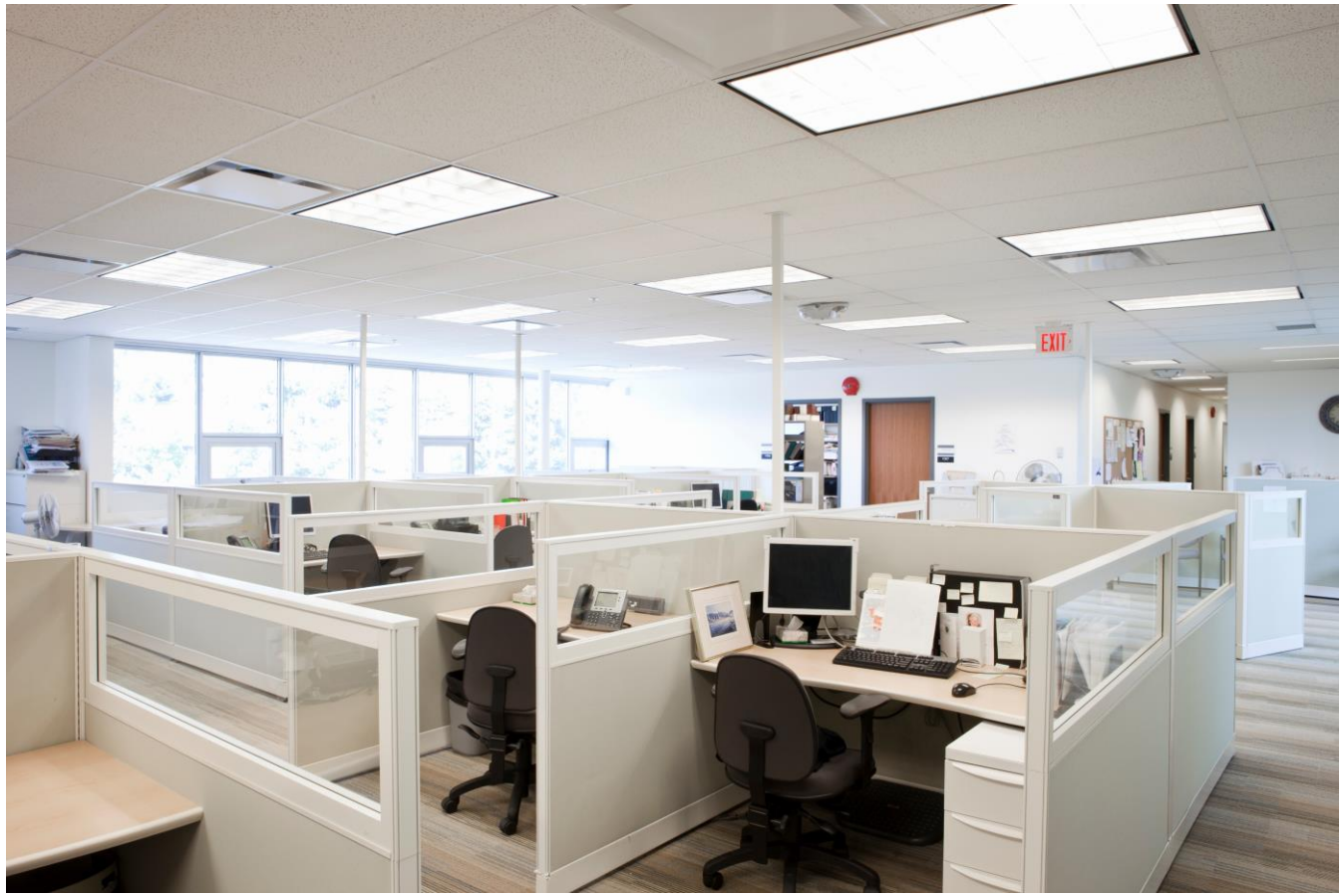
Office Room Occupancy

DS 260

Wes McNall

Introduction



- Offices can be a busy work environment. You may not notice where employees are at any given time.
- Given information about a room, can you determine if that room is occupied?
- This is a dataset courtesy of Luis M. Candanedo and Veronique Feldheim. They kept track of these variable within an office:
  - Temperature
  - Humidity
  - Light
  - CO2
  - Humidity Ratio
  - Occupancy
- Is it possible, given these other variables, that we can predict whether or not someone was in a given room?

Variables

- Temperature
  - Ranged from 20°-24.4° Celsius.
  - Converts to approximately 68° – 76° Fahrenheit.
  - No Outliers.
- Humidity
  - 22 – 31 grams of water vapor per cubic meter of air.
  - No Outliers.
- Light
  - Measured in Lux, the SI unit of illuminance, equal to one lumen per square meter.
    - Sunlight is 10,000 Lux.
    - Full Daylight is 1,000 Lux.
    - Normal Office work is recommended 500 Lux.
  - 0 when Lights are off.
  - Never any occupancy when lights are off.
  - All values > 750 are an outlier.
  - 1,546 is the maximum Lux.
    - Detailed drawing for artists is ~ 1,500.
  - https://www.noao.edu/education/QLTkit/ACTIVITY_Documents/Safety/LightLevels_outdoor+indoor.pdf
- CO2
  - Ranged from ~400-2000
  - 600-800 is Acceptable indoor air quality
  - 1,000 is tolerable indoor air quality
  - > 6,000 is a sign for concern
  - Maximum is ~2,000
  - https://www.vaisala.com/sites/default/files/documents/CEN-TIA-Parameter-How-to-measure-CO2-Application-note-B211228EN-A.pdf
- Humidity Ratio
  - Minimum: 0.002674
  - Maximum: 0.006476
  - No outliers.
- id
  - ID isn't just a row identifier in this dataset, it acts more as a sense of time.
  - No Outliers.
- Occupancy
  - Only has values of 0 and 1.
    - 0: No occupants.
    - 1: One or more Occupants.

Data Preparation

- Do we need to clean the data? How do we know if it needs it?

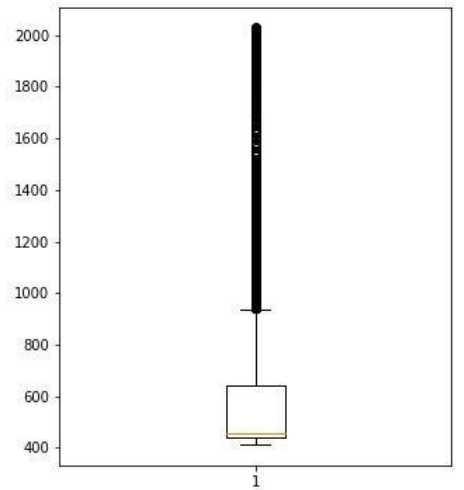
```
In [64]: train.describe()
```

Out[64]:

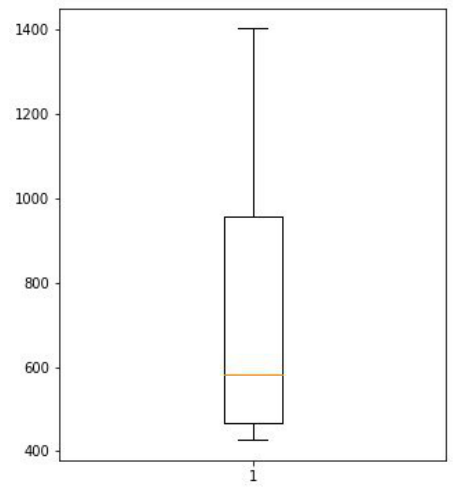
	id	Temperature	Humidity	Light	CO2	HumidityRatio	Occupancy
count	8143.000000	8143.000000	8143.000000	8143.000000	8143.000000	8143.000000	8143.000000
mean	4072.000000	20.619084	25.731507	119.519375	606.546243	0.003863	0.212330
std	2350.825954	1.016916	5.531211	194.755805	314.320877	0.000852	0.408982
min	1.000000	19.000000	16.745000	0.000000	412.750000	0.002674	0.000000
25%	2036.500000	19.700000	20.200000	0.000000	439.000000	0.003078	0.000000
50%	4072.000000	20.390000	26.222500	0.000000	453.500000	0.003801	0.000000
75%	6107.500000	21.390000	30.533333	256.375000	638.833333	0.004352	0.000000
max	8143.000000	23.180000	39.117500	1546.333333	2028.500000	0.006476	1.000000

- We can see here that every column has the max count of 8143, that means that there are no missing values to contend with.
- Temperature, Humidity, and HumidityRatio have no outliers

```
plt.figure(figsize=(5, 6))
plt.boxplot(train.CO2)
plt.show()
```



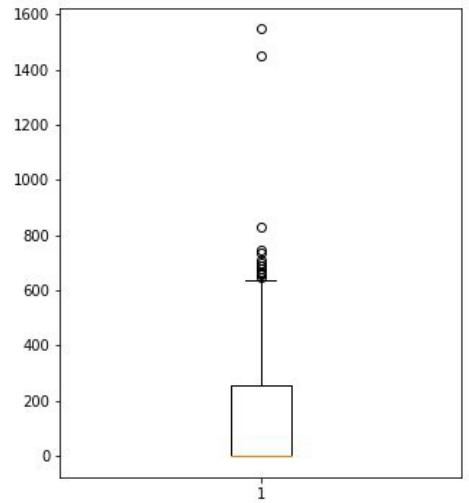

```
plt.figure(figsize=(5, 6))
plt.boxplot(test.CO2)
plt.show()
```



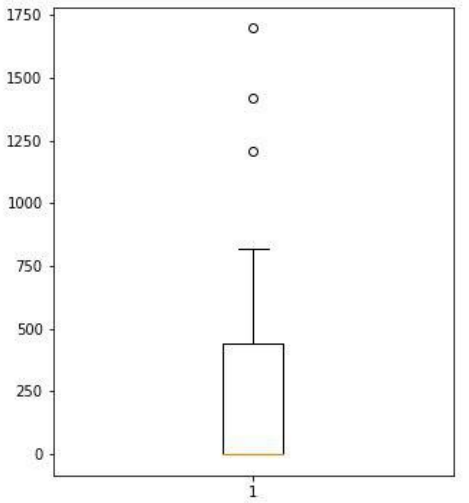
- While there are outliers within the training set for CO2, there are no outliers for the testing set.
  - I decided not to cleanse the training dataset of outliers. The range where it's values are concentrated, < 600, is where over 50% of the values are for the testing set, so it will still have a strong prediction for the testing set

Data Preparation

```
plt.figure(figsize=(5, 6))
plt.boxplot(train.Light)
plt.show()
```



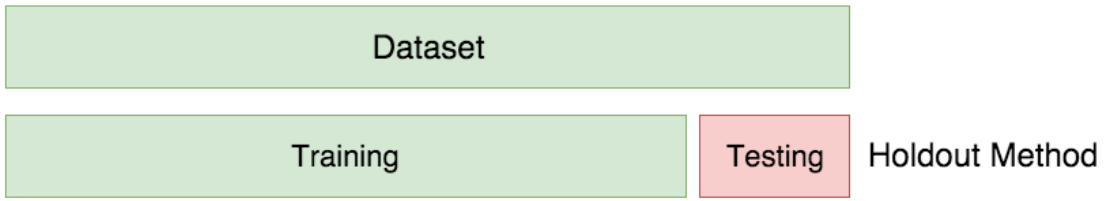

```
[20]: plt.figure(figsize=(5, 6))
plt.boxplot(test.Light)
plt.show()
```

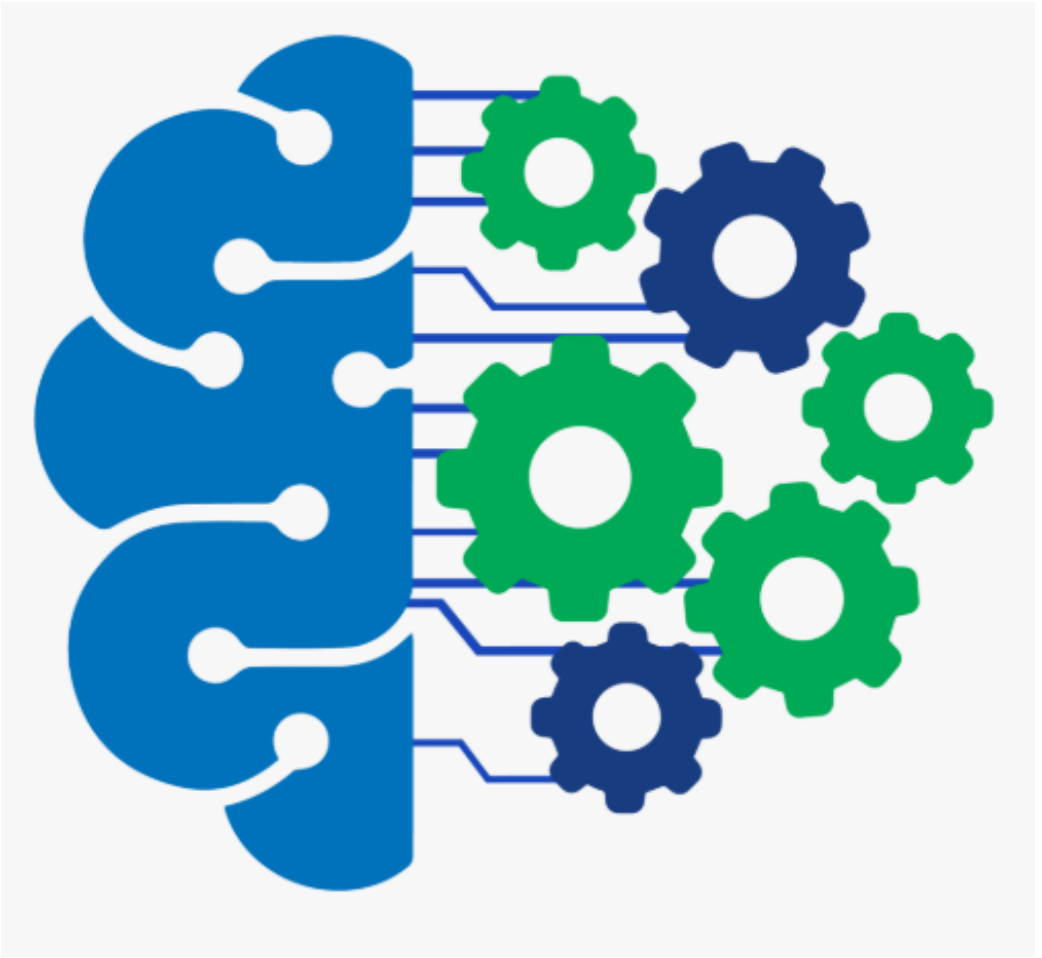


- Light also contains a few outliers, but we can see with the colored line at the bottom of the box in these plots that the majority of values is 0 which will come into play later. I still feel no need to clean up the outliers as there are so few in a dataset of over 8,000 points.

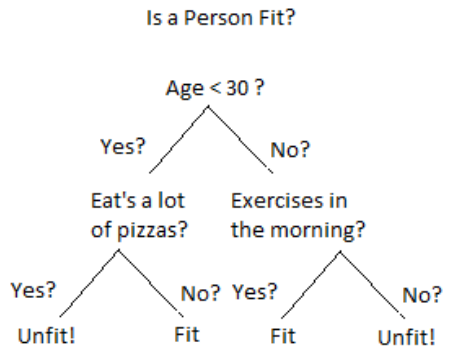
Machine Learning Models

- As we are going to be predicting Occupancy, a variable with only two possible values, 0 (non-occupied) or 1 (occupied), this is a Classification problem. A fancy way of saying we're trying to predict a category.
- We are going to build several different models for this task.
  - Decision Tree
  - Random Forest
  - K-Nearest Neighbors
- We are also going to be using a training set and a testing set.
  - The figure below visualizes this process, of holding out a portion of the dataset to be able to have a way to test our model on new and incoming data.



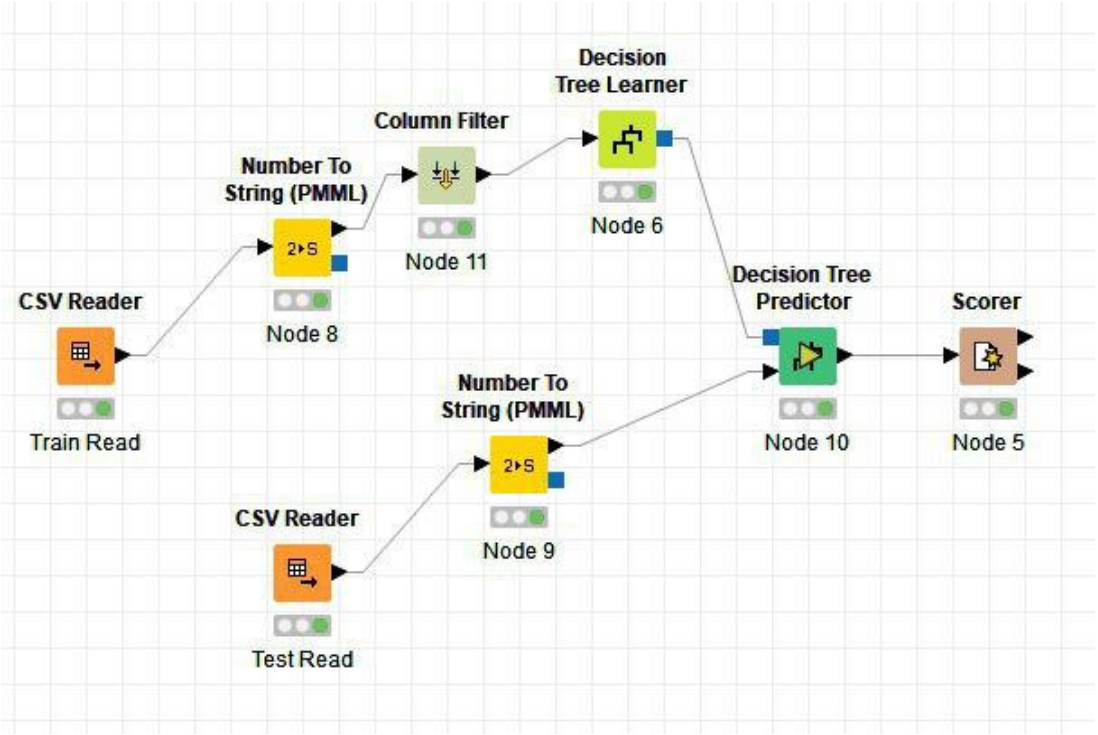


Decision Tree(s)



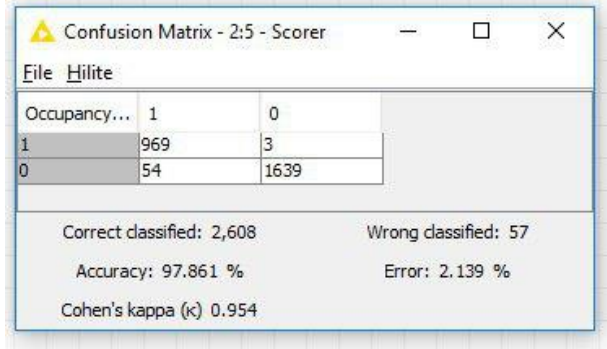
Winner!

- Above is a simple Decision Tree, we can create one with our dataset that will help us decide if a room is Occupied.



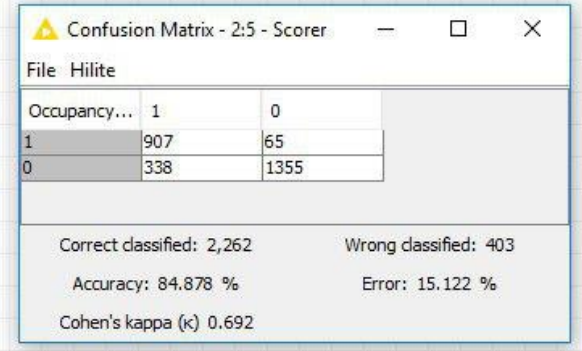
Decision Tree 1

- Using ONLY Light.
- 1,000 min number records per node.
- 5 threads.



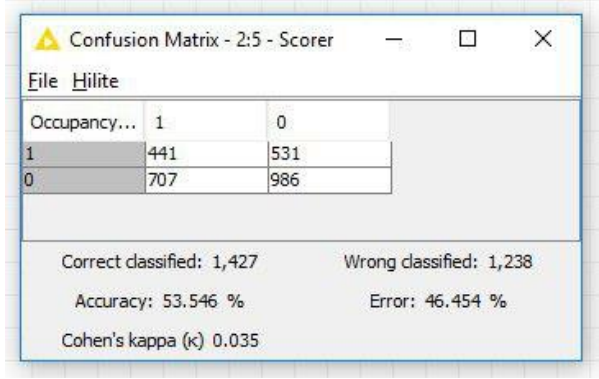
Decision Tree 2

- All variables BUT Light.
- 1,000 min number records per node.
- 5 threads.

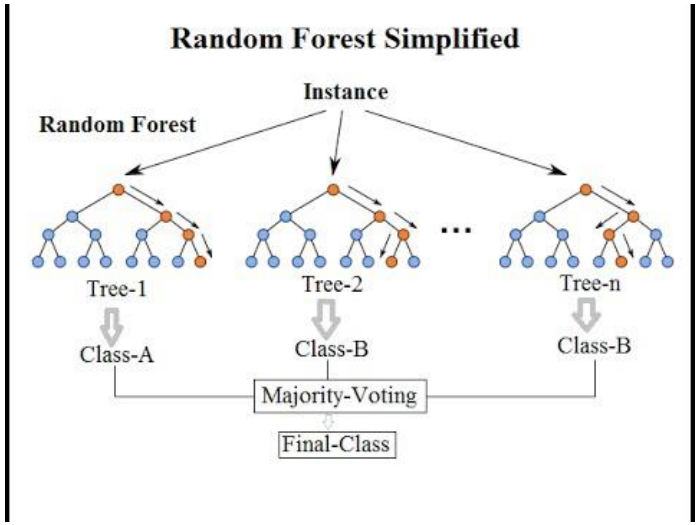


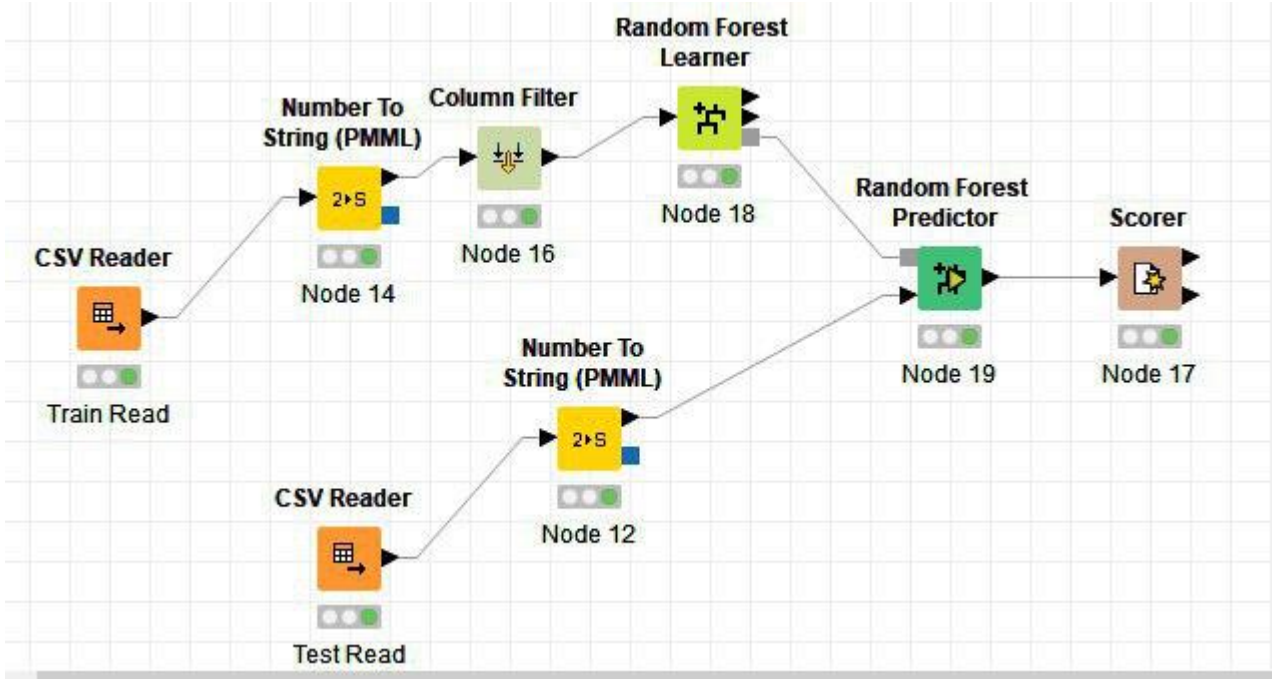
Decision Tree 3

- Using ONLY id.
- 500 min number records per node.
- 3 threads.

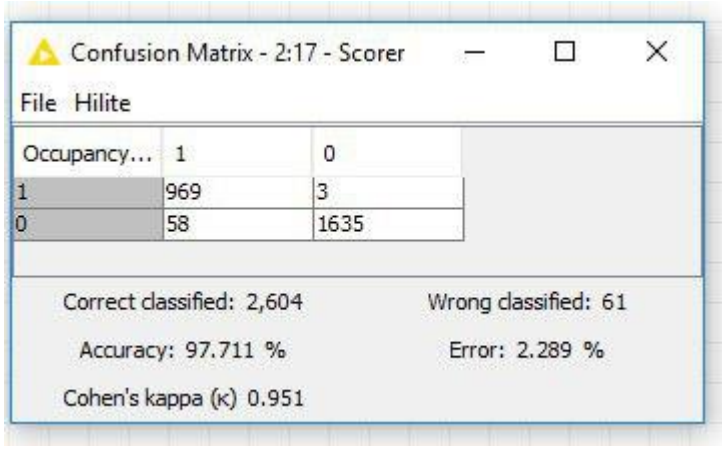


Random Forest





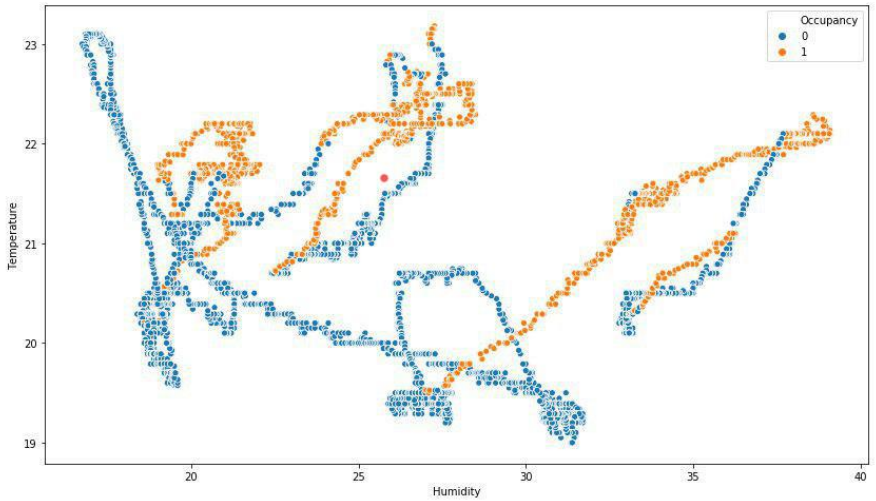
- Above is a simple Random Forest. The basic idea is to use multiple Decision Trees as a way to get towards a stronger model.
- To the right we have a few figures from KNIME, I went ahead and used all the possible columns.

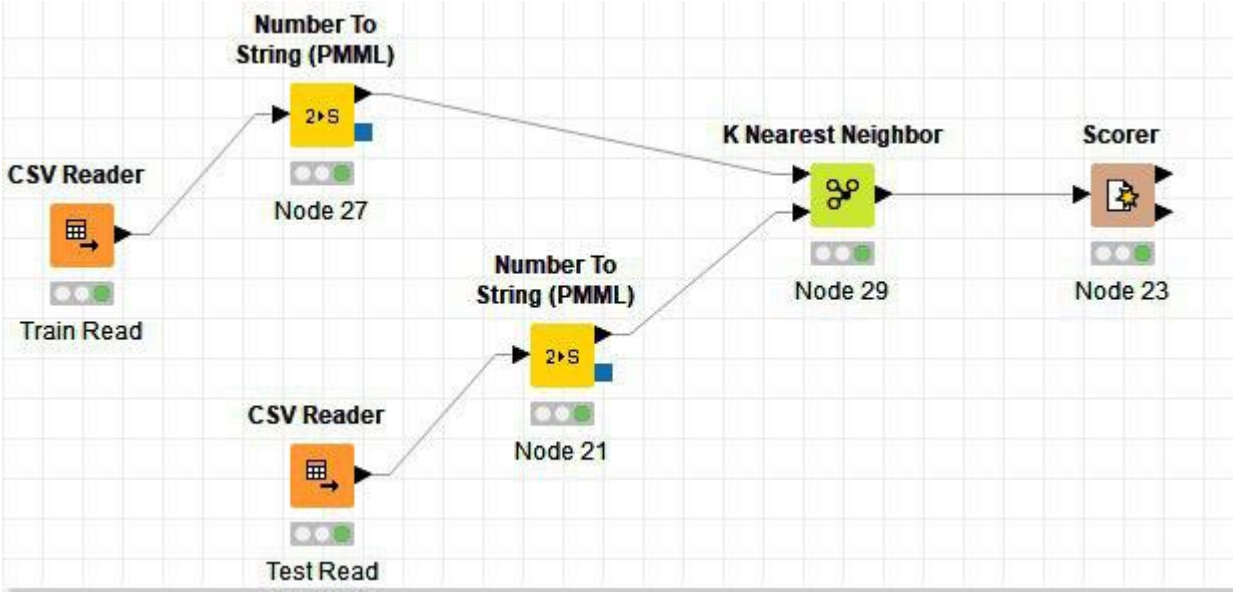


Decision Tree using only Light is still winning

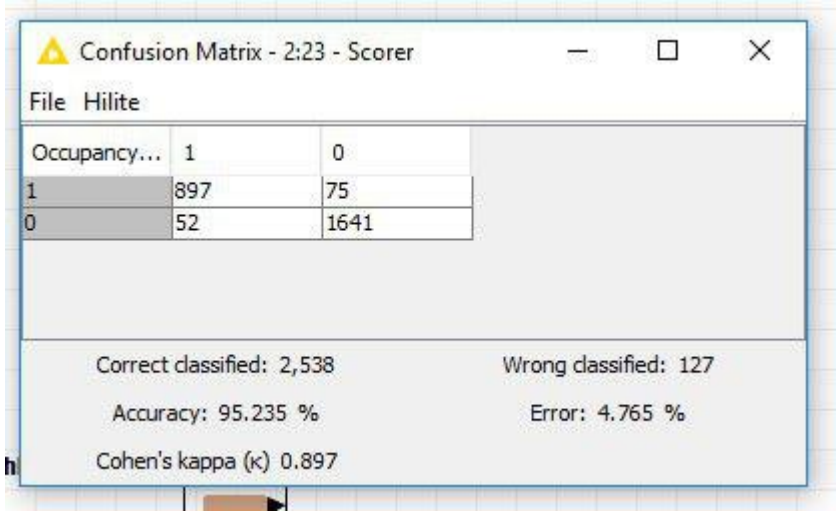
K-Nearest Neighbors

```
plt.figure(figsize=(global_width, global_height))
sns.scatterplot(data = train,
                x = 'Humidity',
                y = 'Temperature',
                hue = 'Occupancy')
ax = plt.gca()
#ax.set_title("Location of Biking Stations and their Popularity")
plt.show()
```



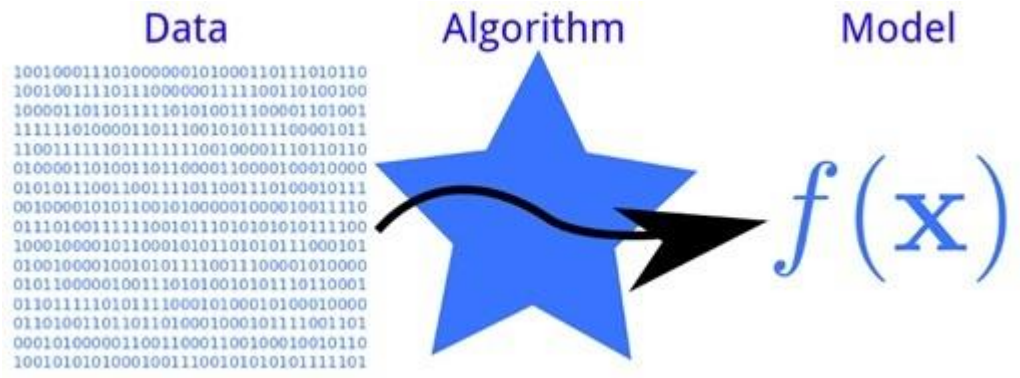


- Above is a graph from this dataset. It's a scatterplot of Humidity / Temperature, with Occupancy set to color. I put a red dot as a new datapoint. A kNN model of 3, would look at it's 3 closest neighbors, all of which would be blue, which would identify the Occupancy of the new point to be 0, not-occupied.
- To the right is a kNN model within KNIME, k = 10.



Decision Tree using only Light remains the winner

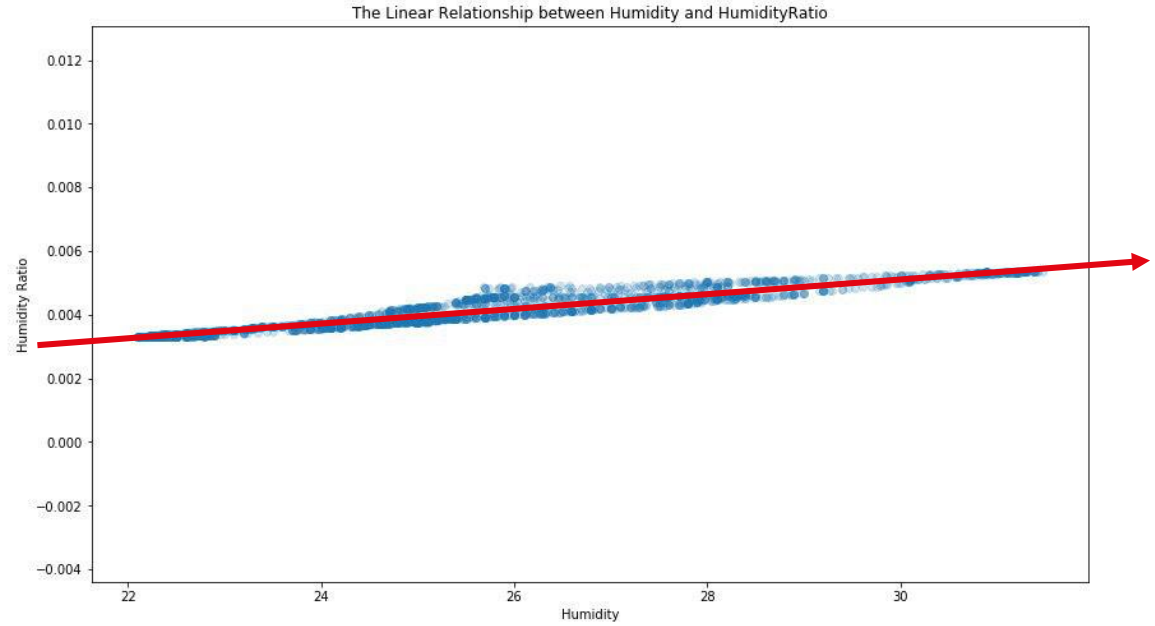
Other Machine Learning Models



- We have focused on **Supervised Learning** models as we were predicting a value, but **Unsupervised Learning** models could be used as well.
  - Association and Clustering are types of Unsupervised Learning.

- There are other Machine Learning models that we didn't touch on this presentation that could be applied towards this problem.
- Logistic Regression
  - Because we are dealing with two choices, a 0 and a 1, we can attempt to use a binary classification method like this.
- Neural Networks
  - Are very popular because of their ability to make quality predictions, but is out of the scope of this course.
- Linear Regression
  - Humidity / HumidityRatio have a high Correlation. There's a graph of it to the right, and we can see a clear red line that would fit perfectly through the data. It would be possible to use Linear Regression to come up with the equation of that red line and predict and attempt to predict any missing values with that

```
plt.figure(figsize=(global_width, global_height))
plt.scatter(test.Humidity, test.HumidityRatio,
            alpha = 0.20)
plt.xlabel('Humidity')
plt.ylabel('Humidity Ratio')
plt.title('The Linear Relationship between Humidity and HumidityRatio')
plt.show()
```



Conclusion

- In this project we've tried to determine if it's possible, given variables about an office, whether or not we can determine if you can predict if an office is Occupied.
- We fit these models to that question:
  - Decision Tree
  - Random Forest
  - K-Nearest Neighbor
- If you had asked me before I started the project what I thought the winner would be, I would have said Random Forest.
- Sometimes simple is better, and with this project that was certainly the case. Using a simple Decision Tree using Light as a predictor for Occupancy was all that was needed for an amazing accuracy score of 97.861%.
- While there are other models that we didn't try, I would be hard-pressed to try something more complicated when something quick and simple did such a strong job of prediction for us.
- With this project I learned to:
  - Ask questions.
  - Question my assumptions.
  - Research variable units I didn't understand.

Thank You