Final Project

Camilla Sisemore

Data Mining (DS 260)

Introduction

One February, likely in 2015 (based on the publication date of their paper[i]), Luis M. Candanedo and Véronique Feldheim, both of Université de Mons in Mons, Belgium performed an experiment to see if they could determine if a room was occupied based on readings from sensors that measured temperature, humidity, carbon dioxide, and light. The readings were taken three to four times each minute and averaged to give one result per minute per feature. A picture was taken once per minute so that occupancy could be determined by the researchers. The dataset also includes humidity ratio. I will be attempting to determine the occupancy based on combinations of temperature, humidity, carbon dioxide, light, and the humidity ratio. I suspect that CO_2 will definitely be one of the indicators.

Data Preparation

I used `df.shape` to view the number of columns and rows in the dataset – it showed 8143 rows and 7 columns. I used `df.head()` and `df.tail()` to view a snapshot of the data. I used `df.isna().any()` to return the number of records in the dataset that contained null values - it showed that the Occupied column contained three null values, so I used `df.dropna(axis = 0, how = 'any', subset=['Occupied'], inplace=True)` to drop the three rows with null values.

To normalize the values in the 'Occupied' column (there were some 'Y' and 'N' sprinkled in amongst the 'yes' and 'no' values), I used `df['Occupied'] = df['Occupied'].str.replace(' ', '')` to remove extra white space. I then used `df['Occupied'] = df['Occupied'].replace('Y', 'yes', regex=True)` and `df['Occupied'] = df['Occupied'].replace('N', 'no', regex=True)` to clean up the 'Y' and 'N' values, along with `df['Occupied'] = df['Occupied'].replace('yeses', 'yes', regex=True)` to compensate for not being quite careful enough the first time, which I figured out after using `df['Occupied'].value_counts()`.

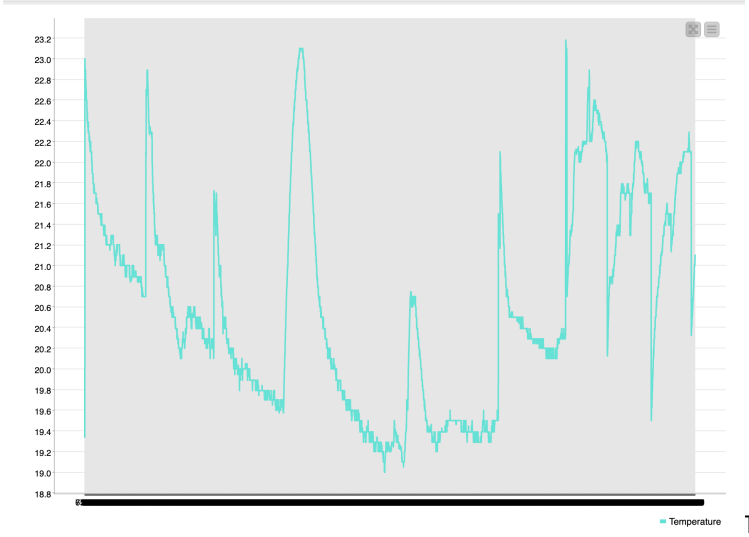
Training Set Exploration

To get the mean of the features, I used `df.mean()`. I used `df.median()` to return the median values.

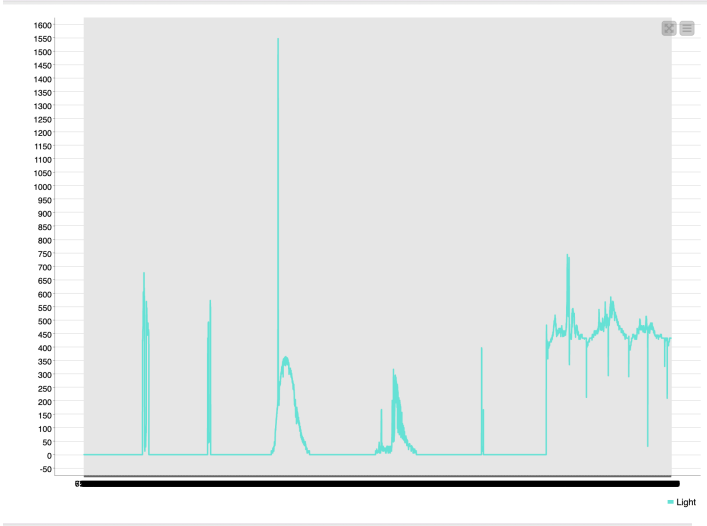
Feature	Mean	Median
Temperature	20.618656	20.390000
Humidity	25.731379	26.222500
Light	119.502981	0.000000
CO2	606.533668	453.500000
HumidityRatio	0.003862	0.003801

Below are some line plots made from the dataset – one for each feature, including 'Occupied', even though that is what I will be trying to predict. These were made using KNIME. I used the RowID for the X-axis, and the features on the Y-axis. I used the RowID, since the dataset did not contain any date/time values.

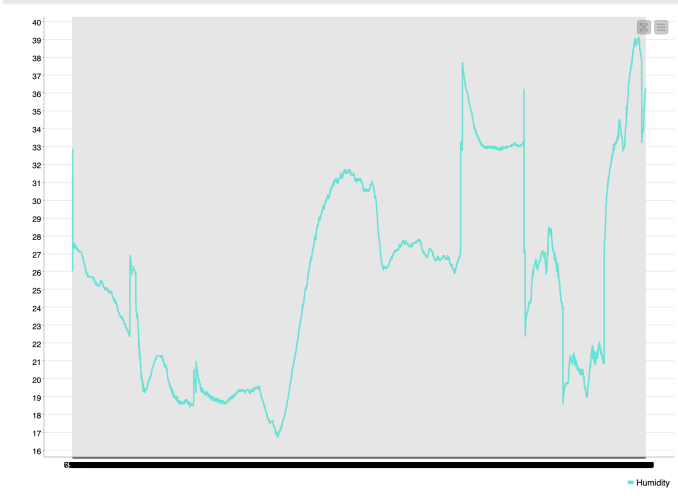
I see a correlation between temperature and humidity, and another between carbon dioxide and humidity ratio, plus a slighter one between light and temperature, so I will likely use these as my predictors. I don't want to use all five at once, because I think that might make the predictions less accurate.



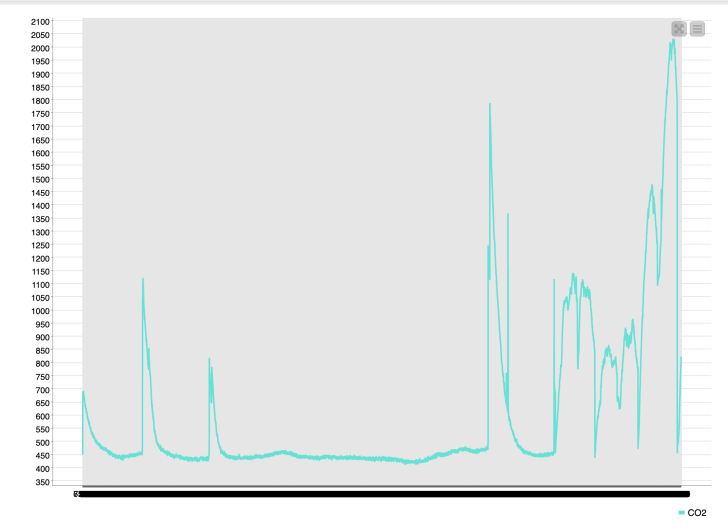
Temperature



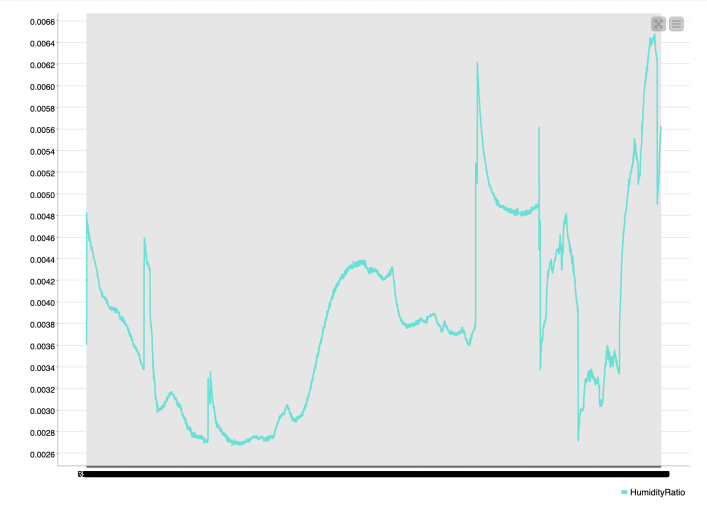
Light



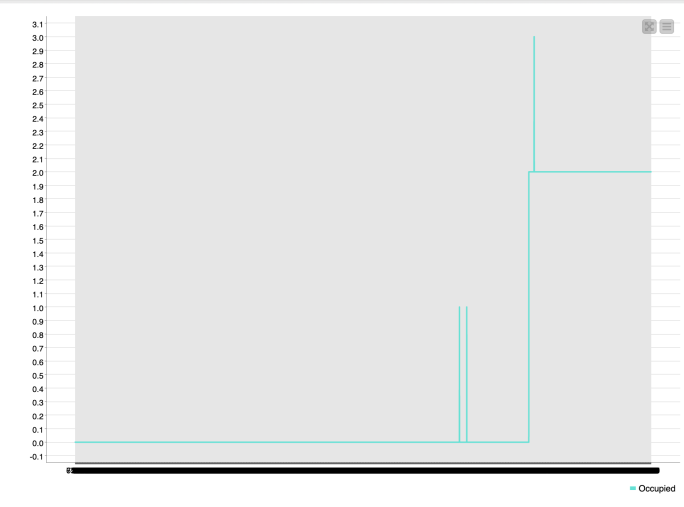
Humidity



Carbon Dioxide



Humidity Ratio



Occupied

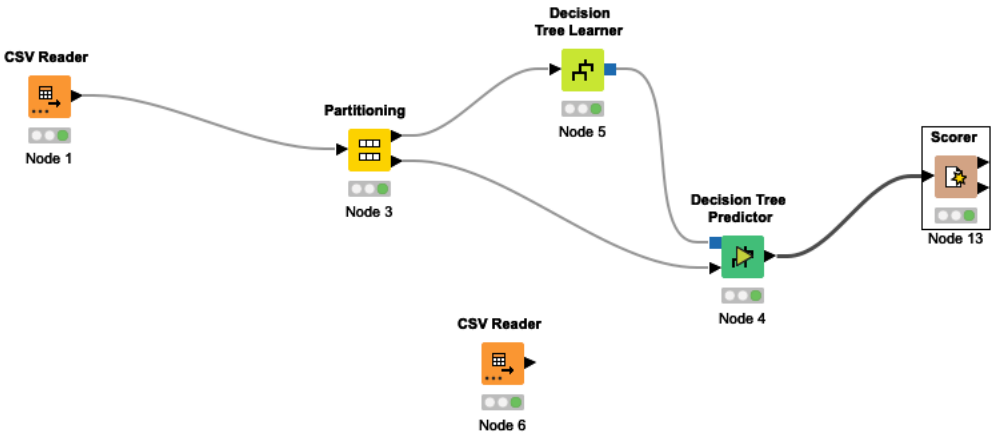
Predicting Occupancy

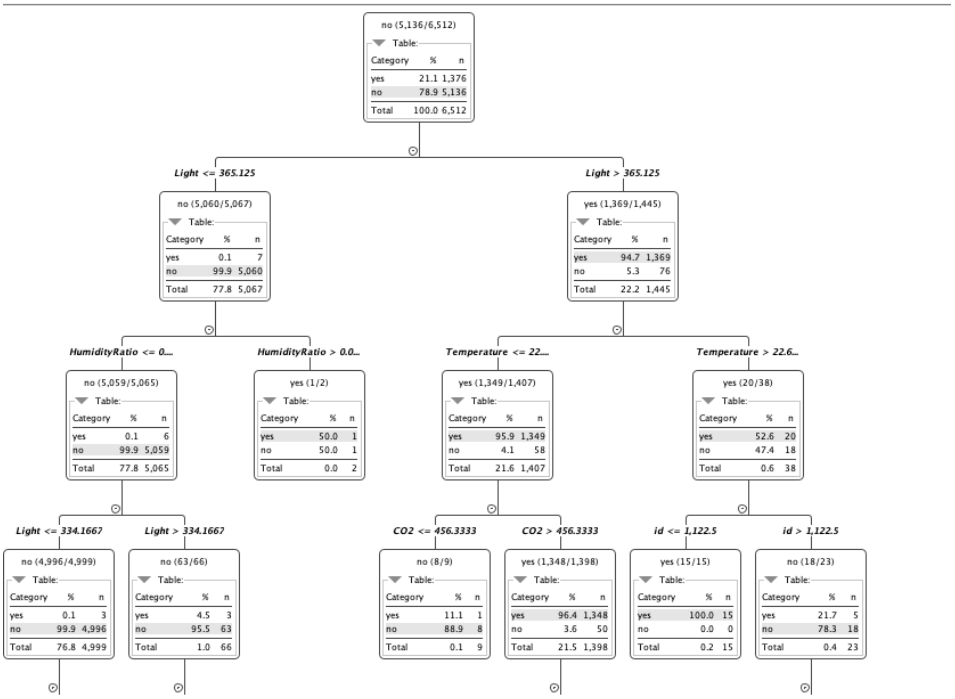
I will be building models to predict the occupancy of the room. I will be using a train set, which is the dataset that I had to clean, and a test set. The train set is the data that was used to train/teach the model – if only one dataset is provided to work with, the train set typically consumes 70-80% of the dataset. The test set is what is used to test the accuracy of the model created by the train set. If the model is highly accurate, then it can be used again with new data. Due to my inability to figure out the nuances of KNIME

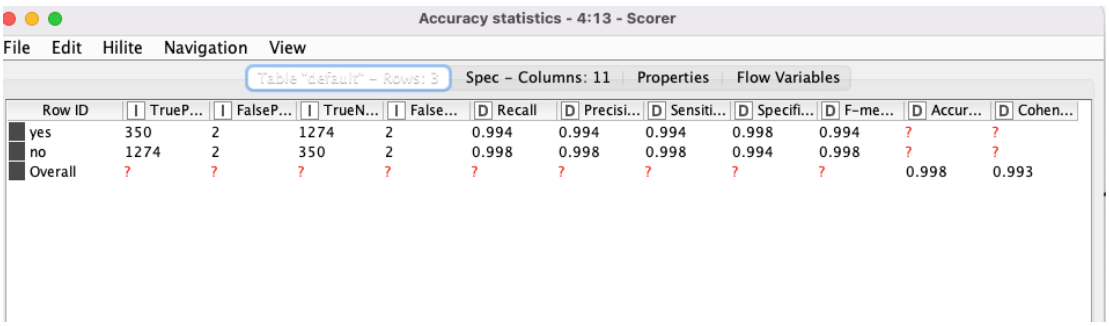
and narrowing down the dataset down to specific columns, I manually had to drop columns from the dataset using `df.drop(['Light', 'CO2', 'HumidityRatio'], axis = 1)` as an example.

Decision Tree One: Temperature and Humidity

I did not choose any stopping features, nor did I use any (honestly, because I did not understand it, especially in KNIME). The accuracy of this model was 99.8%.

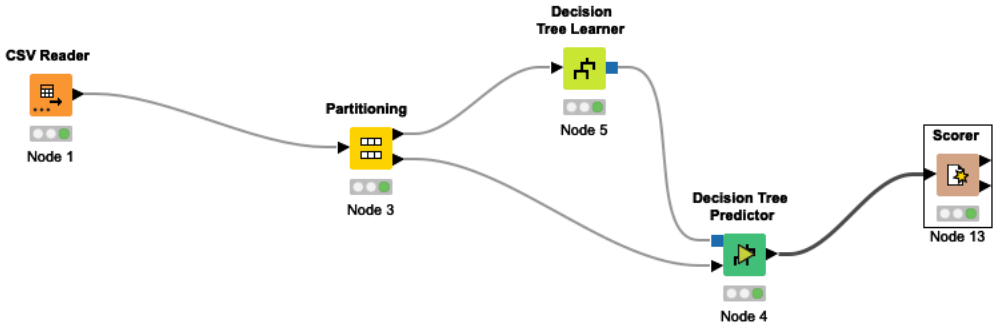


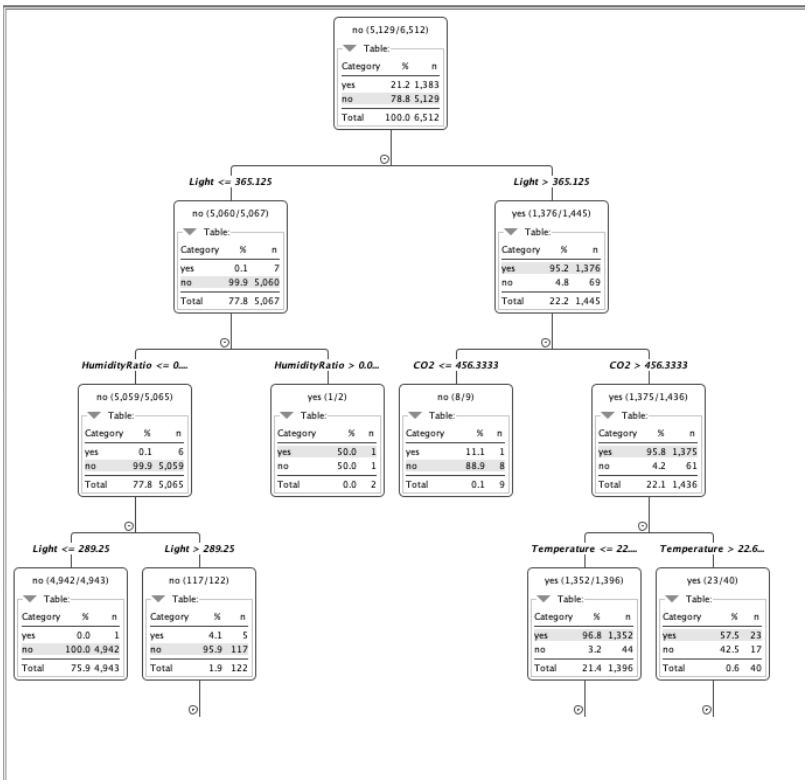


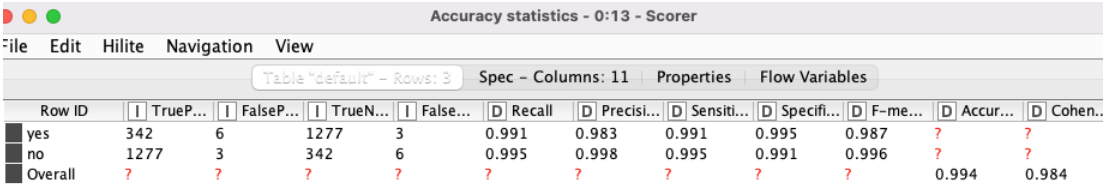


Decision Tree Two: Carbon Dioxide and Humidity Ratio

The accuracy of this model was 99.4%.



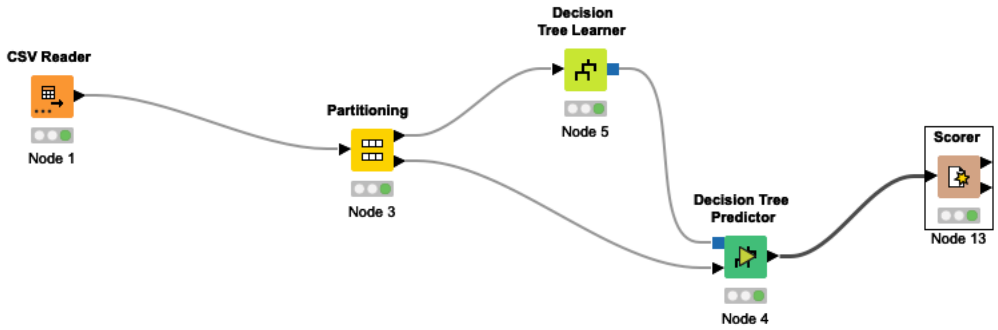


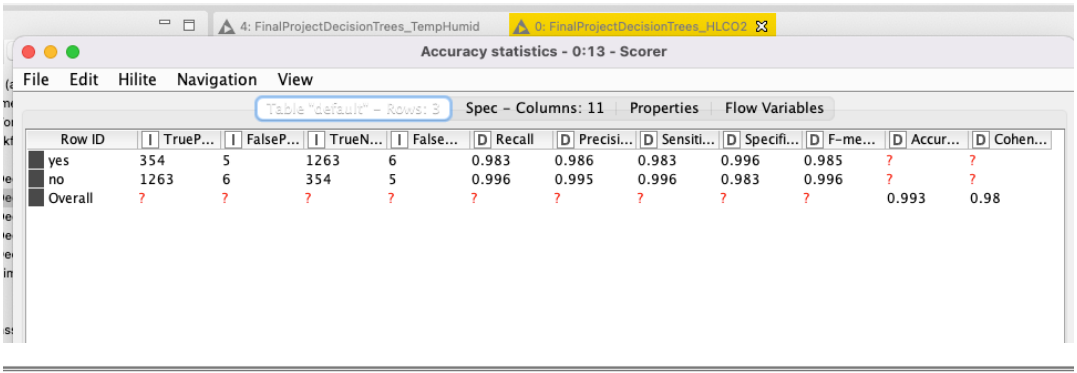


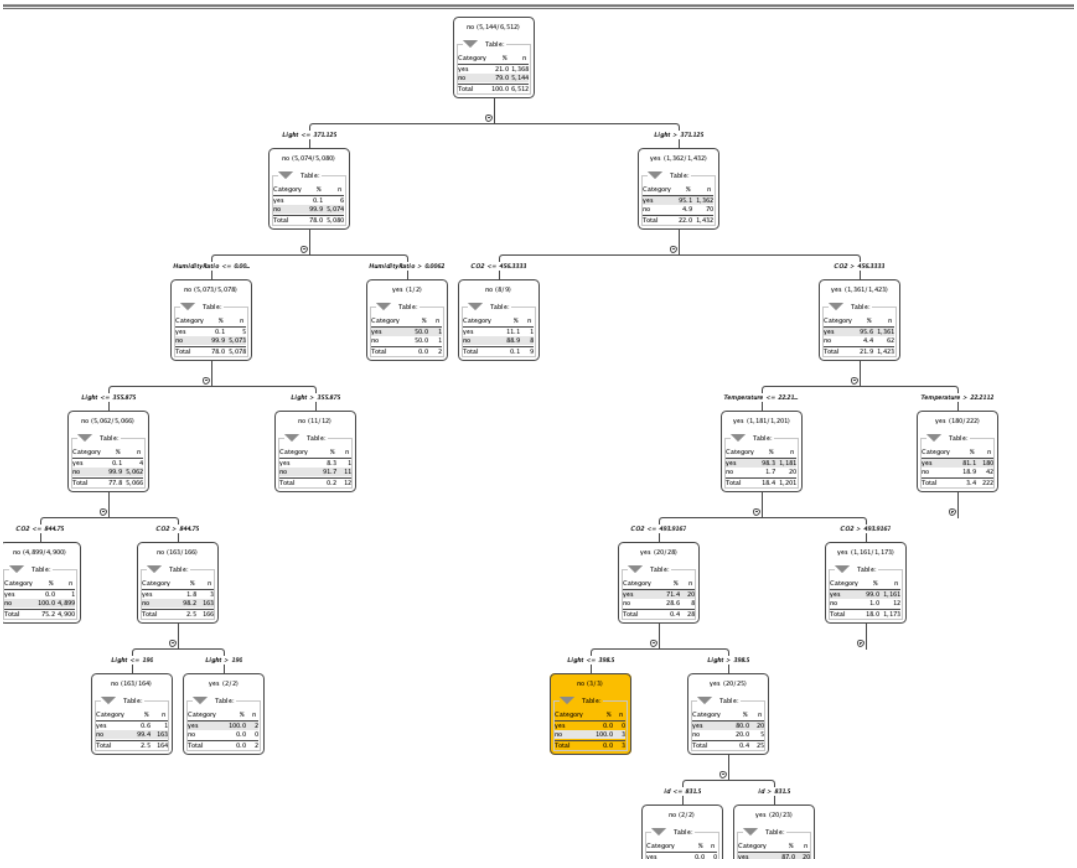
Row ID	TrueP...	FalseP...	TrueN...	False...	Recall	Precisi...	Sensiti...	Specifi...	F-me...	Accur...	Cohen...
yes	342	6	1277	3	0.991	0.983	0.991	0.995	0.987	?	?
no	1277	3	342	6	0.995	0.998	0.995	0.991	0.996	?	?
Overall	?	?	?	?	?	?	?	?	?	0.994	0.984

Decision Tree Three: Humidity, Light, and Carbon Dioxide.

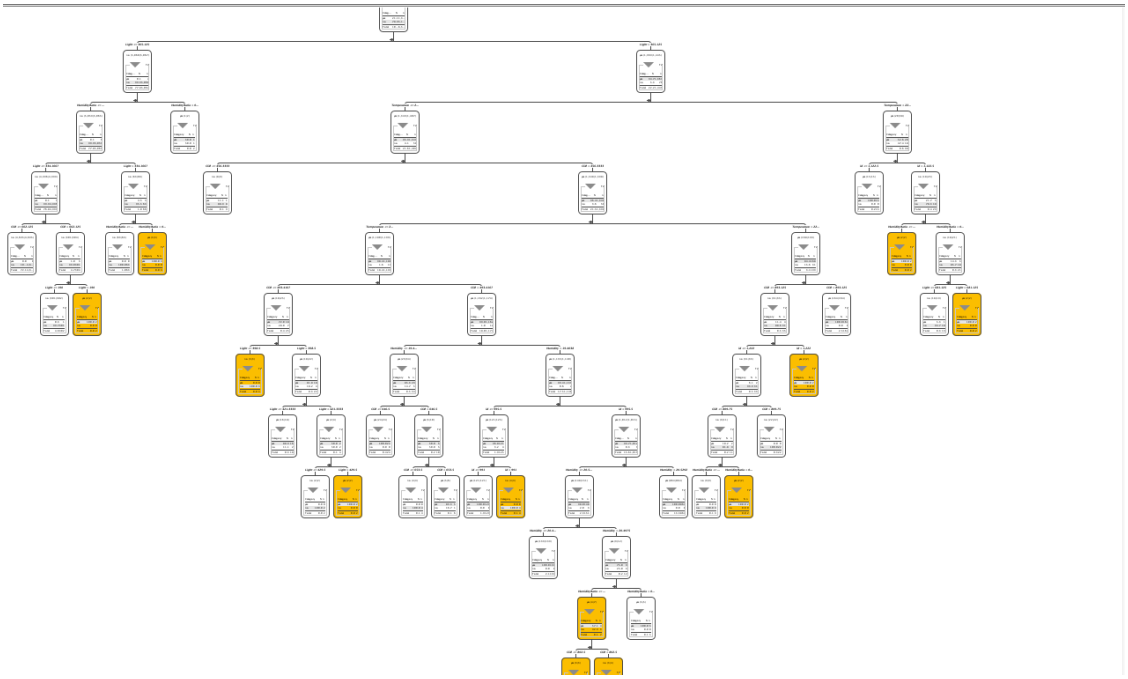
The accuracy of this model was 99.3%.







Of these three decision trees, I would have to go with temperature and humidity, as it had the highest accuracy.



Conclusion

In this project, I cleaned, recleaned, and took the dataset to the dry cleaners. I learned how little I know about KNIME, and that I still have a lot to learn in the Classification area.

I was extremely surprised that the model using temperature and humidity had the highest accuracy, as I had hypothesized to myself that carbon dioxide and humidity ratio would have been higher.

---

[i] (Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models, Energy and Buildings Volume 112, 2016)