Comparative Data Analysis

Dhruv Parikh

Overview

Historically, people have considered earthquakes to be one of nature's most hazardous risks. Earthquakes continue to serve as a reminder that nature is still capable of striking suddenly and in a matter of seconds, leaving destruction and casualties in its wake.

Why is it important?

Our understanding of the causes of earthquakes may enable us to develop measures for their prevention. We might learn how to forecast earthquakes of all sizes in the future, which would enhance how societies react to earthquakes.

Models Used:-

Linear Regression

1)RMSE:-0.42

2)Score:-1

RandomForestRegresssor

1)RMSE:-0.418

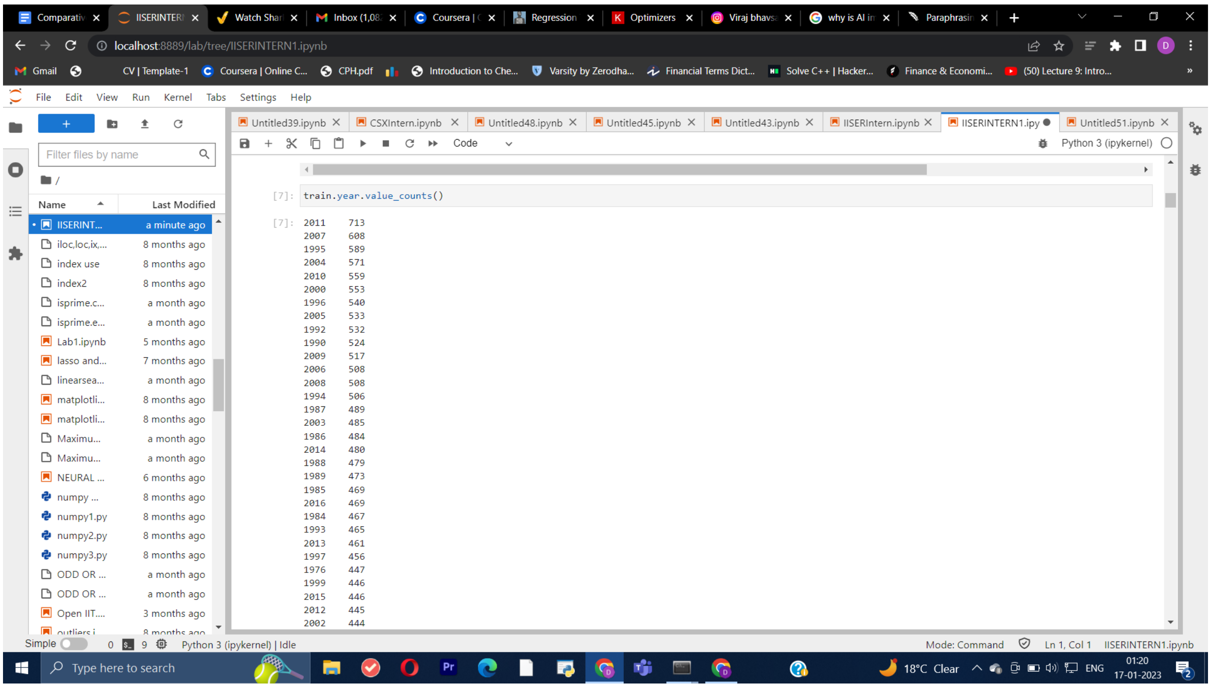
Neural Networks

1)Loss Function:-0.17

Information on the dataset

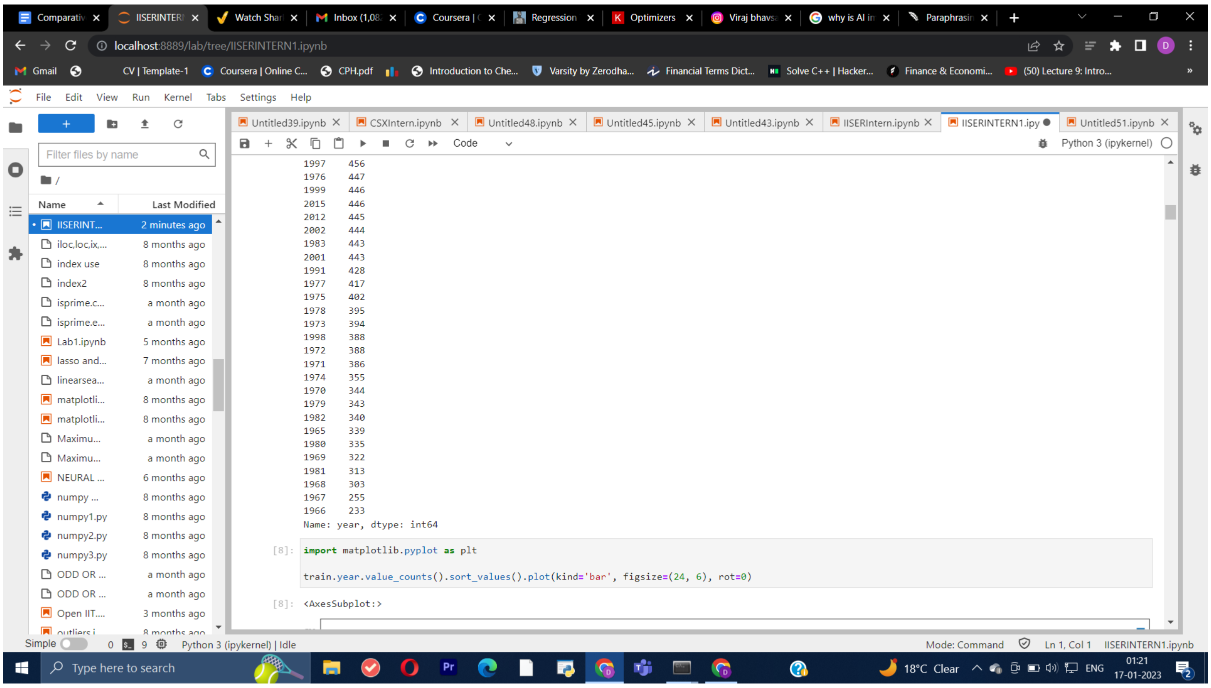
- The maximum earthquake happened in the year 2011 which had a count of 713
- And Minimum earthquake happened in the year 1966 which had a count of 233

- The month in which most earthquakes happened was the month of **March** which had a count of 2104 earthquakes
- The month in which we had the lowest number of earthquakes was the month of **June** with a count of 1807 earthquakes



```
[7]: train.year.value_counts()

[7]: 2011    713
     2007    608
     1995    589
     2004    571
     2010    559
     2000    553
     1996    540
     2005    533
     1992    532
     1990    524
     2009    517
     2006    508
     2008    508
     1994    506
     1987    489
     2003    485
     1986    484
     2014    480
     1988    479
     1989    473
     1985    469
     2016    469
     1984    467
     1993    465
     2013    461
     1997    456
     1976    447
     1999    446
     2015    446
     2012    445
     2002    444
```

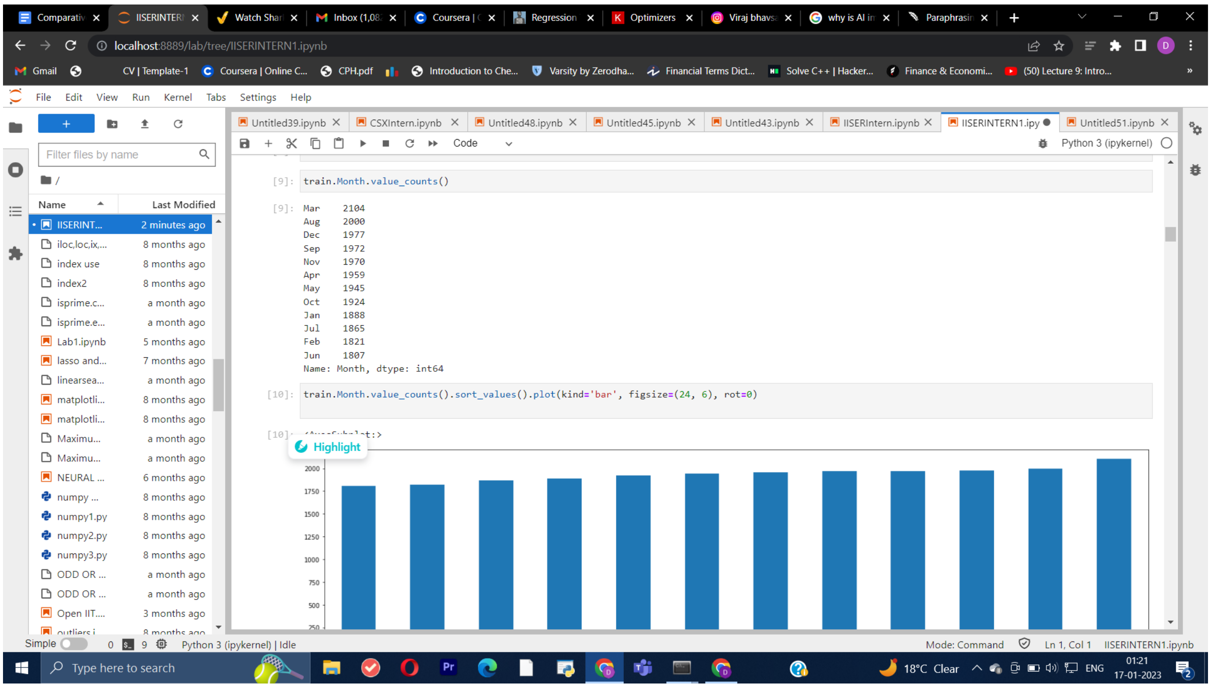
localhost:8889/lab/tree/IISERINTERN1.ipynb
Python 3 (ipykernel) | Idle
Mode: Command
Ln 1, Col 1
IISERINTERN1.ipynb
01:20
17-01-2023



```
     1997    456
     1976    447
     1999    446
     2015    446
     2012    445
     2002    444
     1983    443
     2001    443
     1991    428
     1977    417
     1975    402
     1978    395
     1973    394
     1998    388
     1972    388
     1971    386
     1974    355
     1970    344
     1979    343
     1982    340
     1965    339
     1980    335
     1969    322
     1981    313
     1968    303
     1967    255
     1966    233
     Name: year, dtype: int64

[8]: import matplotlib.pyplot as plt

     train.year.value_counts().sort_values().plot(kind='bar', figsize=(24, 6), rot=0)

[8]: <AxesSubplot:>
```

localhost:8889/lab/tree/IISERINTERN1.ipynb
Python 3 (ipykernel) | Idle
Mode: Command
Ln 1, Col 1
IISERINTERN1.ipynb
01:21
17-01-2023

Month-Wise Count



```
[9]: train.Month.value_counts()
```

```
[9]: Mar    2104
     Aug    2000
     Dec    1977
     Sep    1972
     Nov    1970
     Apr    1959
     May    1945
     Oct    1924
     Jan    1888
     Jul    1865
     Feb    1821
     Jun    1807
     Name: Month, dtype: int64
```

```
[10]: train.Month.value_counts().sort_values().plot(kind='bar', figsize=(24, 6), rot=0)
```

```
[10]: <AxesSubplot:>
```

Information About the Location

I worked on the datasets and used software like geopandas,tqd to find the location using the latitude and longitude

In that dataset, I found that the country with the maximum earthquake was the country of **Indonesia** with 2484 earthquakes followed by **Papua Niugini** with 1183 earthquakes and the country of **India** had only 41 earthquakes

- So with very few earthquakes, it would be tough for me to work on India's Dataset so I worked on Indonesia's Dataset and trained the dataset in the month of March from the year 1965-1999 and then worked on it for the year 2000 to 2016 to find the following the magnitude as the dataset contained very less for a single year.

Final Loss functions for the following datasets when worked for a particular month

Random Forest Regressor

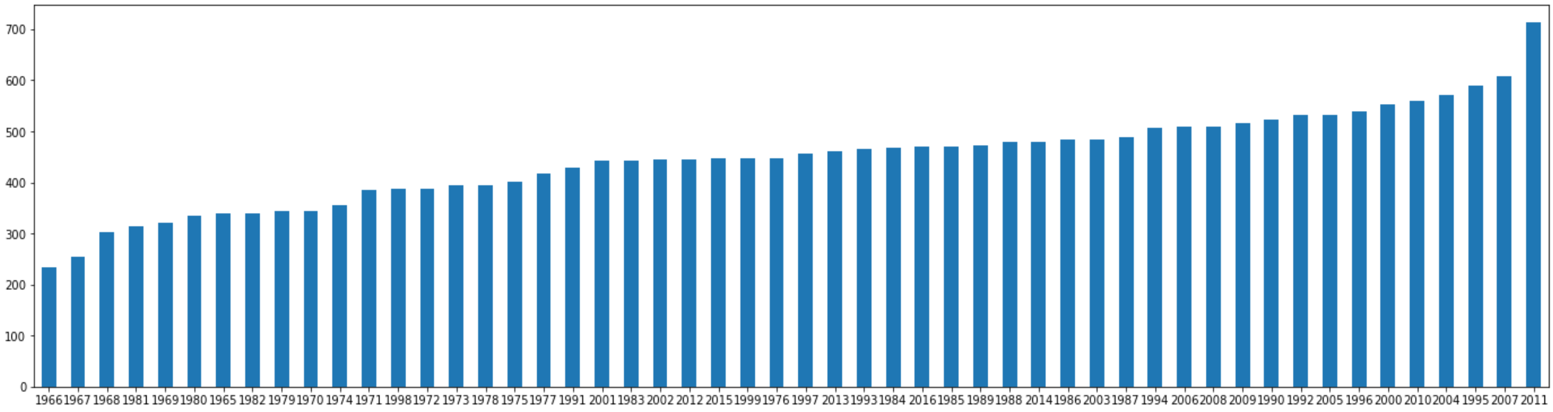
Loss:-0.46

Neural Networks

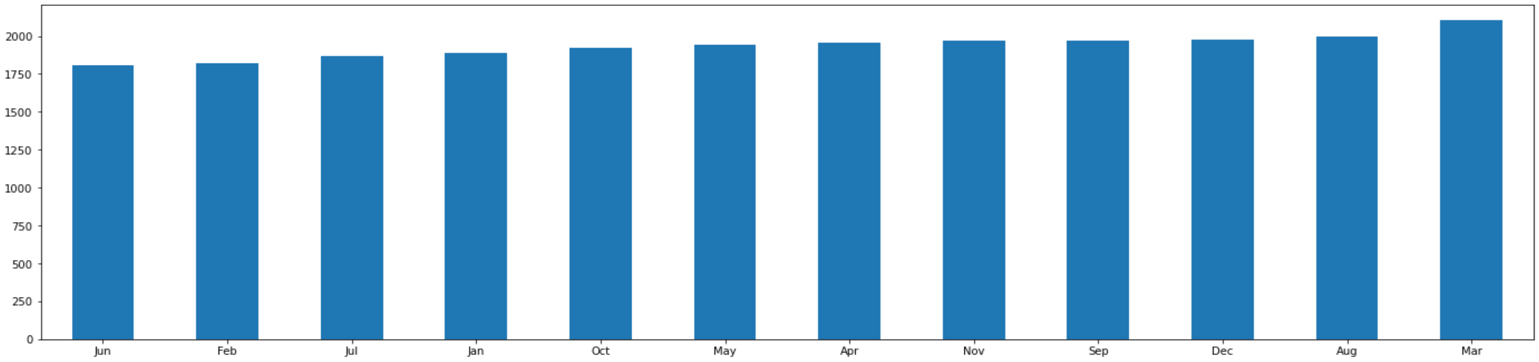
Loss function:-0.1087

I also made two python notebooks so that one contained a dataset which consisted of one dataset of address and in the other, I used the dataset to find the loss function.

FINAL BAR GRAPH for Year vs Number



Final Bar Graph for MONTH vs earthquakes



Accuracy of the models

Now to get a better insight into the model we have to classify how accurately the model predicts the following values of different models

The First Step for the following will be:-

We will set an error limit, For our case, we will set the error limit to 10% of the average value of the dataset which came to about 0.5-0.7

What is the error limit?

Error Limit is the deviance of the given dataset from the predicted value given to us by using various datasets.

How did we convert the values?

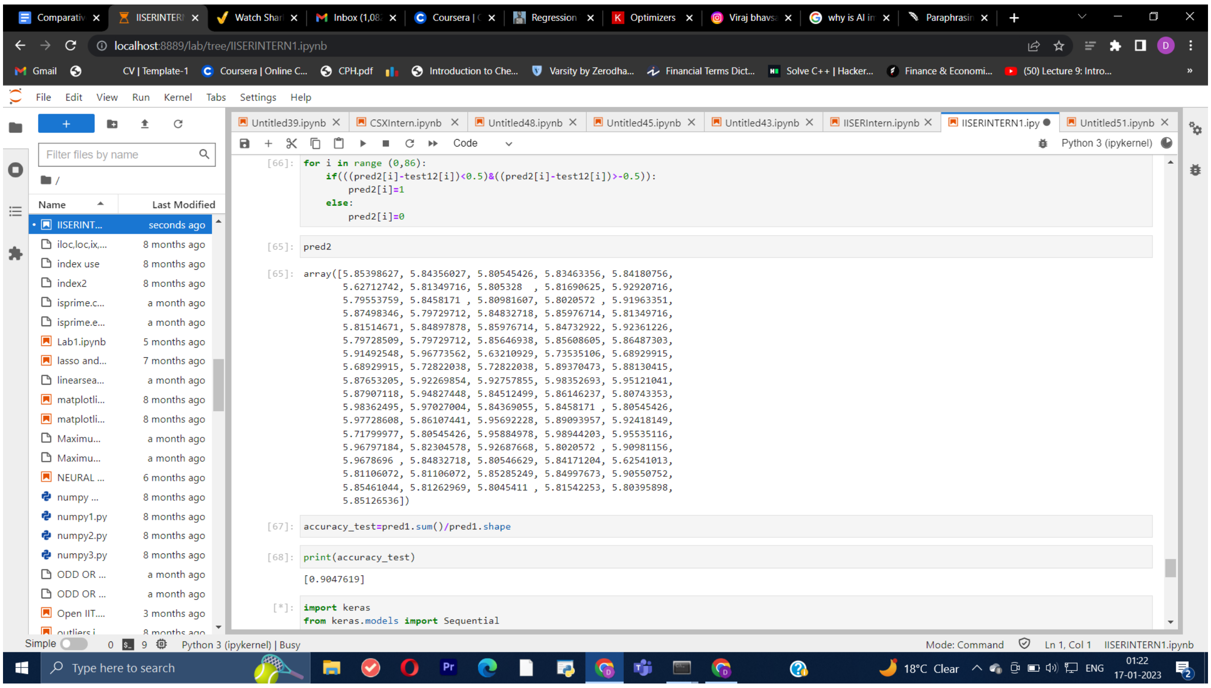
We subtracted the value of output we got from the predicted model and the input train value we had if it had a loss greater than the error limit then we would give that number a 0 and if not then we would give that number a 1.

Next Step:-

We will predict the value of the magnitude using two models

RandomForestRegressor

On using the following pattern we got the following accuracy of the model to be:-0.9047619



```
[66]: for i in range (0,86):
          if(((pred2[i]-test12[i])<0.5)&((pred2[i]-test12[i])>-0.5)):
              pred2[i]=1
          else:
              pred2[i]=0
```

```
[65]: pred2
```

```
[65]: array([5.85398627, 5.84356027, 5.80545426, 5.83463356, 5.84180756,
             5.62712742, 5.81349716, 5.805328  , 5.81690625, 5.92920716,
             5.79553759, 5.8458171 , 5.80981607, 5.8020572 , 5.91963351,
             5.87498346, 5.79729712, 5.84832718, 5.85976714, 5.81349716,
             5.81514671, 5.84897878, 5.85976714, 5.84732922, 5.92361226,
             5.79728509, 5.79729712, 5.85646938, 5.85608605, 5.86487303,
             5.91492548, 5.96773562, 5.63210929, 5.73535106, 5.68929915,
             5.68929915, 5.72822038, 5.72822038, 5.89370473, 5.88130415,
             5.87653205, 5.92269854, 5.92757855, 5.98352693, 5.95121041,
             5.87907118, 5.94827448, 5.84512499, 5.86146237, 5.80743353,
             5.98362495, 5.97027004, 5.84369055, 5.8458171 , 5.80545426,
             5.97728608, 5.86107441, 5.95692228, 5.89093957, 5.92418149,
             5.71799977, 5.80545426, 5.95884978, 5.98944203, 5.95535116,
             5.96797184, 5.82304578, 5.92687668, 5.8020572 , 5.90981156,
             5.9678696 , 5.84832718, 5.80546629, 5.84171204, 5.62541013,
             5.81106072, 5.81106072, 5.85285249, 5.84997673, 5.90550752,
             5.85461044, 5.81262969, 5.8045411 , 5.81542253, 5.80395898,
             5.85126536])
```

```
[67]: accuracy_test=pred1.sum()/pred1.shape
```

```
[68]: print(accuracy_test)

      [0.9047619]
```

```
[*]: import keras
     from keras.models import Sequential
```

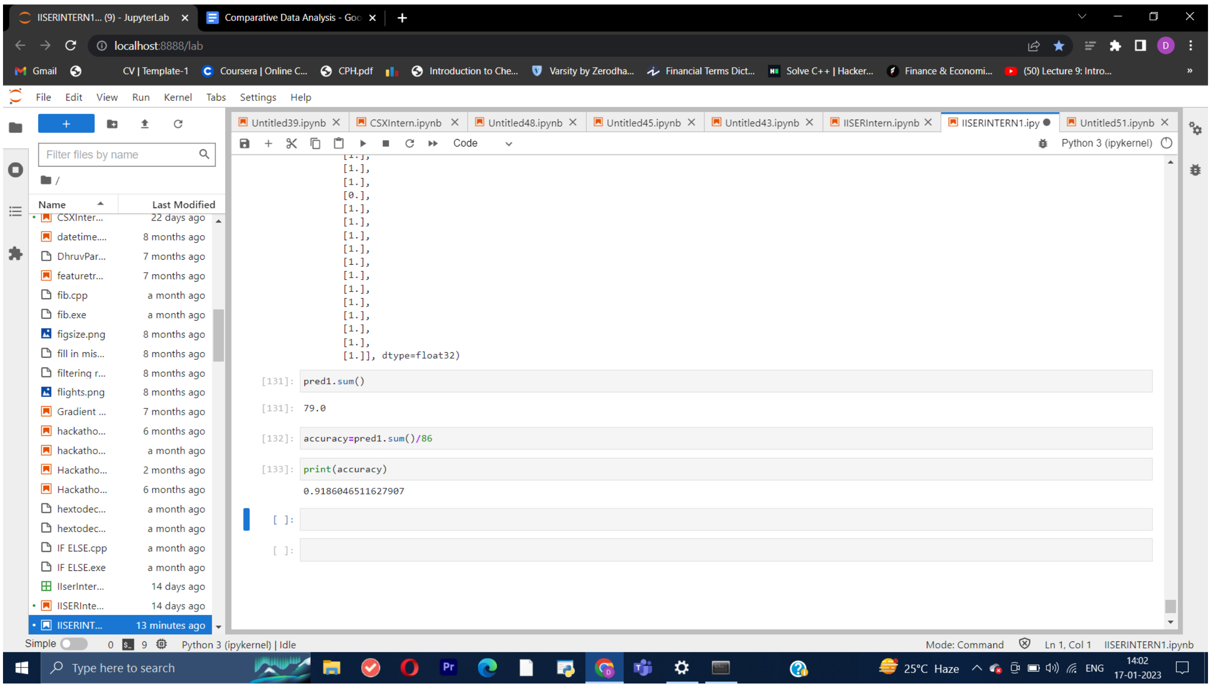
Neural Networks

In Neural Network we had a 5-layer model with the activation function being relu and the final function being linear.

I used the Adam optimizer with the learning rate of the optimizer being 0.02.

I also used the loss which was the mean squared error.

On using the following HyperParameters I found the data gave us an Accuracy of 0.9186.



```
            [1.],
            [1.],
            [0.],
            [1.],
            [1.],
            [1.],
            [1.],
            [1.],
            [1.],
            [1.],
            [1.],
            [1.],
            [1.],
            [1.],
            [1.]], dtype=float32)

[131]: pred1.sum()

[131]: 79.0

[132]: accuracy=pred1.sum()/86

[133]: print(accuracy)

       0.9186046511627907

[ ]:

[ ]:
```