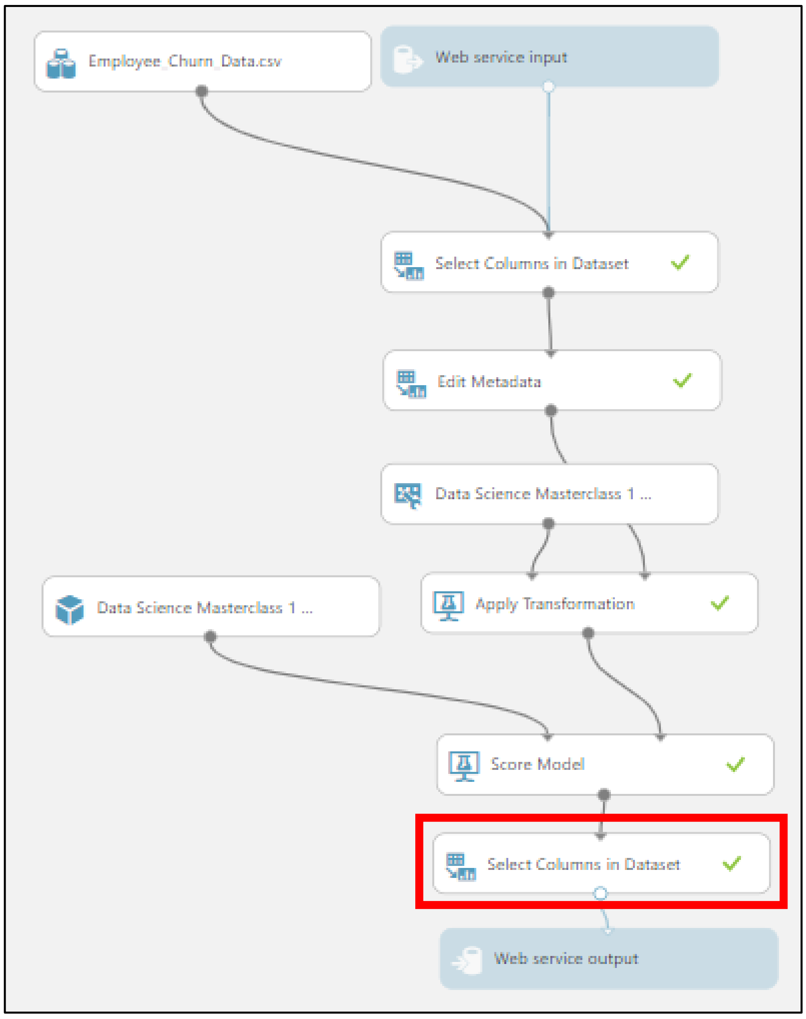
Launch the **Column selector** in the properties pane.

Select the following two columns:

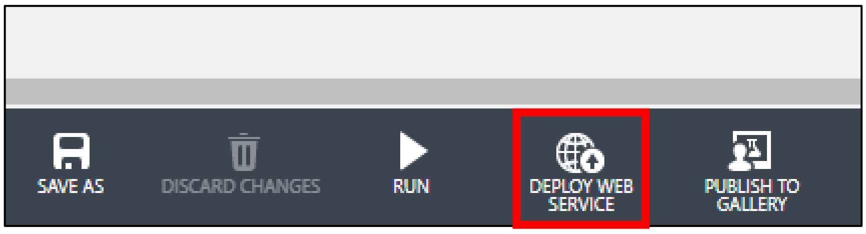
- Scored Labels
- Scored Probabilities

Save and run the experiment.

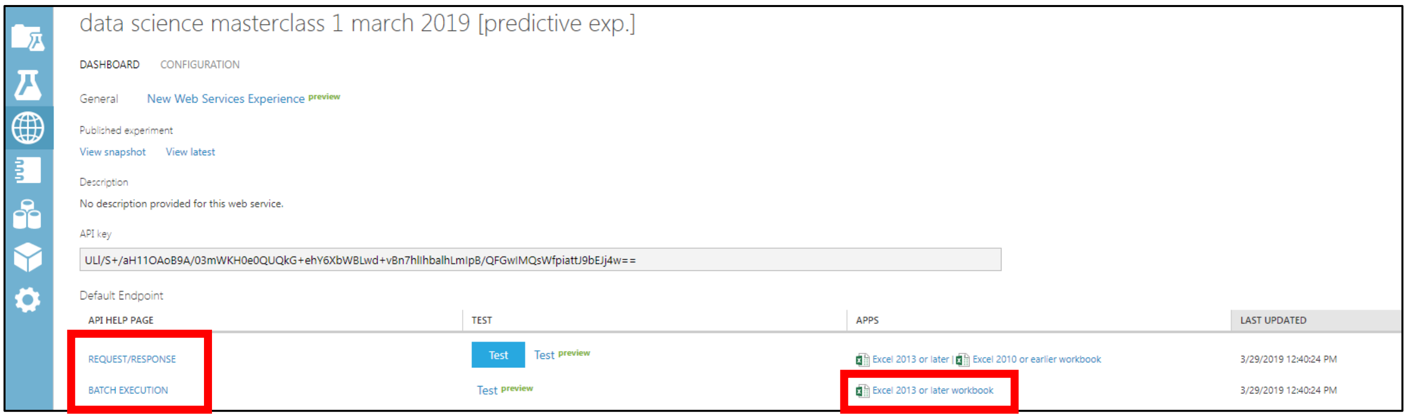
The Predictive Experiment should look like the following:


Employee_Churn_Data.csv
Web service input
Select Columns in Dataset
Edit Metadata
Data Science Masterclass 1 ...
Data Science Masterclass 1 ...
Apply Transformation
Score Model
Select Columns in Dataset
Web service output

On the bottom bar, select **DEPLOY WEB SERVICE**



You will be directed to a new web page.



data science masterclass 1 march 2019 [predictive exp.]

DASHBOARD CONFIGURATION

General New Web Services Experience preview

Published experiment

View snapshot View latest

Description

No description provided for this web service.

API key

ULI/S+/aH11OAoB9A/03mWKH0e0QUQkG+ehY6XbWBLwd+vBn7hlIhbalhLmIpB/QFGwIMQsWfpiattJ9bEJj4w==

Default Endpoint

| API HELP PAGE | TEST | APPS | LAST UPDATED |
|---|---|---|---|
| REQUEST/RESPONSE | Test Test preview | Excel 2013 or later \| Excel 2010 or earlier workbook | 3/29/2019 12:40:24 PM |
| BATCH EXECUTION | Test preview | Excel 2013 or later workbook | 3/29/2019 12:40:24 PM |

There are two options: **REQUEST/RESPONSE** and **BATCH EXECUTION.**

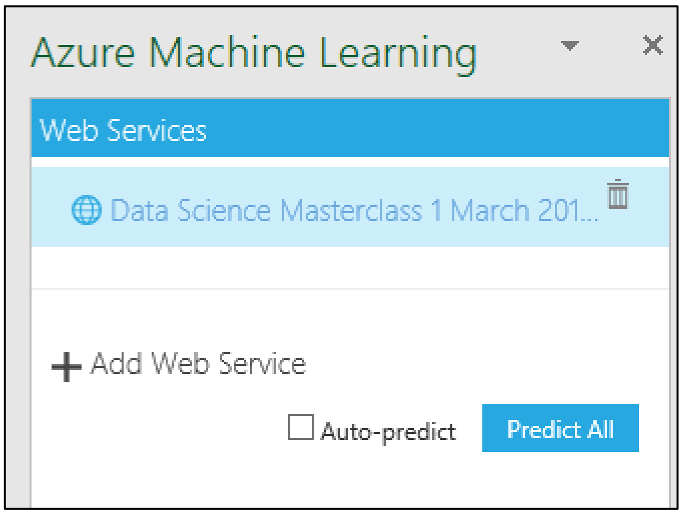
REQUEST/RESPONSE is used for a single input/output call.

BATCH EXECUTION is used to send a batch of data for which you wish to make the predictions, and you will receive a batch of answers.

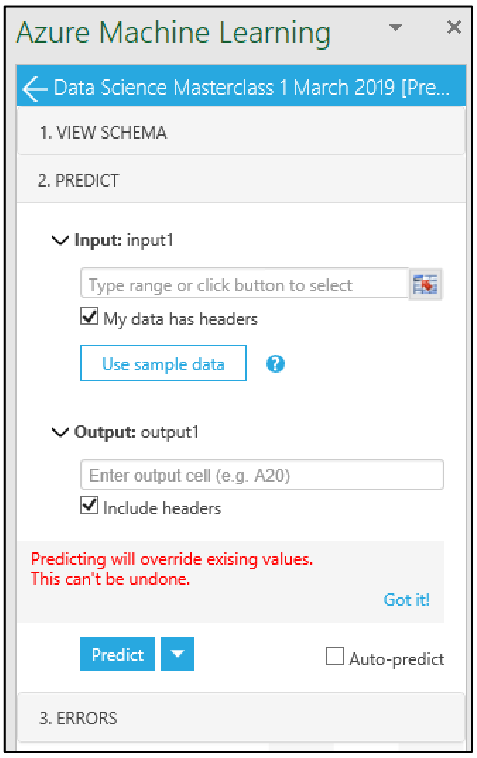
For the **BATCH EXECUTION**, select **Excel 2013 or later workbook**.

An Excel file will be downloaded. Open this file. You will see a warning about the file being in protected view. Select **Enable Editing**.

The Excel file will load and you will see a pane on the right-hand side of the screen. If this is the first time you are opening the Azure Machine Learning add-in in Excel, you will be prompted to trust this add-in. Once you have confirmed that this is a trusted add-in, you will see the following:



The text highlighted in blue is the name of the machine learning service you have created. Select this. The pane will now look like the following:


Azure Machine Learning

Data Science Masterclass 1 March 2019 [Pre...

1. VIEW SCHEMA

2. PREDICT

**Input:** input1

Type range or click button to select

☑ My data has headers

Use sample data

**Output:** output1

Enter output cell (e.g. A20)

☑ Include headers

Predicting will override exising values.
This can't be undone.

Got it!

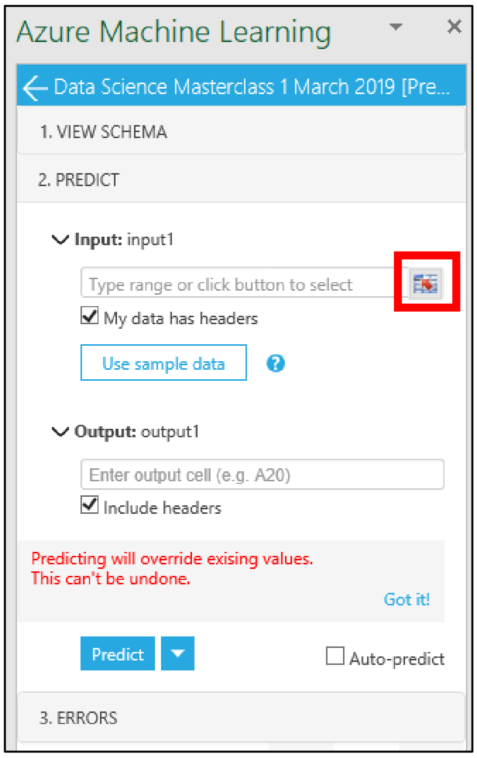
Predict

☐ Auto-predict

3. ERRORS

In order to test the request/response of the service, select the **Use sample data** button.

A table of sample data will appear in the Excel sheet. You now want to select a data range to send to the machine learning service for it to make predictions. Select the following range in the Excel sheet: A1:AI6. With this range selected, click the following button: Click OK to confirm your selection.


Azure Machine Learning

Data Science Masterclass 1 March 2019 [Pre...

1. VIEW SCHEMA

2. PREDICT

**Input:** input1

Type range or click button to select

☑ My data has headers

Use sample data

**Output:** output1

Enter output cell (e.g. A20)

☑ Include headers

Predicting will override exising values.
This can't be undone.

Got it!

Predict

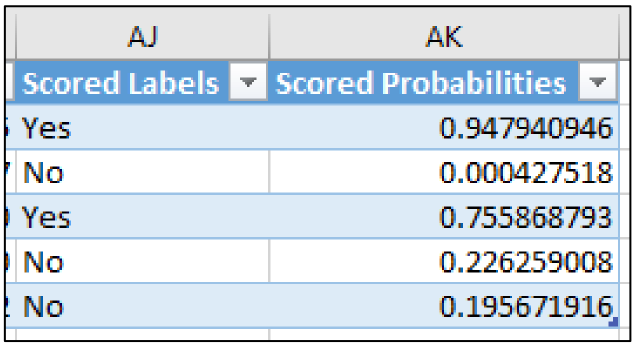
☐ Auto-predict

3. ERRORS

In the **Output** box, type: AJ1. This is where the predicted results will appear.

Click **Predict**.

The experiment will run for a few seconds, and then two new columns will appear. **Scored Labels** and **Scored Probabilities.**



| AJ | AK |
|---|---|
| Scored Labels | Scored Probabilities |
| Yes | 0.947940946 |
| No | 0.000427518 |
| Yes | 0.755868793 |
| No | 0.226259008 |
| No | 0.195671916 |

## 6. Summary

In this exercise, you have created a classification model using the **Two-Class Logistic Regression** algorithm. An initial evaluation of the model seems to indicate that it provides better results than random guessing, but more evaluation is needed, and it may be that the model could be improved.

## Johannesburg

Altron Karabina Building, Design Quarter District; Leslie Ave East, Fourways, Johannesburg, South Africa, 2191

**T**: +27 11 463 8155

## Cape Town

8th Floor The Edge Building, 3 Howick Close, Tyger Falls, Bellville, Cape Town, South Africa, 7925

**T:** +27 21 001 1044

**e**: connect@altronkarabina.com

**w:** www.altronkarabina.com

ALTRON | KARABINA

Gold Microsoft Partner

Microsoft