# AI part 2 assessment

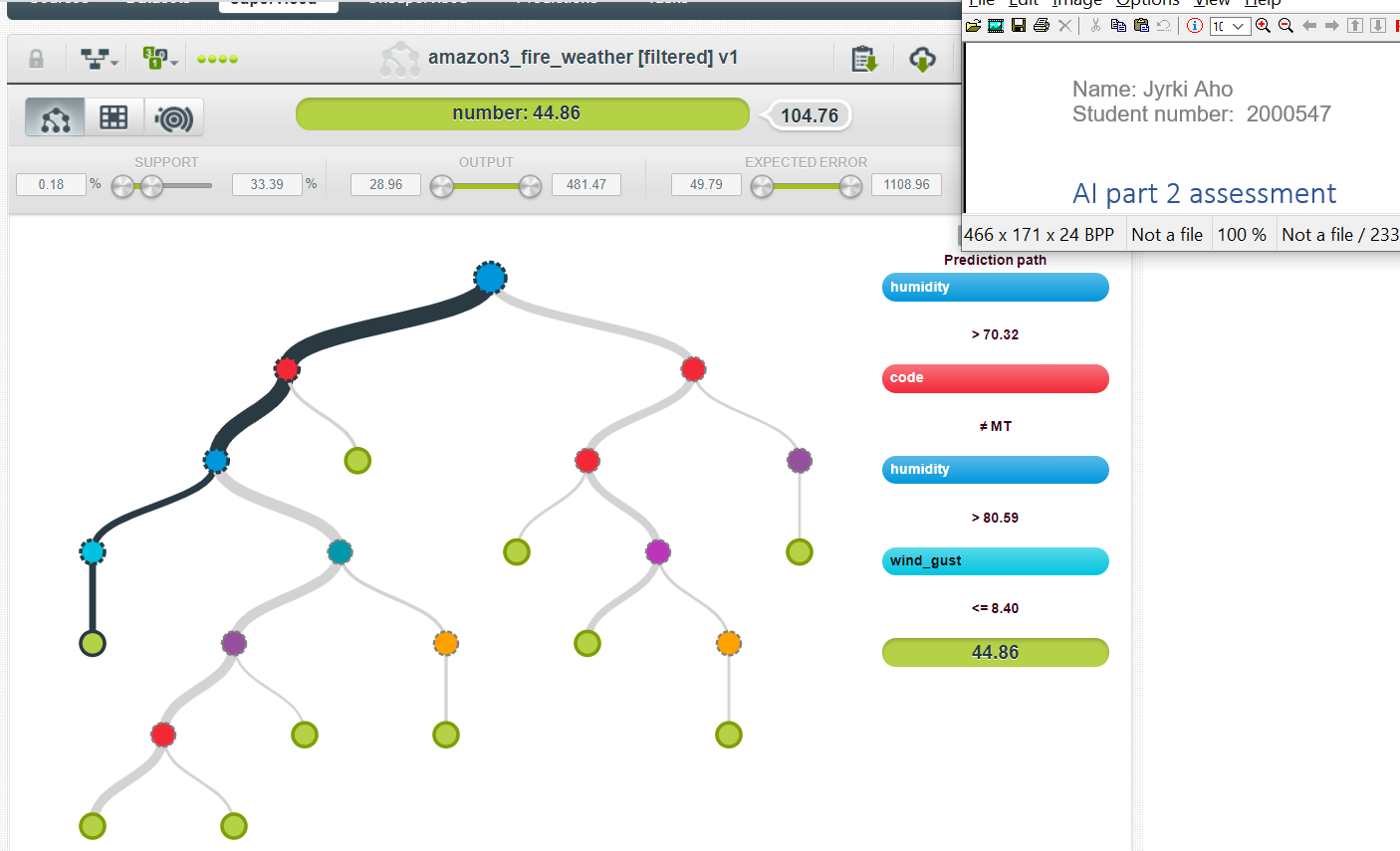
Brazilian forest fires did sound interesting data, but it did lack weather information. Because of that I did download Brazilian forest fire data from address <https://www.kaggle.com/gustavomodelli/forest-fires-in-brazil>. Because this data did not contain weather information, so I downloaded weather data from address <https://www.kaggle.com/saraivaufc/automatic-weather-stations-brazil?select=automatic_weather_stations_inmet_brazil_2000_2021.csv>. This did contain 612 weather stations hourly data from years 2000 to 2021.

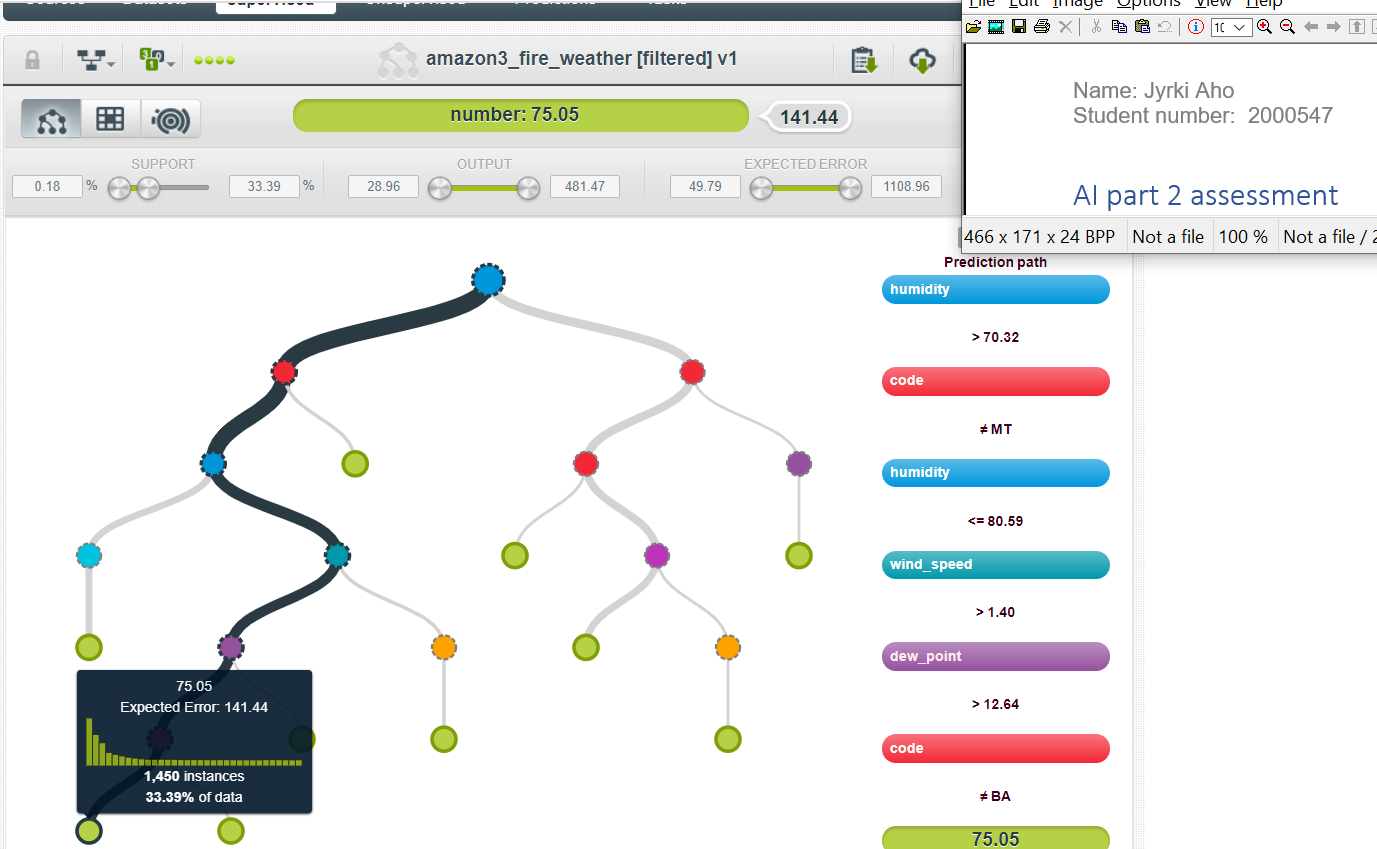
## Processing data

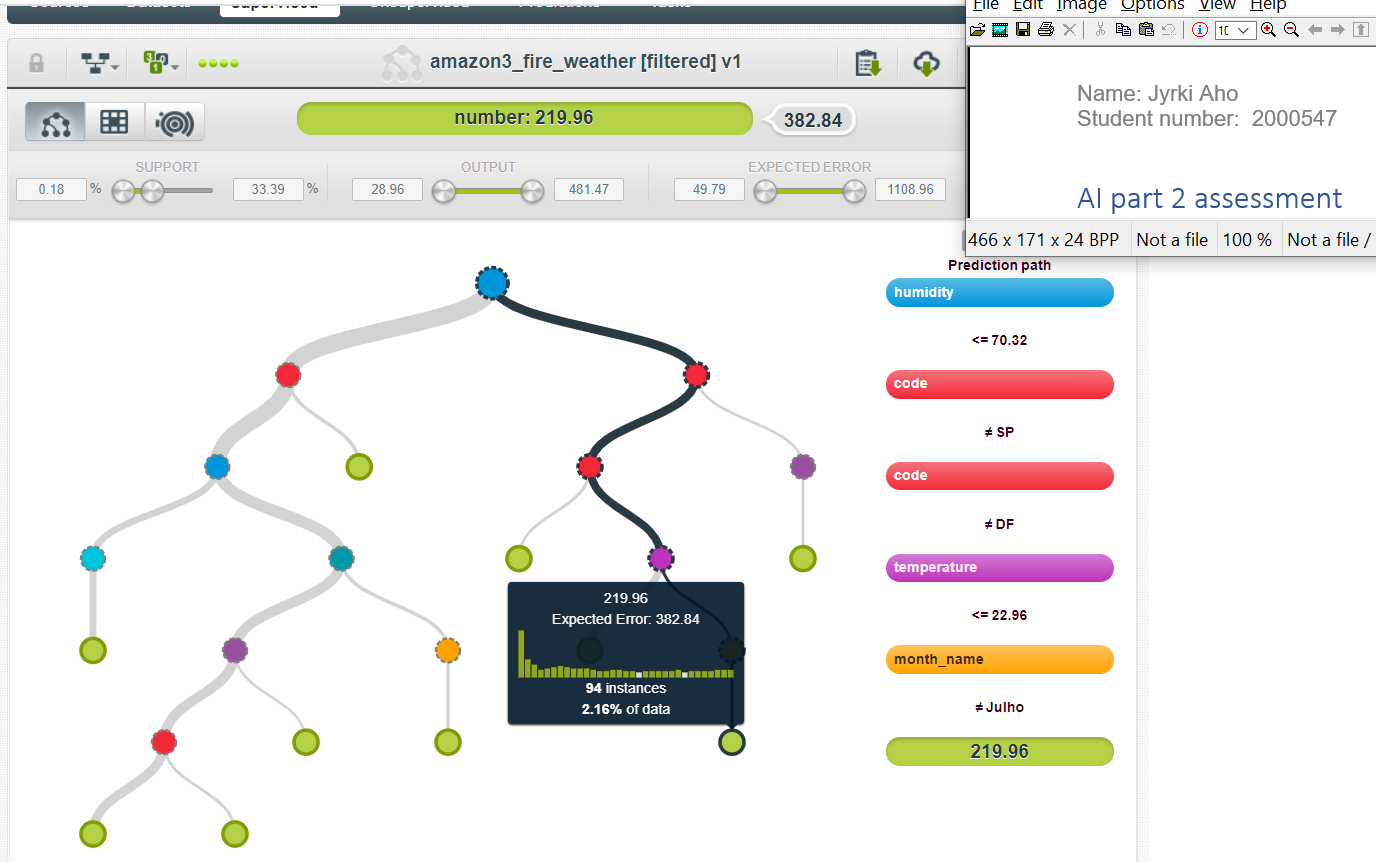
Based on the fores fire datasets dates, the data are gathered monthly and does not contain exact informations when the forest fire has started in which states. Because of this we have to calculate new weather data by grouping data based on month, year and states where the forest fire report has come. We have to also notice that forest fire data has been gathered from 1998 to 2007 and weather data from 2000 to 2021, we have to use inner join to combine these dataset to one big datasets. Because the weather dataset size are around 6 GB, so I did use Pythons Jupyter notebook to combine these datasets. I have added the Python code at the end of this document. I also have to use Python to combine these two datasets, because it was very slow process in Big ML, eventough I did use only one column to combine datasets.

## Decision tree

First I did try to model data with decision trees. Those decision trees did seems to have too many values to make predictions, so I did drop all the min and max values from the datasets. I also did take of month information. When I did choose Big ML to optimize data, the server did take some time to process data and to create tree. It did create following simplified decision tree.







The model summary report did show column importance as

Field importance:

1. code: 40.42%

2. humidity: 24.25%

3. month\_name: 9.75%

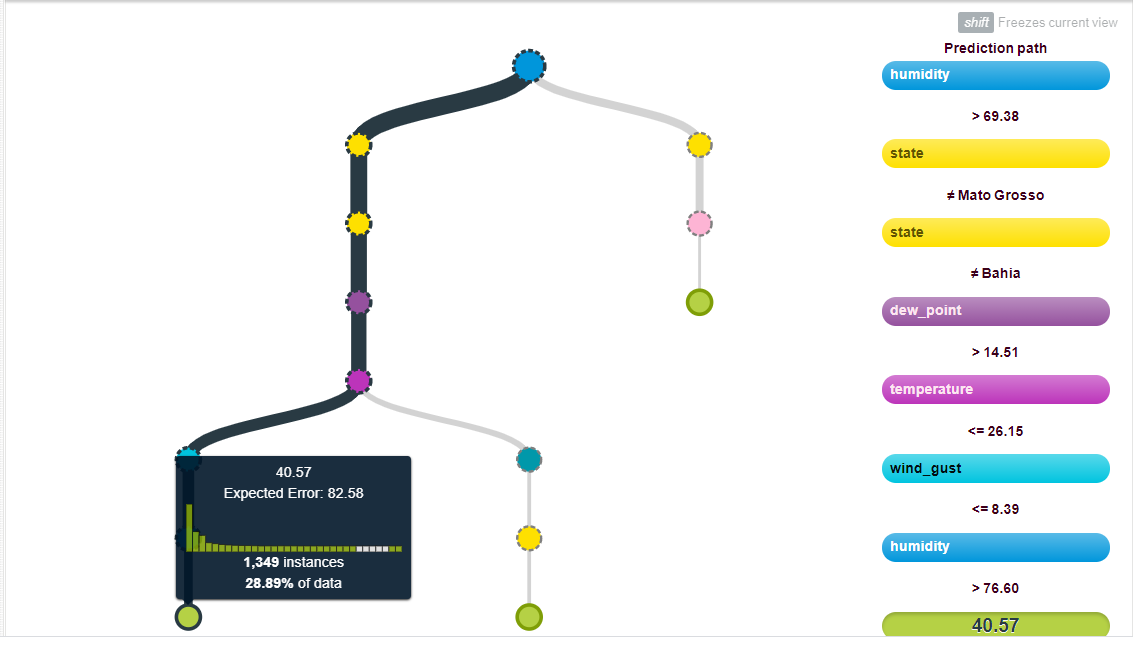
4. dew\_point: 8.81%

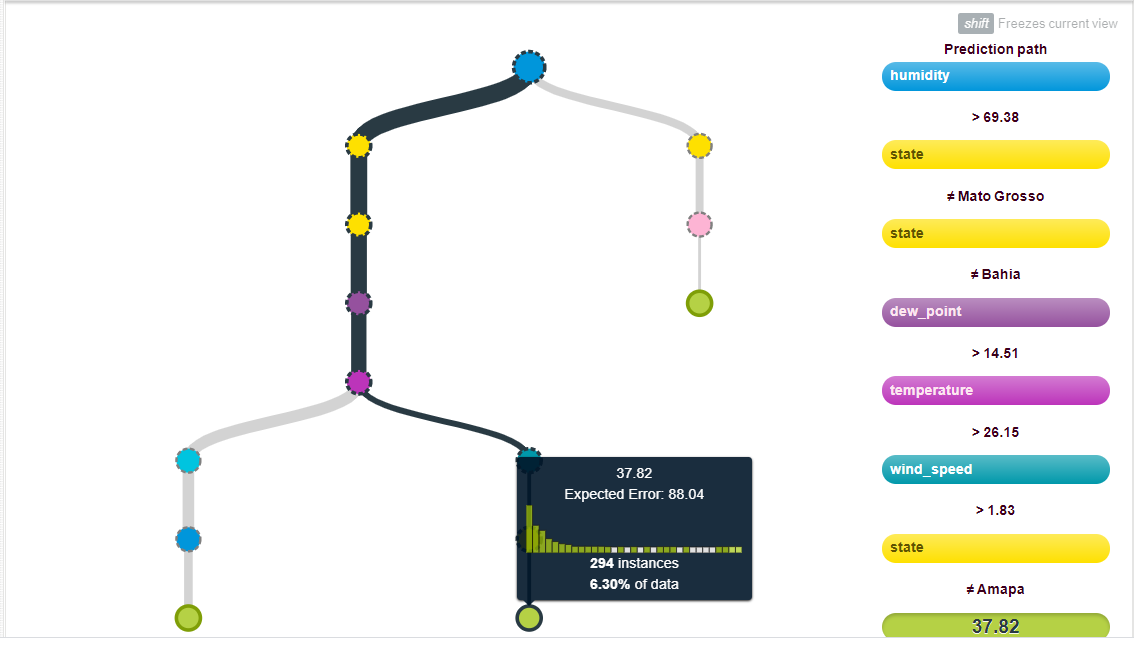
5. temperature: 6.98%

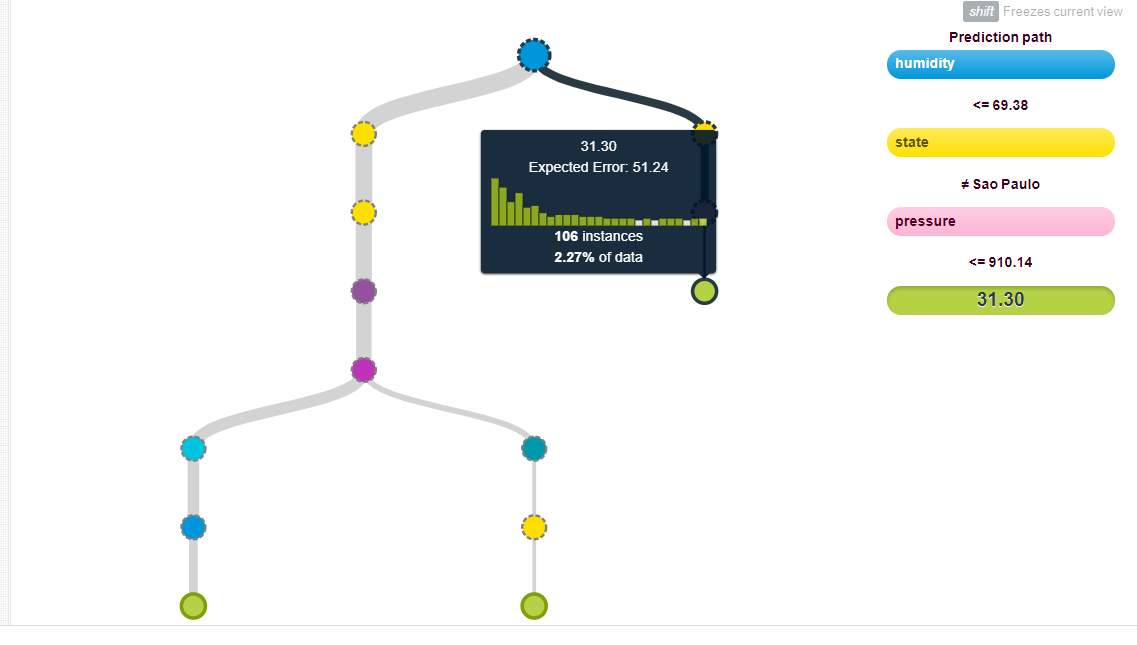
6. wind\_speed: 6.03%

7. wind\_gust: 3.76%

If we try other dataset where we drop min/max information, wind direction and code (which correlates with states). Then we can create new decision tree with optimizing data, and then we get following model.







Then we get new field importance as

1. state: 35.08%

2. humidity: 16.58%

3. temperature: 12.73%

4. dew\_point: 8.87%

5. month\_name: 8.21%

6. pressure: 5.89%

7. wind\_speed: 5.46%

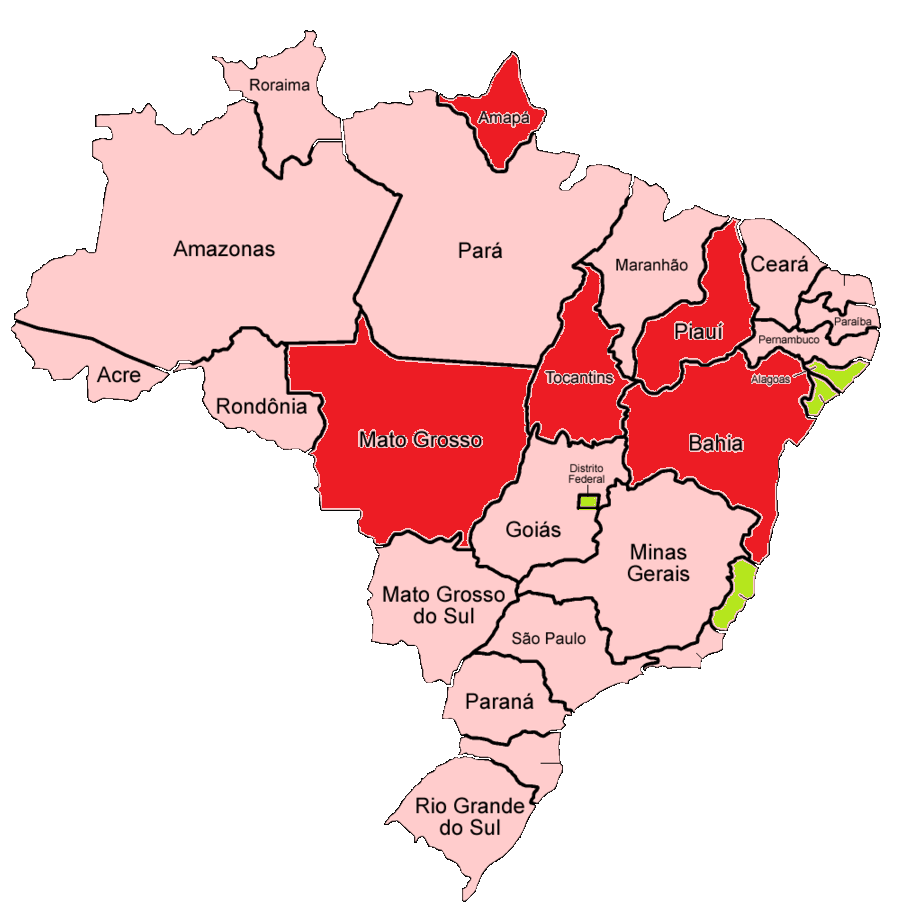
8. wind\_gust: 4.92%

9. radiation: 2.27%

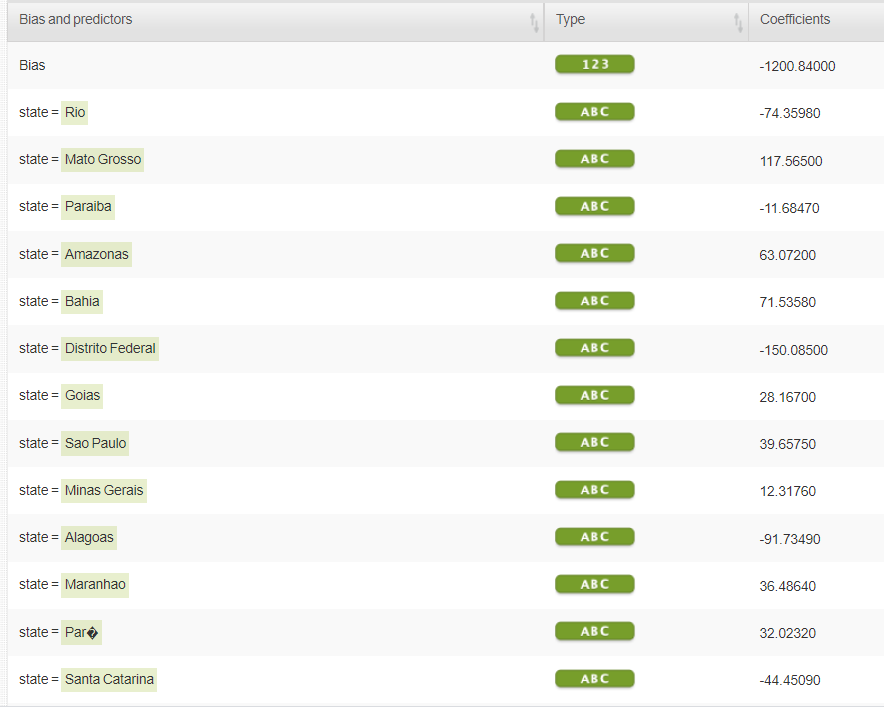
Using this information we could understand that some states have more forest fires than others. From this we could conclude that the vegetation and forest structure differs by state to state. This is understandable given the size of Brazil, in addition to information that the edges of rainforests are being burned in Brazil and new fields are being created in these places. It is also interesting information, that humidity does have effect of forest fire. According to the U.S. National Park Service, relative humidity is important because dead vegetation and air exchange moisture with each other. Low humidity absorbs water from dead vegetation and high humidity in turn transfers water to dead vegetation. In particular, small debris such as conifers react to changes faster than, for example dead branches. Months also have effect of forest fire, which probably means that some months are hotter and some a rainier, which probably has direct correlation to forest fires. It also seems that dew point, temperature and wind speed does have some effect to forest fires. Wind gust probably has some effect which helps fire to spread faster.

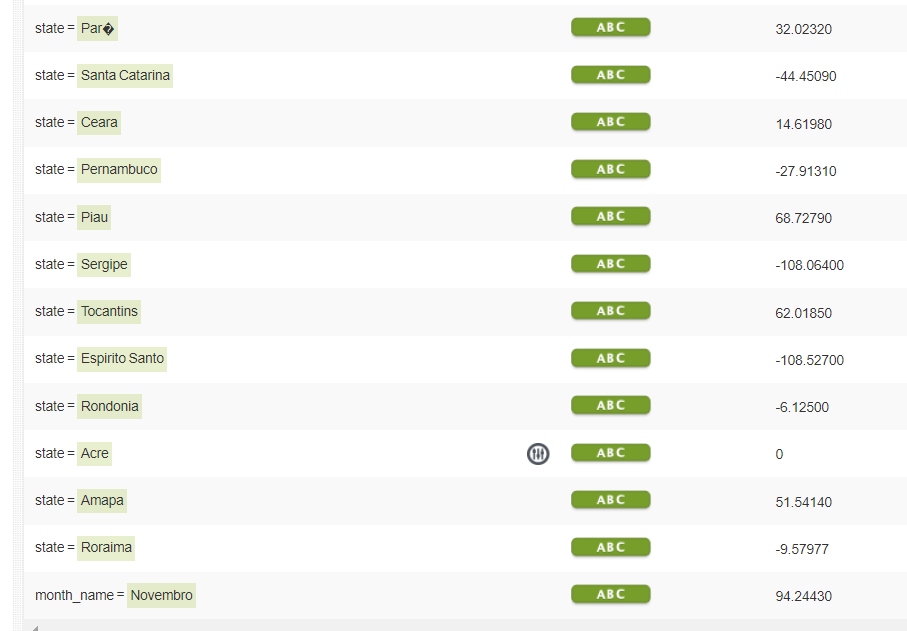
## Linear regression

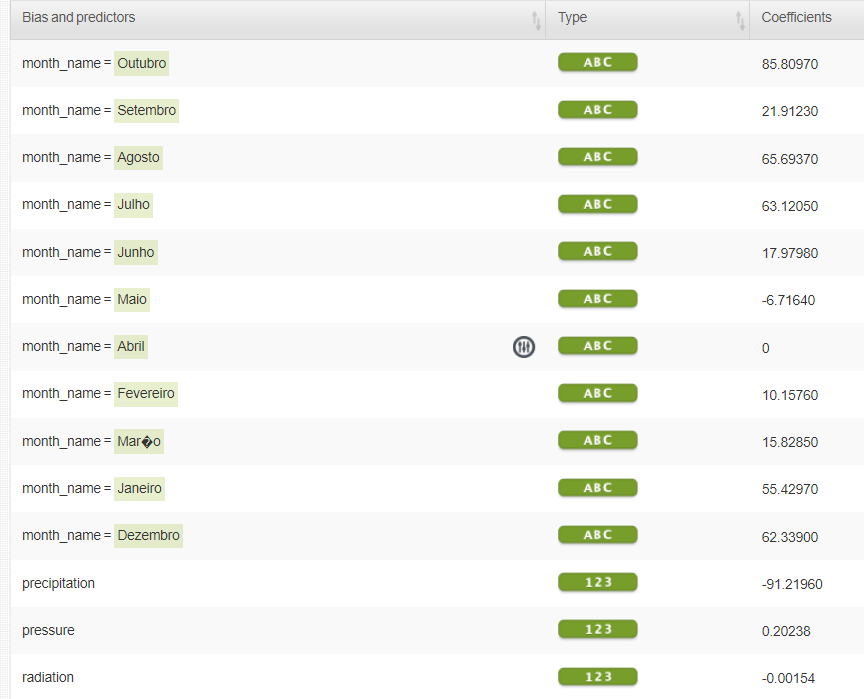
Linear regression model did not seem to work as I were expecting it to work. It does show information of the data, but I have slight problem to understand PDP graph. But when creating different dataset, we can see that some states have much more forest fires than others. Example Espirito Santo, Sergipe, Alagoas, Distrito Federal and Rio have much less forest fires than others. Instead Mato Grosso, Amapa, Tocantis, Piau and Bahia have much more forest fires. I did use Wikipedia pictures of Brazil (<https://commons.wikimedia.org/wiki/File:Brazil_states_named.png>), in which I did add green color to show states where are less forest fires and with red color states where are more forest fires than other states.



Linear model did also show that most likely forest fire occurs in July, October, November, and December. It also seems that higher temperature and precipitation lowers the risk of forest fires, but dew point increases the risk of forest fires. Dataset also did show that forest fires are slowly increasing.









## Sources

US National Park Service. 2021. Understanding fire danger. Can be read at: <https://www.nps.gov/articles/understanding-fire-danger.htm>. Read: 16.7.2021.

# Python code

#Handling forest fire data

import pandas as pd

import numpy as np

import matplotlib as plt

amazon\_df = pd.read\_csv("amazon.csv")

print(len(amazon\_df)) # 6 454 datapoints

# change month to numbers

month\_map={'Janeiro': 1, 'Fevereiro': 2, 'Mar�o': 3, 'Abril': 4, 'Maio': 5,

'Junho': 6, 'Julho': 7, 'Agosto': 8, 'Setembro': 9, 'Outubro': 10,

'Novembro': 11, 'Dezembro': 12}

amazon\_df['month\_nbr']=amazon\_df['month'].map(month\_map)

amazon\_df['yearmonth'] = amazon\_df['year']\*100 + amazon\_df['month\_nbr']

amazon\_df.head(50)

# Check for NaN values

print( amazon\_df['month\_nbr'].isnull().values.any() ) # False

k = amazon\_df['state'].unique()

k.sort()

print(k) # contains name of 23 Brazilian states

# creating are codelist

sc\_df = pd.read\_csv("automatic\_stations\_codes\_2000\_2021.csv",delimiter=";")

sc\_df.head(20)

codigo = sc\_df['CODIGO']

uf = sc\_df['UF']

state\_map2 = dict(zip(codigo,uf))

print(state\_map2)

print(len(codigo))

#creating state map list

amazon\_df['code']=amazon\_df['state'].map(state\_map)

print(amazon\_df)

#Save data to file

amazon\_df.to\_csv("amazon2.csv",sep=';')

#Handling weather data using previous lists

weather\_df = pd.read\_csv("automatic\_weather\_stations\_inmet\_brazil\_2000\_2021.csv", delimiter=";")

weather\_df.head(40)

n = len(weather\_df)

print(n) # contains 60 452 376 data points

#Contains multiple difficult column names, which have to simplify

h = weather\_df.columns

#print(h)

new\_header=['station', 'date', 'hour', 'precipitation', 'pressure', 'max\_preasure', 'min\_preasure',

'radiation', 'temperature', 'dew\_point', 'max\_temperature', 'min\_temperature',

'max\_dew\_point', 'min\_dew\_point', 'max\_humidity', 'min\_humidity', 'humidity', 'wind\_direction',

'wind\_gust', 'wind\_speed']

z = dict(zip(h,new\_header))

weather\_df= weather\_df.rename(columns=z)

#weather\_df.head(20)

# data contains NaN values, which are ok, but -9999 values have to change to NaN values

cols = ['precipitation', 'pressure', 'max\_preasure', 'min\_preasure',

'radiation', 'temperature', 'dew\_point', 'max\_temperature', 'min\_temperature',

'max\_dew\_point', 'min\_dew\_point', 'max\_humidity', 'min\_humidity', 'humidity',

'wind\_gust', 'wind\_speed']

for c in cols:

weather\_df[c]= weather\_df[c].apply(lambda x: np.nan if x<-1000 else x)

#lets create yearmonth column

weather\_df['year'] = [int(x[0:4]) for x in weather\_df['date']]

weather\_df['month'] = [int(x[5:7]) for x in weather\_df['date']]

weather\_df['yearmonth'] = [(100\*int(x[0:4]) + int(x[5:7])) for x in weather\_df['date']]

#Lets add area code column to data

weather\_df['code']=weather\_df['station'].map(state\_map2)

weather\_df.head(20)

# Now we will create new dataframes and then we combine weather data to one file

dataset\_mean = weather\_df.groupby(['yearmonth', 'code']).mean()

dataset\_max = weather\_df.groupby(['yearmonth','code']).max()

dataset\_min = weather\_df.groupby(['yearmonth','code']).min()

dataset\_weather = dataset\_mean.copy()

dataset\_weather.drop(columns=['hour'])

dataset\_weather['max\_preasure'] = dataset\_max['max\_preasure']

dataset\_weather['min\_preasure'] = dataset\_min['min\_preasure']

dataset\_weather['max\_temperature'] = dataset\_max['max\_temperature']

dataset\_weather['min\_temperature'] = dataset\_min['min\_temperature']

dataset\_weather['max\_dew\_point'] = dataset\_max['max\_dew\_point']

dataset\_weather['min\_dew\_point'] = dataset\_min['min\_dew\_point']

dataset\_weather.head(20)

dataset\_weather.to\_csv("amazon\_weather.csv", sep=";")

#Lets now combine brazilian fire and weather data

amazon\_weather = pd.read\_csv("amazon\_weather.csv", delimiter=";")

amazon\_weather.drop(amazon\_weather.columns[amazon\_weather.columns.str.contains('unnamed',case = False)],axis = 1, inplace = True)

amazon\_weather.head(10)

a = amazon\_weather['code']

b = amazon\_weather['yearmonth']

amazon\_weather['id'] =amazon\_weather[['code','yearmonth']].astype(str).apply(''.join,1)

amazon\_weather= amazon\_weather.drop(columns=['hour','yearmonth'])

amazon\_weather.head(10)

amazon\_fire = pd.read\_csv("amazon2.csv", delimiter=";")

amazon\_fire.drop(amazon\_fire.columns[amazon\_fire.columns.str.contains('unnamed',case = False)],axis = 1, inplace = True)

amazon\_fire.head(10)

amazon\_fire['id'] =amazon\_fire[['code','yearmonth']].astype(str).apply(''.join,1)

amazon\_fire = amazon\_fire.drop(columns=['year', 'month\_nbr','yearmonth','date','code'])

amazon\_fire.head(10)

amazon3 = pd.merge(amazon\_fire, amazon\_weather, how='inner', left\_on='id', right\_on ='id')

amazon3.head(20)

amazon3.to\_csv("amazon3\_fire\_weather.csv",sep=";",index=False)