Document for Data Mining project

Implementing Data Mining and ML methods to Datasets

Amirhossein Shahsafi

Dr.Mirzarezaei

Islamic Azad University SRB

7 - Jul - 22

Implemented Methods:

Classification:

- K-Nearest Neighbors
- Desiccation Tree
- Random Forest

Clustering:

- K-Means
- Agglomerative
  - Single
  - Complete
  - Average

Feature Selection:

- Backward Selecting
- Forward Selecting

Dataset:

We have 2 dataset,
(Cancer data set and Latitude and longitude data set)

Cancer dataset have been used for Classification and feature selecting methods

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean
0	1	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001
1	1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869
2	1	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974
3	1	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414
4	1	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980

5 rows × 32 columns

LatLog dataset have been used for the Clustering methods

	Longitude	Latitude
0	36.861544	-5.177747
1	51.463766	5.392935
2	51.190492	4.453765
3	51.326247	6.085953
4	51.463766	5.392935

Breast Cancer Data

Breast cancer (BC) is one of the most common cancers among women worldwide, representing the majority of new cancer cases and cancer-related deaths according to global statistics, making it a significant public health problem in today's society.

Classification and data mining methods are an effective way to classify data. Especially in medical field, where those methods are widely used in diagnosis and analysis to make decisions.

Recommended Screening Guidelines:

Mammography. The most important screening test for breast cancer is the mammogram. A mammogram is an X-ray of the breast. It can detect breast cancer up to two years before the tumor can be felt by you or your doctor.

Women age 40–45 or older who are at average risk of breast cancer should have a mammogram once a year.

Women at high risk should have yearly mammograms along with an MRI starting at age 30.

Data Preparation

Attribute Information:

- ID number 2 Diagnosis (M = malignant, B = benign) 3–32)

Ten real-valued features are computed for each cell nucleus:

- **radius** (mean of distances from center to points on the perimeter)
- **texture** (standard deviation of gray-scale values)
- **perimeter**
- **area**
- **smoothness** (local variation in radius lengths)
- **compactness** (perimeter² / area — 1.0)
- **concavity** (severity of concave portions of the contour)
- **concave points** (number of concave portions of the contour)
- **symmetry**
- **fractal dimension** (“coastline approximation” — 1)

The mean, standard error and “worst” or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius.

This analysis aims to observe which features are most helpful in predicting malignant or benign cancer and to see general trends that may aid us in model selection and hyper parameter selection. The goal is to classify whether the breast cancer is benign or malignant. To achieve this i have used machine learning classification methods to fit a function that can predict the discrete class of new input.

Visualization of data is an imperative aspect of data science. It helps to understand data and also to explain the data to another person.
Python has several interesting visualization libraries such as Matplotlib, etc.

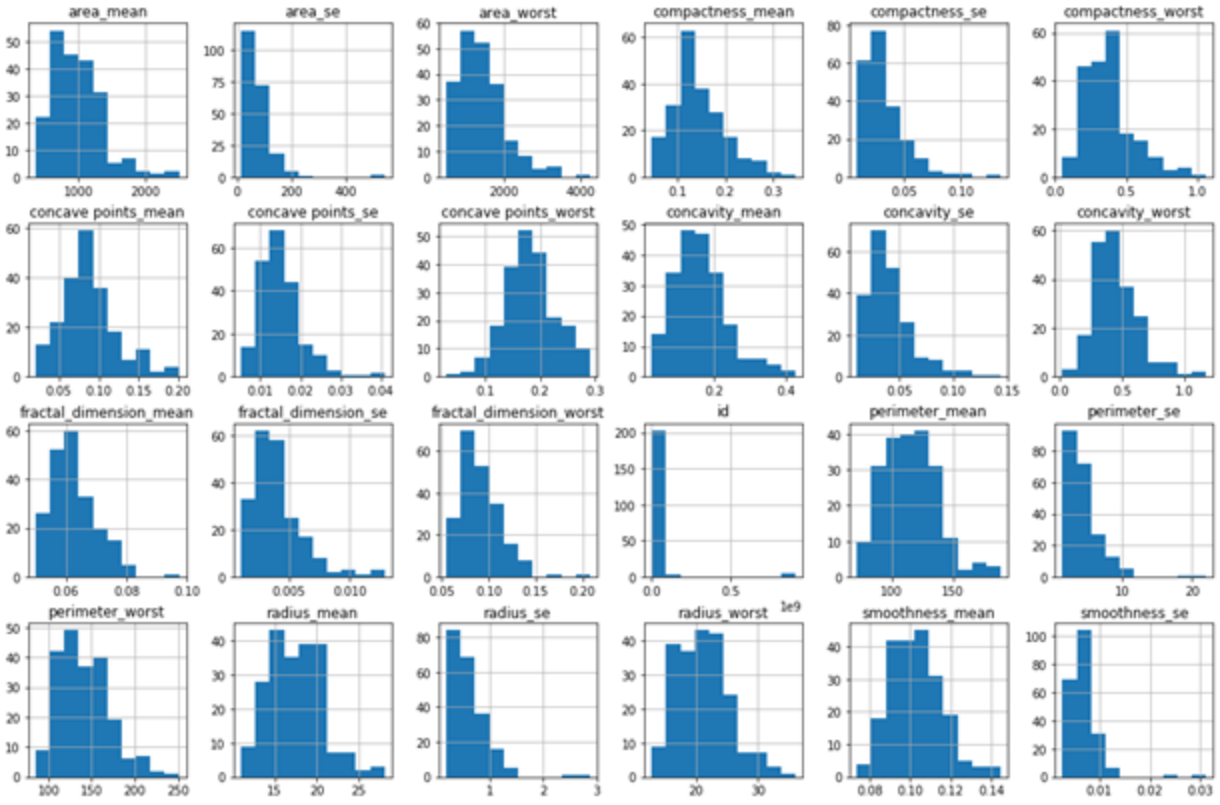
Missing or Null Data points

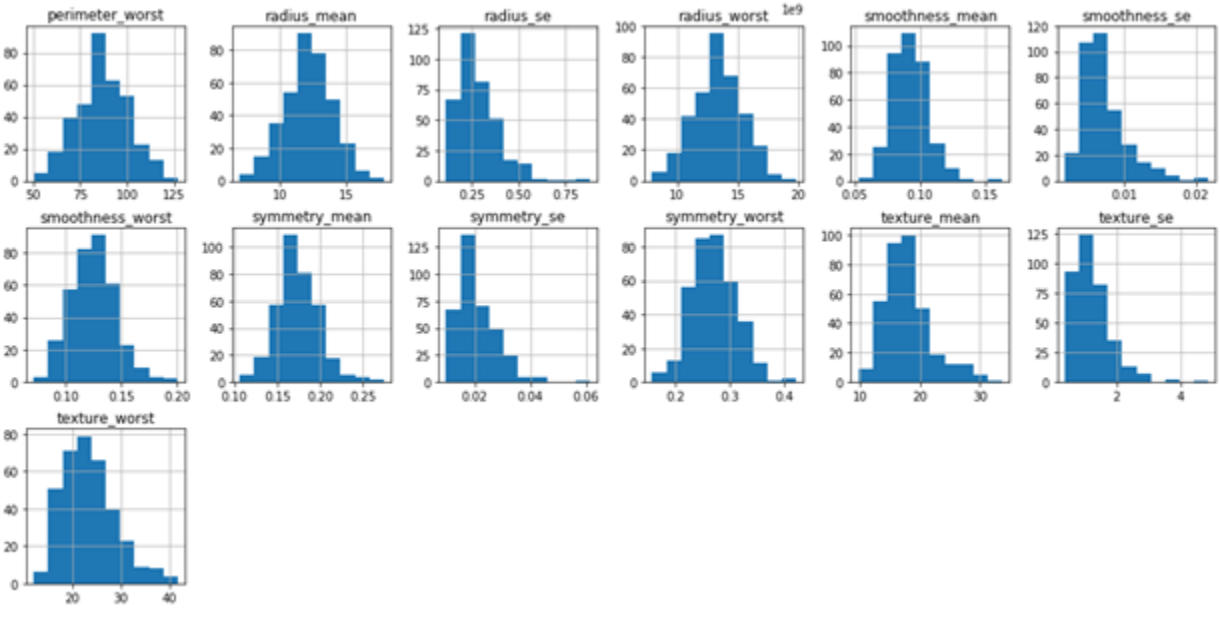
We can find any missing or null data points of the data set (if there is any) using the following pandas function.

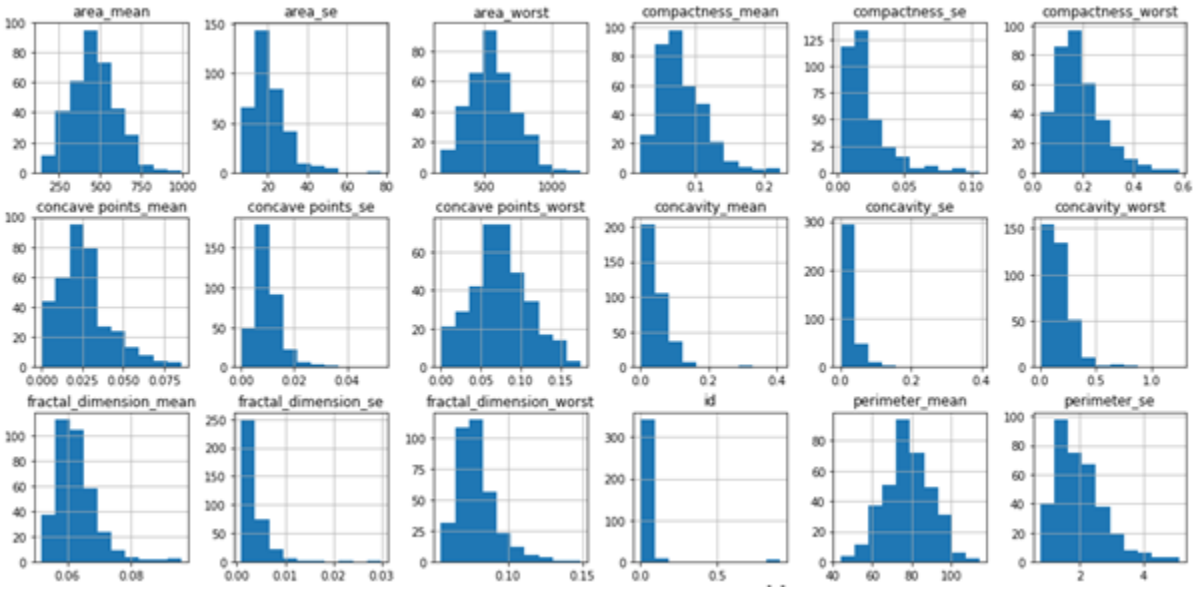
```
dataset.isnull().sum()
dataset.isna().sum()
```

```
diagnosis
B    357
M    212
dtype: int64
```

I have created new file calling `data-cancer-numerical.csv` that is all numerical and deleted NaaN values

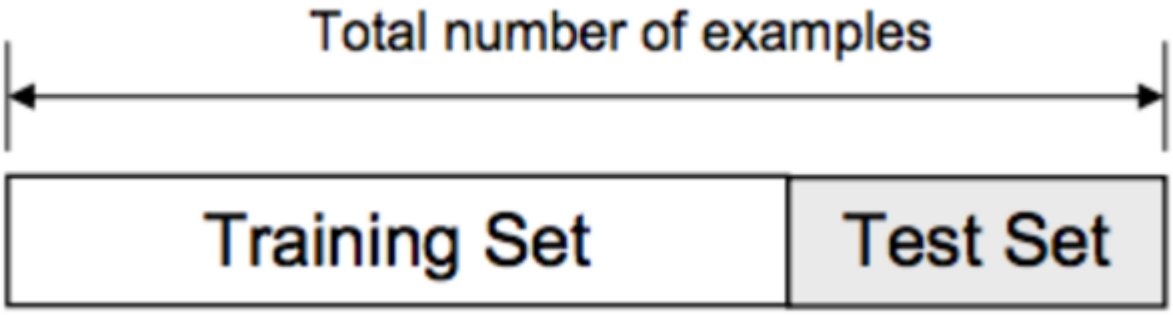

area_mean
area_se
area_worst
compactness_mean
compactness_se
compactness_worst
concave points_mean
concave points_se
concave points_worst
concavity_mean
concavity_se
concavity_worst
fractal_dimension_mean
fractal_dimension_se
fractal_dimension_worst
id
perimeter_mean
perimeter_se
perimeter_worst
radius_mean
radius_se
radius_worst
1e9
smoothness_mean
smoothness_se


perimeter_worst
radius_mean
radius_se
radius_worst
1e9
smoothness_mean
smoothness_se
smoothness_worst
symmetry_mean
symmetry_se
symmetry_worst
texture_mean
texture_se
texture_worst


area_mean
area_se
area_worst
compactness_mean
compactness_se
compactness_worst
concave points_mean
concave points_se
concave points_worst
concavity_mean
concavity_se
concavity_worst
fractal_dimension_mean
fractal_dimension_se
fractal_dimension_worst
id
perimeter_mean
perimeter_se

We can observe that the data set contain 569 rows and 32 columns. '*Diagnosis*' is the column which we are going to predict , which says if the cancer is M = malignant or B = benign. 1 means the cancer is malignant and 0 means benign. We can identify that out of the 569 persons, 357 are labeled as B (benign) and 212 as M (malignant).

Splitting the dataset



The data we use is usually split into training data and test data. The training set contains a known output and the model learns on this data in order to be generalized to other data later on. We have the test dataset (or subset) in order to test our model's prediction on this subset.

Feature Scaling

Most of the times, your dataset will contain features highly varying in magnitudes, units and range. But since, most of the machine learning algorithms use Euclidean distance between two data points in their computations. We need to bring all features to the same level of magnitudes. This can be achieved by scaling. This means that you're transforming your data so that it fits within a specific scale, like 0–100 or 0–1.
We will use StandardScaler method from SciKit-Learn library.

At first run, I have not used feature scaling for normalizing the data

I will compare the difference between two modeling in the related section.

Model Selection

This is the most exciting phase in Applying Machine Learning to any Dataset.
It is also known as Algorithm selection for Predicting the best results.

You can view each algorithm through the related method in files

The required description about the code has been described on codebase.

Scoring

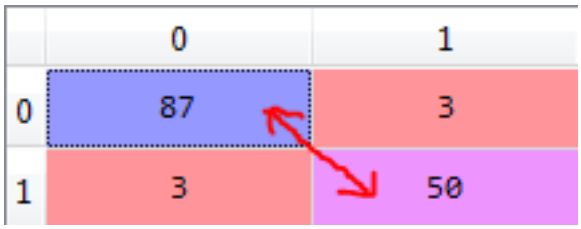
To check the accuracy we need to import confusion_matrix method of metrics class. The confusion matrix is a way of tabulating the number of mis-classifications, i.e., the number of predicted classes which ended up in a wrong classification bin based on the true classes.

We will use Classification Accuracy method to find the accuracy of our models.
Classification Accuracy is what we usually mean, when we use the term accuracy. It is the ratio of number of correct predictions to the total number of input samples.

Fig: Accuracy

$$Accuracy = \frac{Number\ of\ Correct\ predictions}{Total\ number\ of\ predictions\ made}$$

Fig: Confusion Matrix

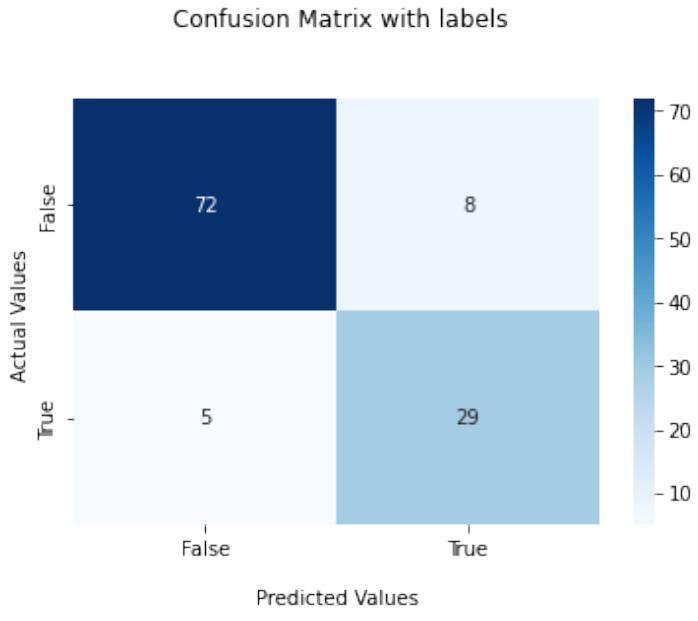


After applying the different classification models, we have got below accuracies with different models:

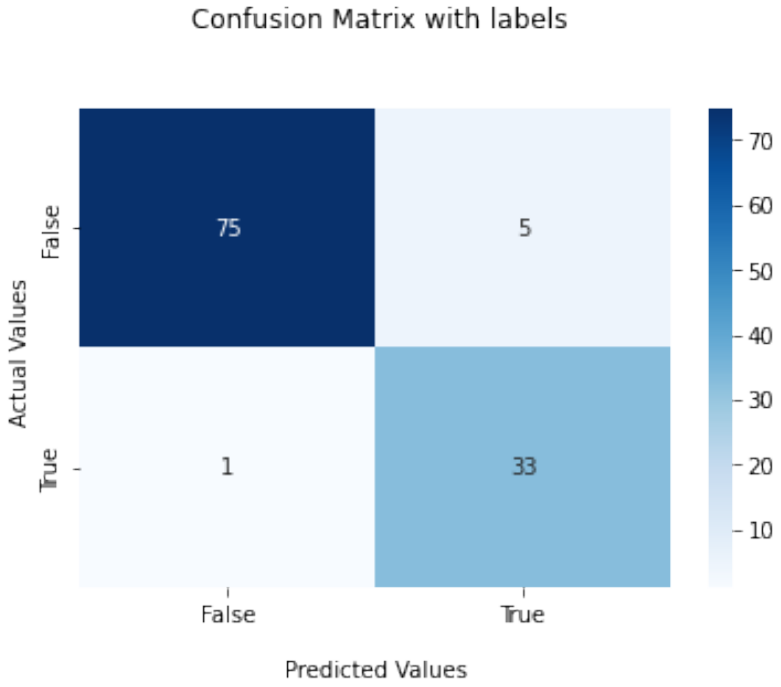
KNN:

I have been modeling my data under difference conditions

1.Using whole data with no Normalization — 89.5%



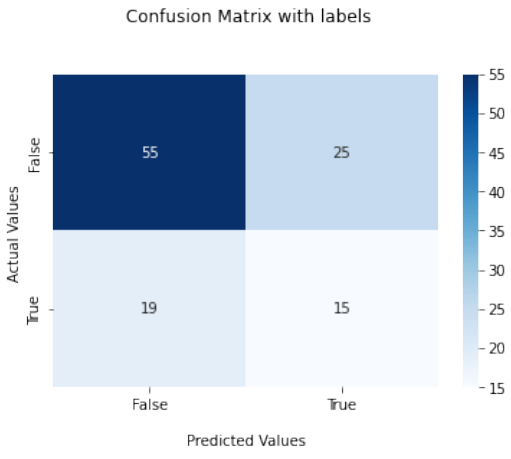
2. Using whole data with Normalization — 97%
3. Using Reduced dimensions data with Normalization — 97%



Naive Bayes:

accuracy — 92.1 %

confusion matrix :

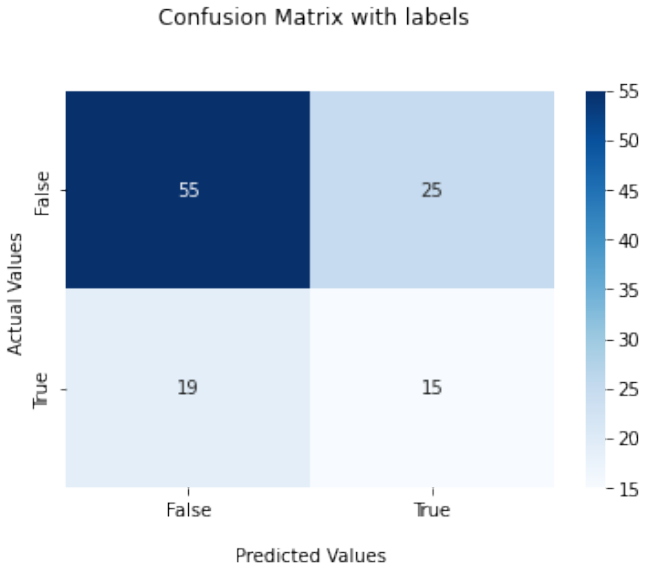


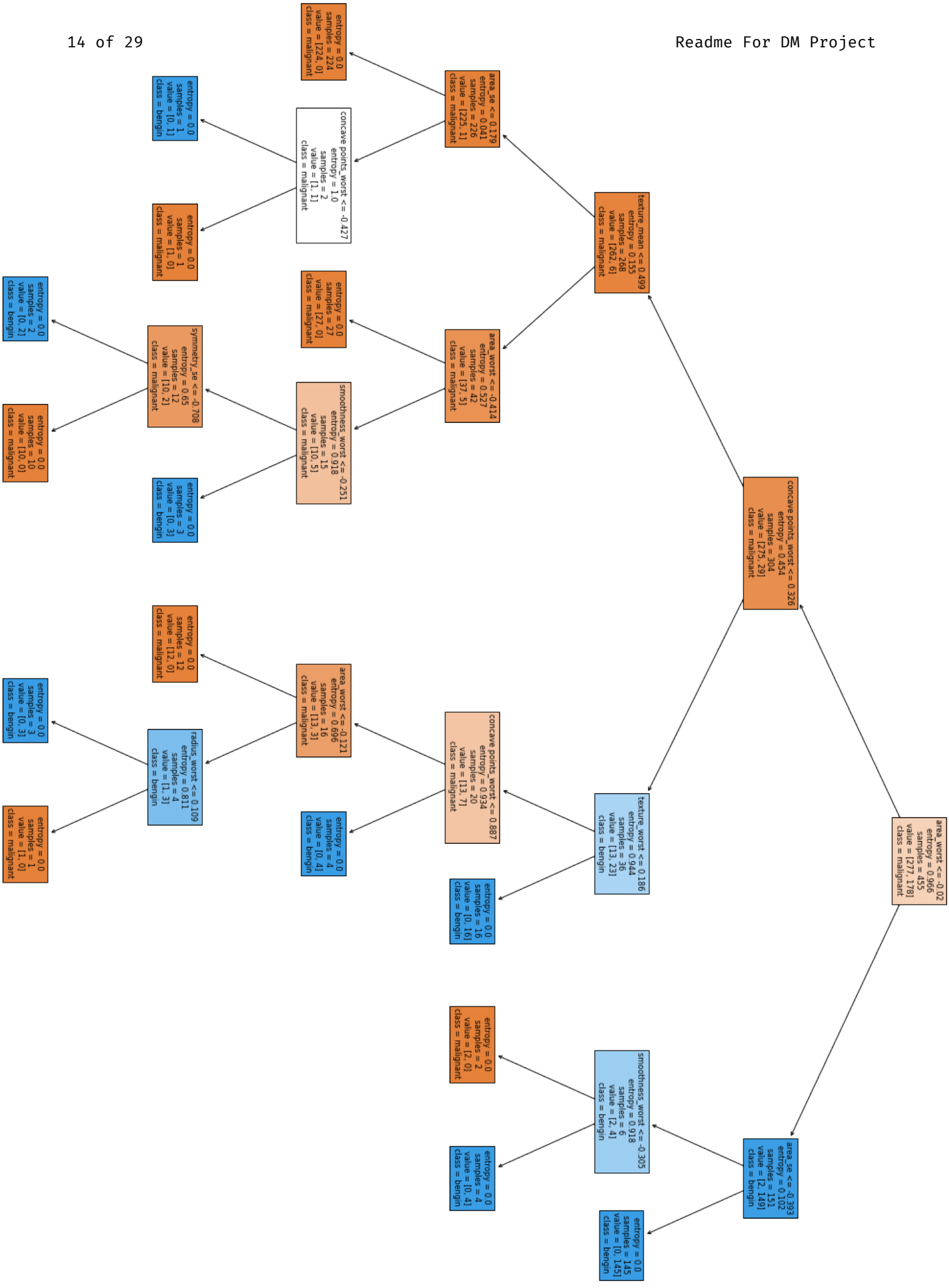
Decision Tree (D-Tree):

Here the calculated D-tree schema using ID3 method :

accuracy — 91 %

Confusion Matrix:




concave points_worst <= 0.326
entropy = 0.454
samples = 304
value = [275, 29]
class = malignant
texture_mean <= 0.499
entropy = 0.155
samples = 268
value = [262, 6]
class = malignant
area_se <= 0.179
entropy = 0.041
samples = 226
value = [225, 1]
class = malignant
entropy = 0.0
samples = 224
value = [224, 0]
class = malignant
concave points_worst <= -0.427
entropy = 1.0
samples = 2
value = [1, 1]
class = malignant
entropy = 0.0
samples = 1
value = [0, 1]
class = benign
entropy = 0.0
samples = 1
value = [1, 0]
class = malignant
area_worst <= -0.414
entropy = 0.527
samples = 42
value = [37, 5]
class = malignant
entropy = 0.0
samples = 27
value = [27, 0]
class = malignant
smoothness_worst <= -0.251
entropy = 0.918
samples = 15
value = [10, 5]
class = malignant
symmetry_se <= -0.708
entropy = 0.65
samples = 12
value = [10, 2]
class = malignant
entropy = 0.0
samples = 2
value = [0, 2]
class = benign
entropy = 0.0
samples = 10
value = [10, 0]
class = malignant
entropy = 0.0
samples = 3
value = [0, 3]
class = benign
texture_worst <= 0.186
entropy = 0.544
samples = 36
value = [13, 23]
class = benign
concave points_worst <= 0.887
entropy = 0.934
samples = 20
value = [13, 7]
class = malignant
entropy = 0.0
samples = 16
value = [0, 16]
class = benign
area_worst <= -0.121
entropy = 0.696
samples = 16
value = [13, 3]
class = malignant
entropy = 0.0
samples = 12
value = [12, 0]
class = malignant
radius_worst <= 0.109
entropy = 0.811
samples = 4
value = [1, 3]
class = benign
entropy = 0.0
samples = 3
value = [0, 3]
class = benign
entropy = 0.0
samples = 1
value = [1, 0]
class = malignant
entropy = 0.0
samples = 4
value = [0, 4]
class = benign
area_worst <= -0.02
entropy = 0.966
samples = 455
value = [277, 178]
class = malignant
area_se <= -0.393
entropy = 0.102
samples = 151
value = [2, 149]
class = benign
smoothness_worst <= -0.305
entropy = 0.918
samples = 6
value = [2, 4]
class = benign
entropy = 0.0
samples = 2
value = [2, 0]
class = malignant
entropy = 0.0
samples = 4
value = [0, 4]
class = benign
entropy = 0.0
samples = 145
value = [0, 145]
class = benign

If we compare D-Tree and the Bayes results, we can see the similar results on them.

Random Forest:

The result of this methods surprised me.
I have been expected that the random forest perform better result compare with the D-Tree

But unfortunately we do not have get acceptable results from this method!

for any bootstrap subclass of the Dataset our Y-hat will be always :

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0]
```

So the votes between them is always be

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0]
```

Accuracy — 18%

Conclusion:

with comparing the accuracy between discussed classified methods I believe the KNN is our ideal method for this dataset with 97% accuracy and also the logic is not to complicated.

Feature Selection For Cancer data:

Feature selection is **the process of reducing the number of input variables when developing a predictive model**. It is desirable to reduce the number of input variables to both reduce the computational cost of modeling and, in some cases, to improve the performance of the model

—Google

Goal

The goal of feature selection in machine learning is to find the best set of features that allows one to build useful models of studied phenomena.

The techniques for feature selection in machine learning can be broadly classified into the following categories:

```
'radius_mean','texture_mean','area_mean','smoothness_mean',
'compactness_mean','concavity_mean','concave points_mean',
'symmetry_mean','fractal_dimension_mean',
'radius_se','texture_se','area_se','smoothness_se',
'compactness_se','concavity_se','concave points_se',
'symmetry_se','fractal_dimension_se',
'radius_worst','texture_worst','area_worst','smoothness_worst',
'compactness_worst','concavity_worst','concave points_worst',
'symmetry_worst','fractal_dimension_worst',
```

I have tried two different feature selecting:

- **Forward Feature Selecting**
- **Backward Feature Selecting**

And after generating the feature, I have used them on

different classification method

The results will describe later.

Forward Feature Selecting

because of the enormous size of the data

I gather the logs in `feature_slecting/logs/fs_logs.txt`

```
Final Variables: ['intercept', 'concave points_worst',
'radius_worst', 'texture_worst', 'area_worst',
'smoothness_se', 'symmetry_worst', 'compactness_se',
'radius_se', 'fractal_dimension_worst',
'compactness_mean', 'concave points_mean',
'concavity_worst', 'concavity_se', 'area_se']
```

Backward Feature Selecting

because of the enormous size of the data

I gather the logs in `feature_slecting/logs/bs_logs.txt`

```
Final Variables: ['intercept', 'radius_mean',
'compactness_mean', 'concavity_mean', 'concave
points_mean', 'radius_se', 'smoothness_se',
'concavity_se', 'concave points_se', 'radius_worst',
'texture_worst', 'area_worst', 'concavity_worst',
'symmetry_worst', ‘fractal_dimension_worst']
```

So I use theese features as my X in the classification methods(K-NN,Bayes,D-Tree) and there is no specific improvement of our model.

I believe the cause of this result is for the size of features, and the model is use all of the feature as useable input.

Longitude and Latitude Data

What type of problem clustering can be solved?

Clustering algorithms are an effective Machine Learning (ML) technique for unsupervised data (unlabeled data). The most popular algorithms for ML are `K–Means` clustering. This algorithm is extremely efficient when applied to many ML problems.

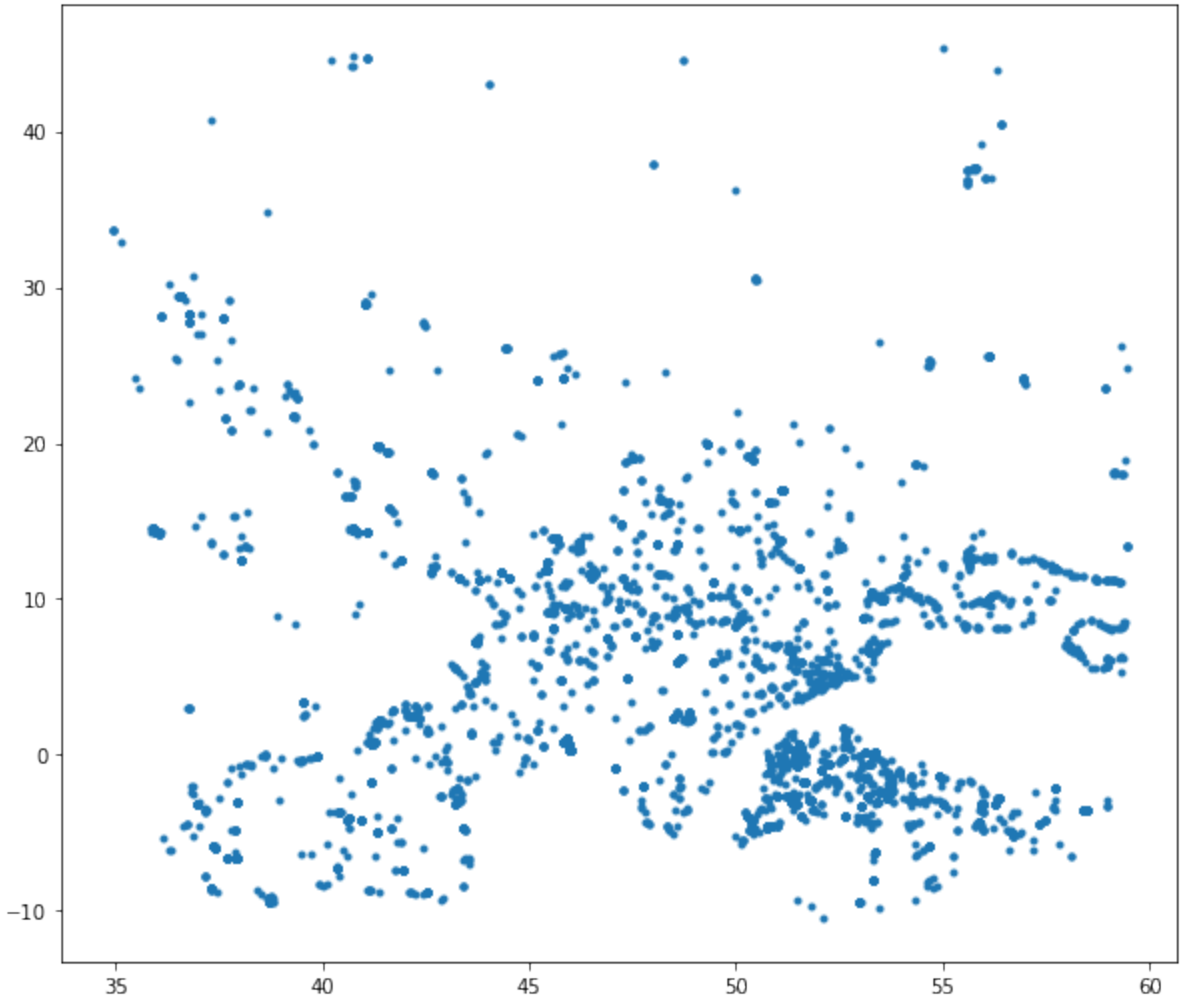
The `K–Means` clustering has been applied to different scenarios in many different problems area, such as:

- **Information Technology:** used to identify the spam filter, classify network traffic, and identify fraudulent or criminal activity.
- **Marketing:** used to characterize & discover customer segments for marketing purposes.
- **Biology:** used for classification among different species of plants and animals.
- **Insurance:** used to acknowledge the customers, their policies and identifying the frauds.

Clustering is not an easy job to perform. Indeed, in such a setting, data crunching and discovery are often motivated by domain knowledge, if not pure based on experience, and made difficult as there is no way to test the precision of the resulting segmentation.

Data Preparation

Here we have csv file that contains Longitude and Latitude Data for each point and there is no broken or missing (NaaN) data in there.



So let's visualize our data for better understanding for it:

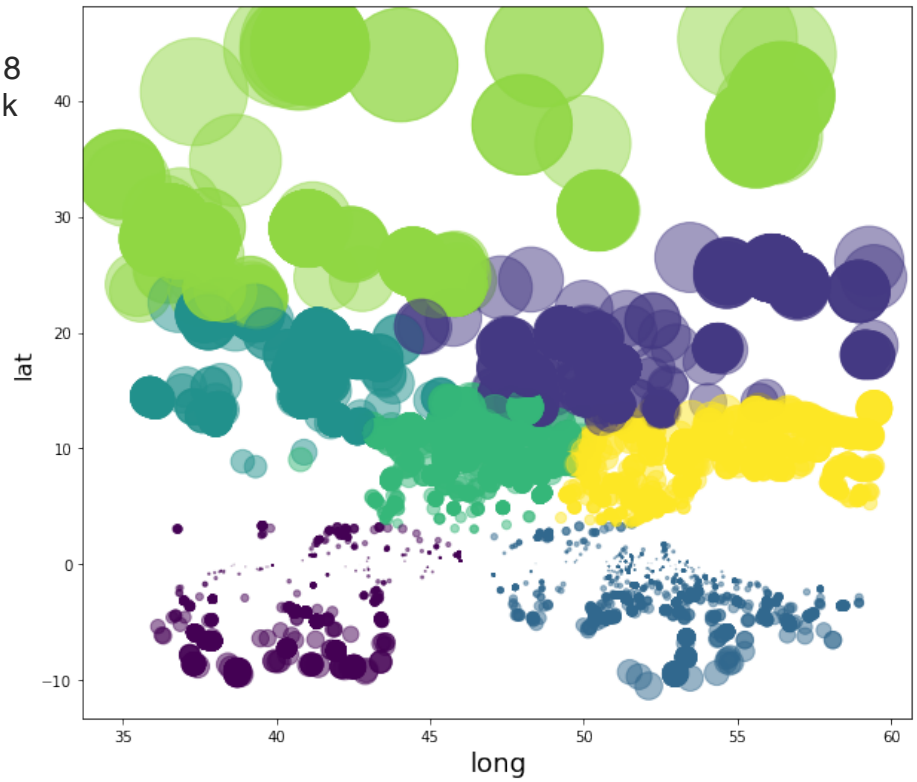
Here we have a good variety of data and can cluster them using the clustering methods

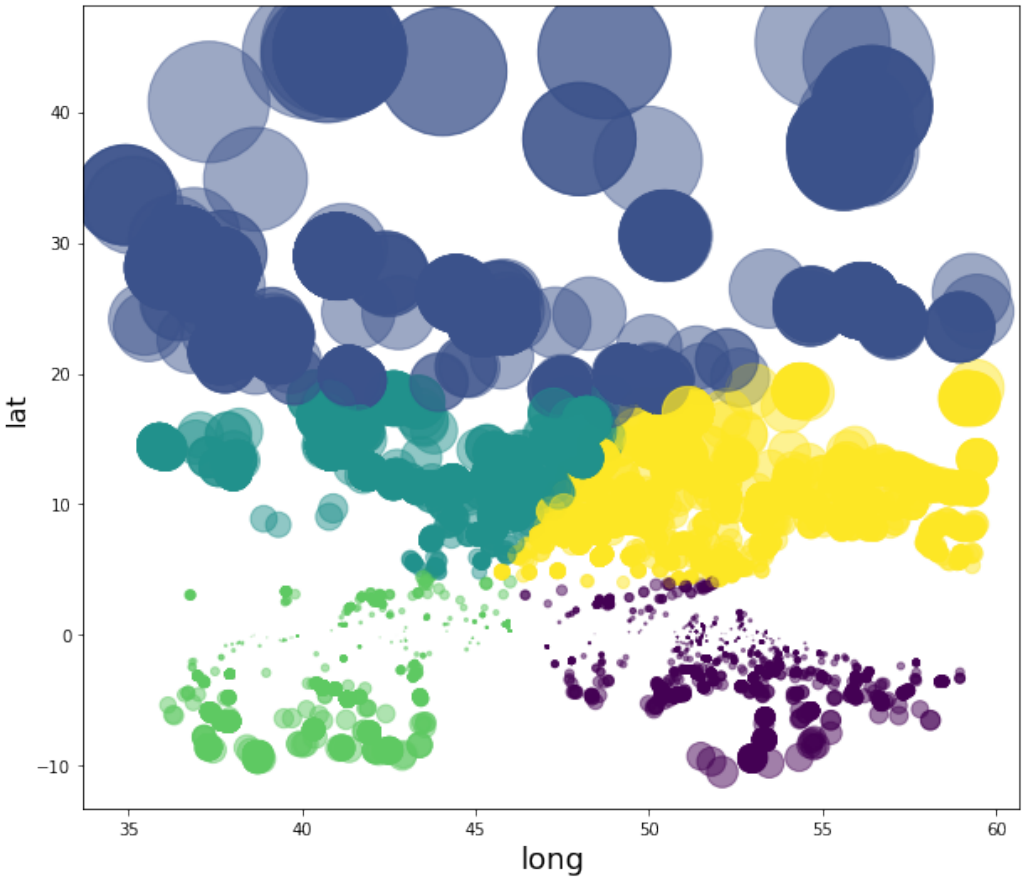
Model Selection

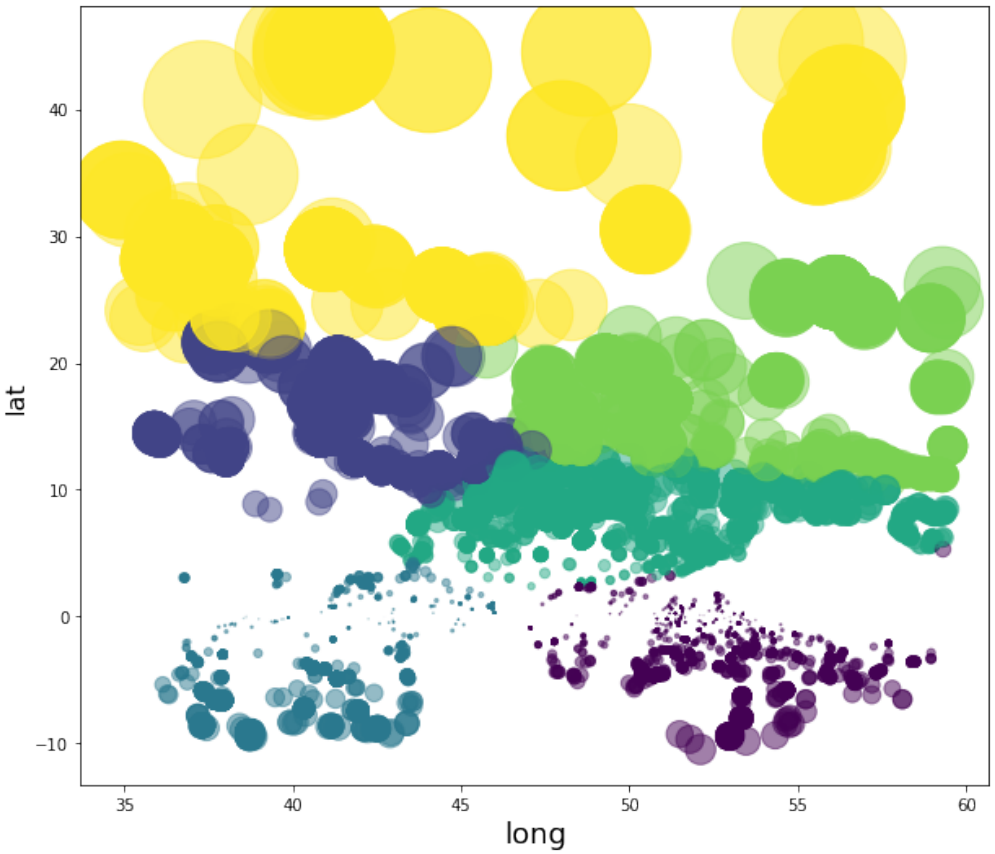
K-means:

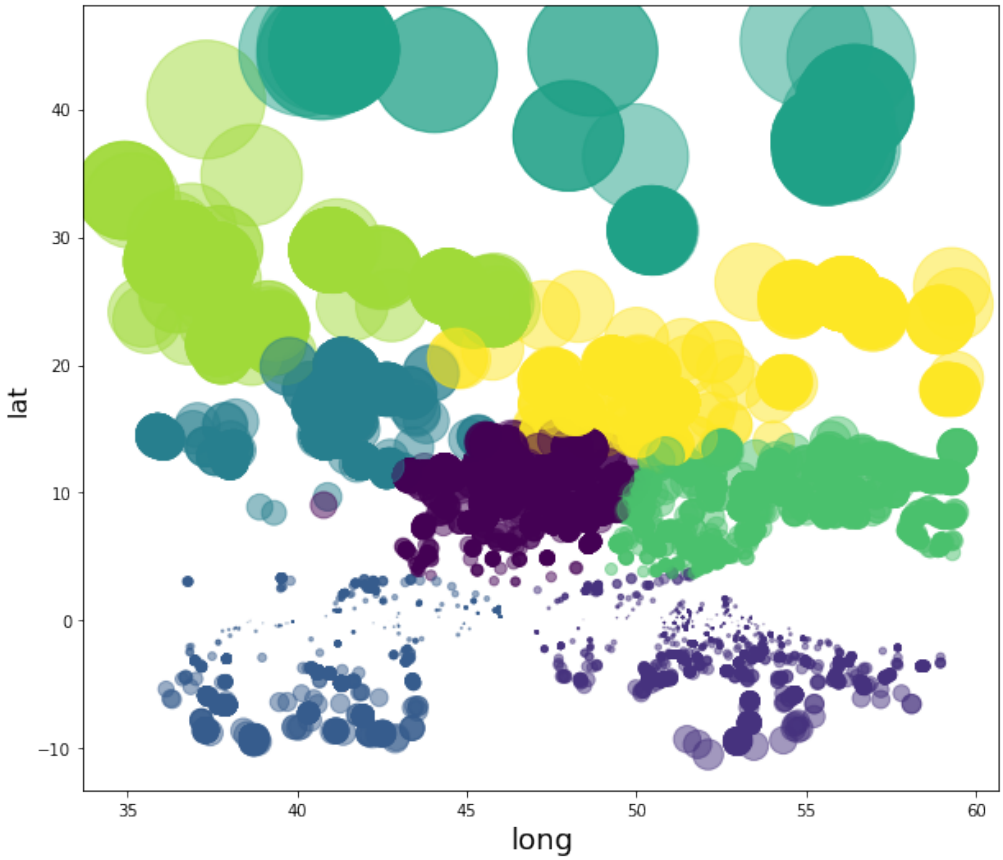
First of all I used sklearn K-means builtin method so I could have a sample to compare my own written method with:

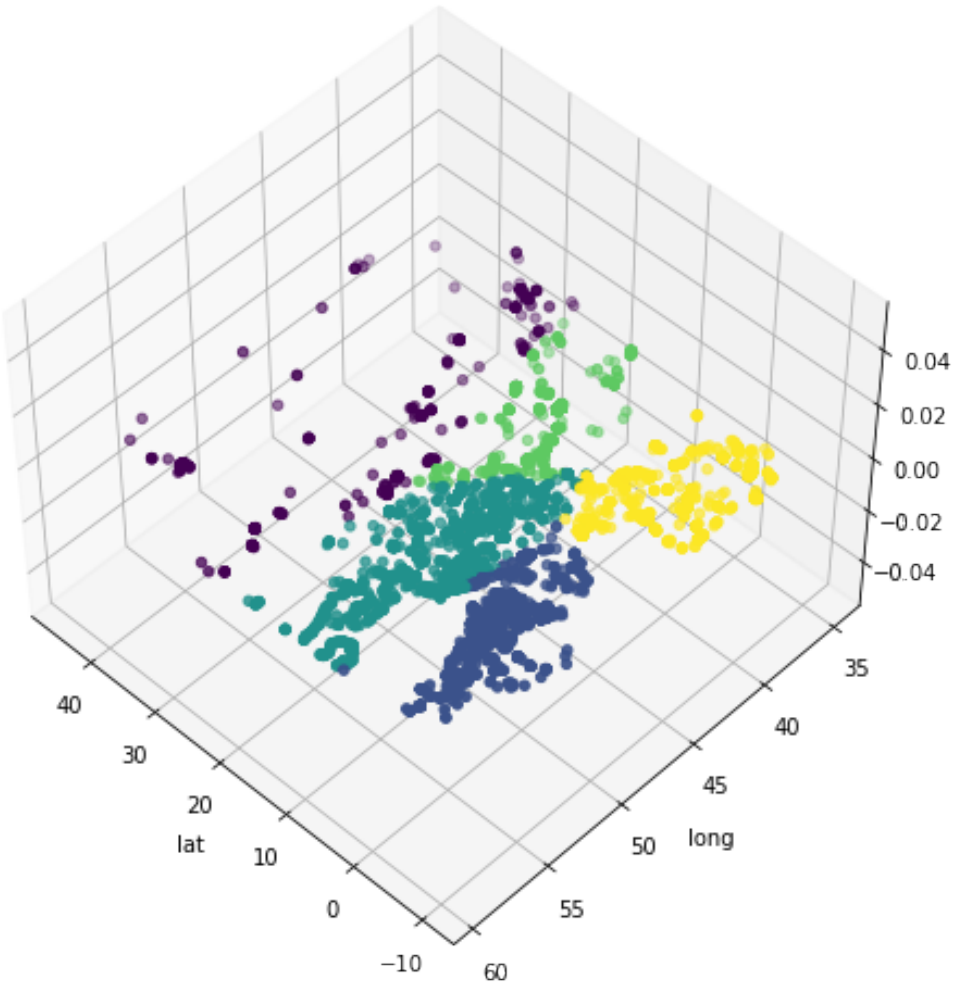
I start my k with 4 to 8
So each plot for this k
is as bellow:






lat
long
40
30
20
10
0
−10
35
40
45
50
55
60

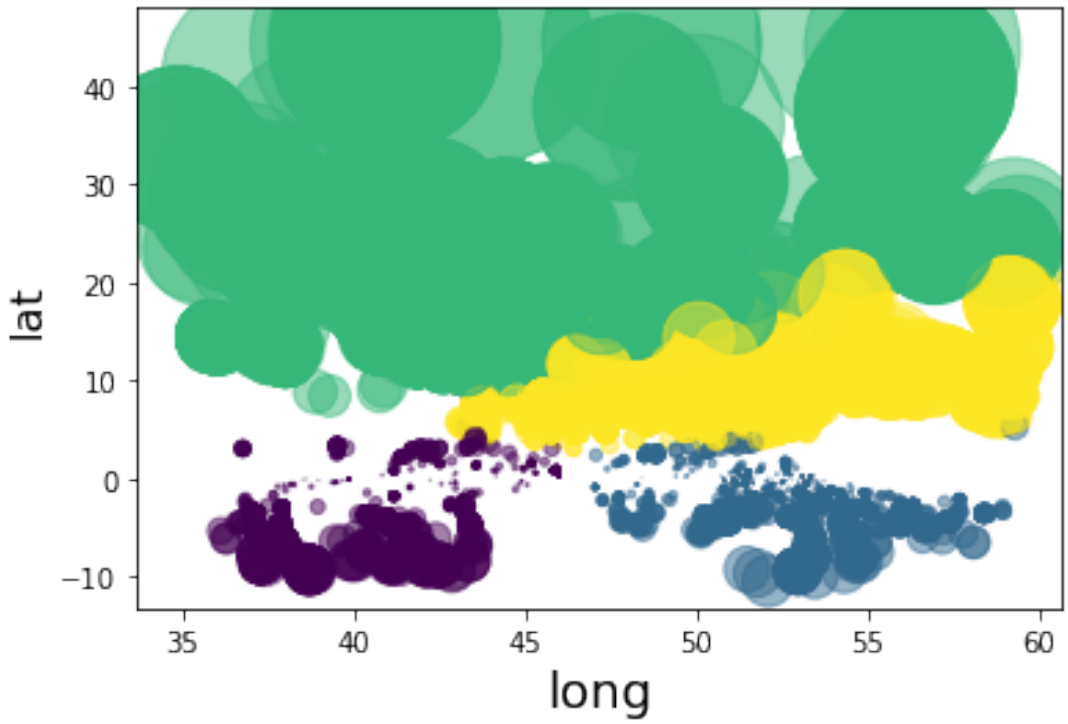

lat
long
40
30
20
10
0
−10
35
40
45
50
55
60



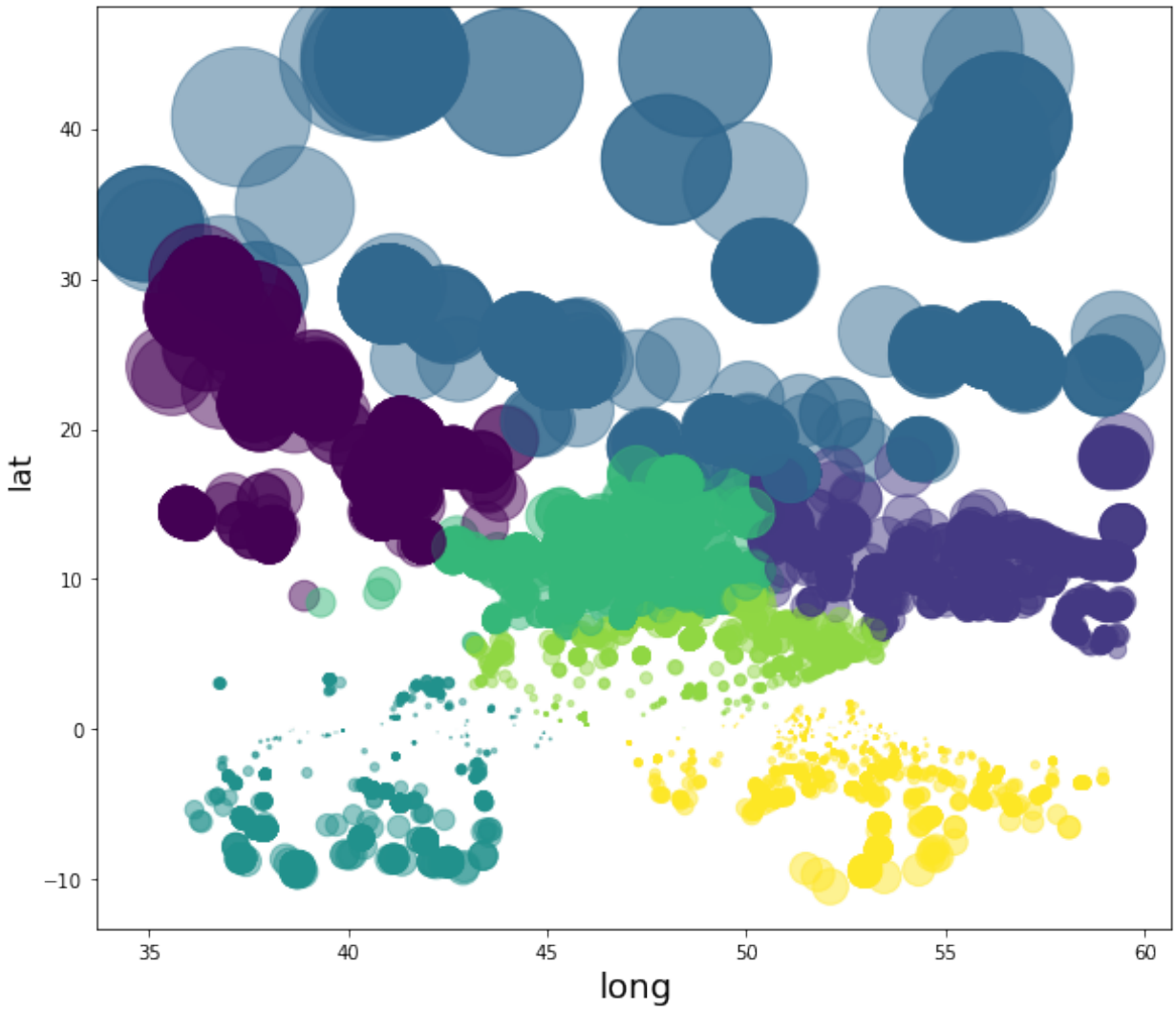
To determine the best K for your model usually the domain expert tell the number

Now let’s compare our own created algorithm with different Ks:

k = 4

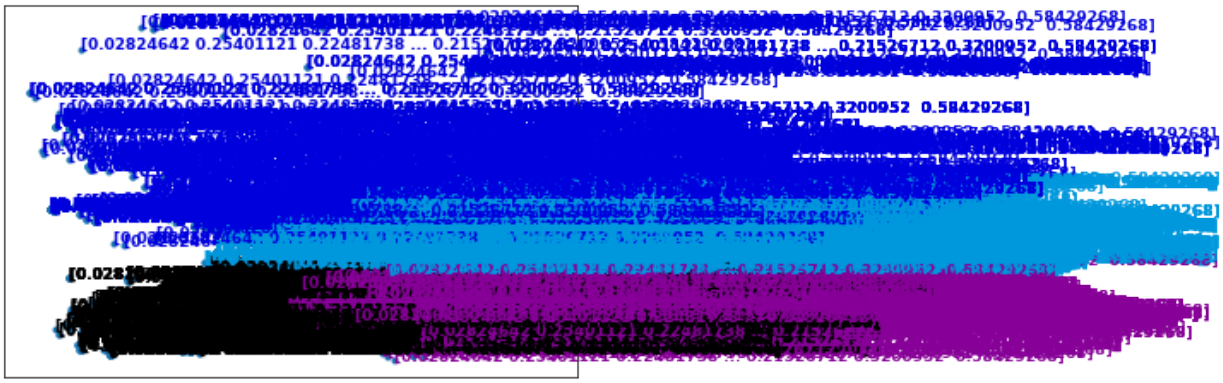

40
30
20
10
0
−10
lat
35
40
45
50
55
60
long

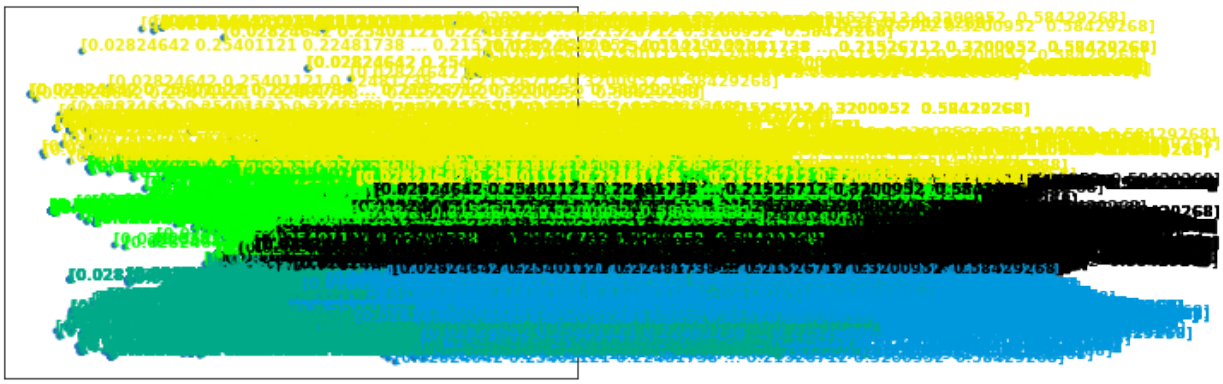
k=7


40
30
20
10
0
−10
lat
35
40
45
50
55
60
long

Comparing with generated sklearn clusters, we can see here we have pretty good clustered data that the ideal K number should determine from the domain expert again.

And here another type of plot we have for the clusters.





Agglomerative methods:

Hierarchical Clustering - Agglomerative¶

We will be looking at a clustering technique, which is **Agglomerative Hierarchical Clustering**. Remember that agglomerative is the bottom up approach.

In this lab, we will be looking at Agglomerative clustering, which is more popular than Divisive clustering.

We will also be using Complete Linkage as the Linkage Criteria.

NOTE: You can also try using Average Linkage wherever Complete Linkage would be used to see the difference!

NOTE2:

Because the agglomerative methods contains a lot of mathematical processes,
I have used cython inside my agglomerative method.

so Be sure you installed the cython before the running.

Here I compare two complete and average agglomerative method and there is not much difference between the generated dendrogram figures.

The situation for the simple linkage is similar to other ones.

Fig: Complete Linkage

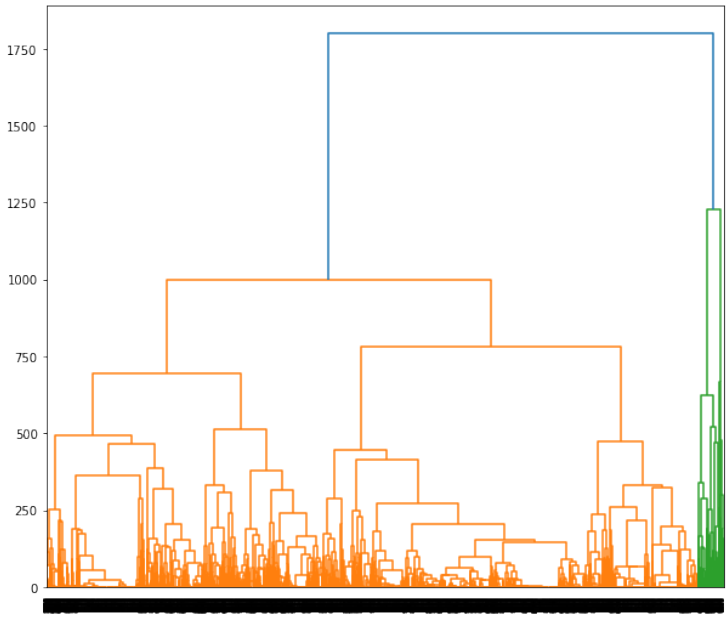
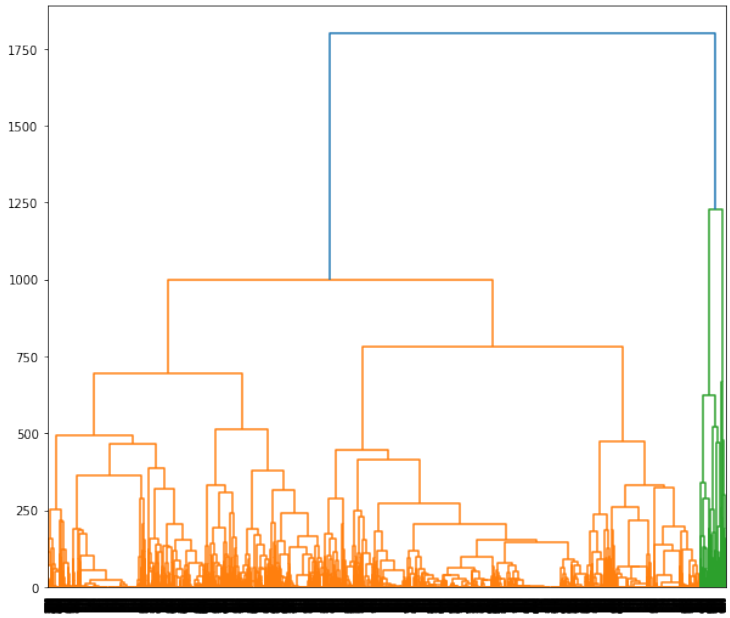


Fig: Average Linkage



I have created reduced generated dendrogram for this agglomerative method

The result it as bellow: