Final Project Report

Abstract

During the course, we have learned different subjects when each topic was based on the previous one. We have learned – Data structures and algorithms, Data analysis and visualization using 'Numpy' and 'Pandas', and at last Machine Learning.

In this final project, we got three datasets that contains information about covid19 - the recovered, confirmed and deaths.

In the first section, we explore the databases. First, we import the data from a CSV file to Python and explore the data in statistical aspects using Pandas library which helps us find interesting information about the datasets. Then we proceeded to more advanced analysis and used more advanced functions to learn more meaningful information about our data, do advanced analytics, and present it with visualization so we could see the results more clearly and see the data behavior. In order to use the machine learning and deep learning algorithm, we performed pre-processing to convert the raw data to a sustainable format for analysis. Then, we perform a re-analysis of the data in order to draw new conclusions about the relevant information.

In the last section, we used the Dataset after pre-processing and built a 'Kmeans' model for the unsupervised learning algorithm and a Naive Bayes model for the supervised learning algorithm. We used our previous knowledge to arrange the data to better fit the model and presented the model result in a graph to show the results more clearly.

The question for the supervised learning is: Can we predict a continent by the numbers of confirmed/deaths/recovered cases during the time?

The question for the unsupervised learning is: Can we split our data to N clusters such that every cluster is including countries from the same continent or the same geographic region?

Method

First exploring

In the beginning of exploring those 3 data frames we built a dictionary which contains them, this dictionary will help us to run a few functions easily and with better visibility. The first thing we explored was how many rows and columns there are. Except for the recovered table which has 253 rows, the others two data frames have 263 rows. all data frames have the same number of columns – 153.

We checked the names of the columns and we found out that most of the column's names contain dates with the cumulative numbers of confirmed, death and recovered cases. The dates start at 1.22.2020 and end at 6.18.2020. The other columns contain the country's names, the province of the country (if exists), the latitude and the longitude.

After that, we used 'describe' function to help us learn more about the statistic of the table, it was not so helpful because of the cumulative data in the dates. We built "heatmap" about the correlation between a few variables. We built a new table which contains the sum of confirmed, death, recovered and active cases of the "Covid-19". We also added the latitude and the longitude. We will choose to expand to talk about the correlation between the confirmed cases to the deaths and recovered. We saw a positive correlation between confirmed to recovered (0.53) which is very understandable. However, the correlation between confirmed to deaths was even higher (0.93) in comparison to the correlation between the recovered to confirmed cases. Those results must concern us as humans.

We also built three "Donut" graphs about the numbers of confirmed, deaths and recovered cases in each month. We multiplied the months "January" and "June" to create an equal ratio between each month, because the data does not contain the whole details of those two months. We could learn from those graphs that April was the month with the biggest number of deaths and June was the month with the biggest number of recovered, these results show us that the peak of the pandemic is behind us.

At the end of this part, we tried to find any special distribution in each table (in this part of the project, the numbers are cumulative). From the confirmed table, we can learn that the ascending of the cases was steady with a little bit more ascending in April. Moreover, we could see in the deaths graph that the ascending in April is sharper than we saw in the confirmed table with a steady ascending later. The recovered distribution along the dates is steady ascending.

In conclusion, the first exploring did not provide us enough information. Therefore, we thought about adding a new column. After a deep thought, we decided to add a continent column which can help us to get more information about the corona's spread around the world.

Pre-processing

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a format that is feasible for the analysis.

Different data preprocessing techniques:
First, the step of our data preprocessing is to handle missing data in the datasets. If our dataset contains some missing data, then it may create a huge problem for our machine learning model. Hence it is necessary to handle missing values present in the dataset. We use Pandas library to check the data with the function "isnull" and by the function we built, we delete the specific columns which consist of above 100 null values in the column in each dataset. With this technique we found that we need to remove the column Province/State. The next thing we took care of was turning the cumulative data we received into non-cumulative data in order to have full transparency for each day and making our data more reliable. Then, because the information about the country may cause over-fitting because there are a lot of countries, we have added a new column which according to the column in the country will indicate which continent it is on. In this way, we can get a broader picture and draw more conclusions about the continents. At the end of the process, we deleted columns Lat, Long and Country/Region that were not relevant to our data analysis.

Diagnosis between Supervised Learning and Unsupervised Learning: For Supervised Learning, we used the Sklearn library and the model preprocessing that have a label Encoder object to convert categorical values to numerical. In this algorithm, we classify the data and therefore should know how many times the value has appeared in each column of a continent.

For unsupervised learning we use the Pandas library to convert categorical values to indicators - one hot vector value. In this algorithm, we clustered the data so we need to know the distance of each value in order to group it according to the number of groups to be defined.

Second exploring

In this part of the assignment, we did additional statistical analyzes for the data sets to see if the preprocessing process actually affected the data as we expected.

For statistical analysis, we used the original data sets and used the data sets after the preprocessing but without matching them to machine learning algorithms (without one hot vector and encoders). First, we built a function that would describe the data statistically. We then examined for each data set the distribution of the confirmed, deaths and recovered from the corona according to all the days given in the data sets. The next function built is 'data_for_each_continent'. This function receives the data after preprocessing as well as the original data from the 'corona_data_sets' class. Then The function print an output to the user of the total amount by each continent plus a pie chart describing the distribution of confirmed/ recovered/deaths by continents. After that, we used the 'value_counts ()' function from the 'pandas' library for examined how many countries belong to each continent. With this given, we built another graph of pie showing the same data distribution but this time, relative to the number of countries in each continent. The next function built to describe the data is 'corona_in_each_continent', this function gets a data frame and outputs a plot of the data by division to continents. The last function to describe the data is the 'plot_by_month' function, this function gets the data output of the function 'numbers_of_corona_by_dates', summarizes the data by division to months using the 'month' function, and outputs a plot for this description.

Naive Bayes

In this part of the assignment, we used a supervised algorithm from classification type. Our goal was to build a machine learning in order to predict, according to the daily data of the corona, to which continent each data matrix of a country belongs. For this purpose, one general function was built for all the data frames called ‘naive_bayes_algorithm’. The function gets a data frame, which separates the target column, ‘Continent’, from the rest of the table. The function starts with a loop that divides the data into a test group from 10% to 40% of the total data and the rest goes to the training group. In each iteration, the function calculates the accuracy measure for each test group size and saves the results in the ‘accuracy’ list. After all these calculations, the ‘naive_bayes_algorithm’ function proceeds to a loop that aims to send an output to the user about the size of the test group he should use. In addition, the function indicates the result of the best accuracy measure. The function continues to create a graph for the user in which it shows him the percentage of accuracy according to the size of the test group. Finally, the function sends an output to the user of a Confusion matrix graph according to the ideal size of the test group.

K-means

When we were asked to build an unsupervised algorithm for our data, we chose to use K-means algorithm. This kind of algorithm clusters every group by similar samples which we need to find out what
are those similar things and analyze them.

We choose to not normalize the data because the division of the clusters and the sum of square were the same and less informative when we normalized the data.

The first thing we did is to find out how many clusters do we need to divide into each data we used. In order to deal with this dilemma, We built a function which called ‘number_of_clusters()’. This function will return us two graphs. The first graph is called "Sum of square error ", in this graph we will see that as long as the number of clusters increases, our error decreases. now comes the question of what is the number of clusters that are "suspicious" to be optimal. For answering this question, we will use the "Elbow Method". In this method, when we see a break in some number clusters, we will take that number as an option about how many clusters should we use in each corona data frame. The second graph we will use is called "Silhouette", the silhouette is a measure about how good our algorithm work, his range is between -1 to 1. As long as the value is close to 1 we can say that the clustering was good. In our graph, we will wish to choose the number of clusters by the best silhouette we got. We need to find the best cluster by a combination of those two graphs, a number of clusters which has a good "Elbow break" and ones that has the best silhouette.

In the corona confirmed table we saw in "Sum of square error" graph that there are "Elbow Fracture" in 3,4,5,6 numbers of clusters. When we move on to the silhouette graph, we saw that the best silhouette between those four options is 3 that has a silhouette of 0.93 which is very good! In this case, we will divide this table to three clusters.

In the corona death table, we saw in the first graph sharp "Elbow Fracture" in 4 clusters, "Elbow break" a little bit less sharp in 3 clusters and small ones in 5 and 6 clusters. In the second graph, we saw that the silhouette of 3 and 4 clusters is equal. In conclusion, because both of their silhouettes are very high (more than 0.94) and the "Elbow break" of the 4 clusters was sharper than the 3 we will choose to divide the corona deaths table to 4 clusters.

In the corona recovers table, we saw in "Sum of square" graph sharp "Elbow break" in 4 clusters, "Elbow break" a little bit less sharp in 3 clusters and small ones in 5 and 6 clusters. When we moved on to the "Silhouette" graph we saw the number 3 is a bit higher than 4, but they are both very high (more than 0.9), so will choose to divide the corona recovers table to 4 clusters.

Conclusions

Second exploring

From the results of the graphs in this part, it can be concluded that the corona epidemic began relatively slowly in January and February, mainly in the Asian region, and from there to Europe and the rest of the world.

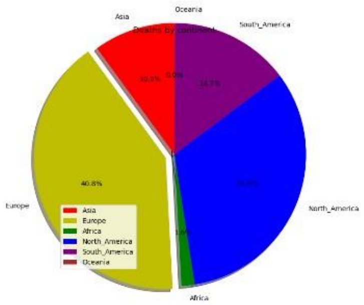
The initial conclusions from this part of the assignment are the statistical descriptions obtained from the function 'describe_data' and the function 'number _of_corona_by_dates'. According to the daily average of the countries and according to the graph, it can be seen from the data that the number of confirmed is rising rapidly starting in February. From the graph, it can be seen that in the confirmed data set, the curve flattens out in April, but from May onwards, the acceleration of virus infections begins again. As we know, in April most of the world was in quarantine so that makes sense. From the recovering graph, it can be concluded that the amount of recovering increases quite close to linear. This given also makes sense when, at the same time, the number of people infected increases. The graph of the dead behaves differently, it can be seen that the peak of the dead in the world is between the months of March and mid-April. The number of deaths is accelerating at the same time as the number of confirmed is accelerating. This given stems from the fact that the countries of the world did not know yet how to deal with the virus, which created a high amount of confirmed in the risk groups.

Another statistic that is interesting to discuss is the standard deviation of the data. While the standard deviation from the deaths data set is relatively minor and stable, the standard deviation in the table of the confirmed and recovered is quite high and dynamic. This figure is due to the fact that there are large gaps between countries in the quality of medicine (standard deviation in the recovered data set), large gaps in the population behaver and government's response to the epidemic (standard deviation in the confirmed data set) but the virus is fatal in all countries at the same level (standard deviation in the deaths data set).

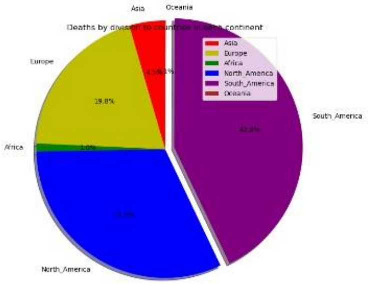
The results obtained from the ‘data_for_each_continent’ function shows that the continents that received the epidemic in the most severe way are Asia, Europe, and North America. But when examining these data by division to countries across the continent the data is changing drastically. It can be seen from the graph, after the division into countries on each continent that the 'slice of cake' of South America (in purple) is the largest. South American countries have dealt with the virus in the worst way compared to other countries in the world.

The graphs below show the difference after the division by sum of the countries.

Graph 1 - Deaths by continents



Graph 2 - Deaths by division to countries



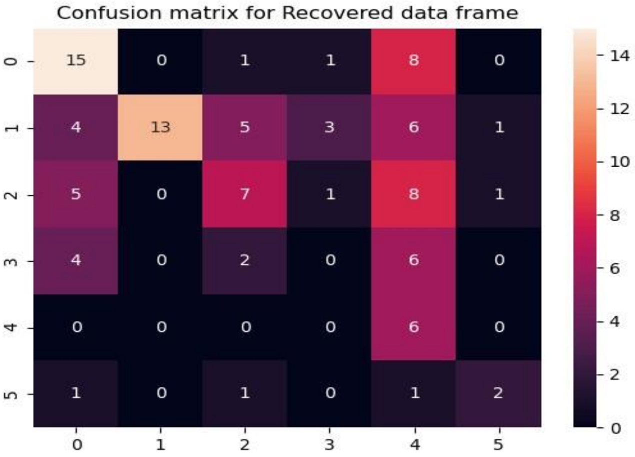
The data obtained from the functions 'corona_in_each_continent' and 'plots_by_month' are consistent with the data from the previous graphs. According to the graphs, the continents that have suffered the most are North America, Europe, and Asia. It can be seen, that recovering from the disease faster in Asia and Europe than in North America. The amount of deaths is much higher in Europe and North America than in Asia. In the graphs of the months, it can be seen that there is an acceleration in the numbers starting in March and there is a linear increase (considering that June is missing 12 days from the data). The graph of recovered is soon to a normal distribution. It can be concluded that its peak in April is due to the quarantine that has occurred all over the world in this month.

Naïve Bayes

For the confirmed data set, the division that will bring us the best accuracy is a test group of 10%. Although, we decided to choose a test group of 20%. The accuracy measure of both of them is pretty the same (display in the accuracy graph). So, to decrease the chance of overfitting we will prefer the test group will train with 80% of the data and not with 90%. The accuracy that results from this division is 40% accuracy. Relatively high percentage considering that the target column has 6 options of continents. without the algorithm, the prediction was 16.67% (1/6). From the Confusion matrix, it can be concluded that the algorithm identified most of the countries from Asia and Europe (1, 2). These continents are the most radically different data from the other continents.

The ideal distribution of the recovered data set is 40% for the test group. The accuracy that results from this division is 42% accuracy. This percentage is a higher percentage of accuracy than the confirmed data. It can be assumed that this given is due to the fact that this test group is significantly larger than the test group of the confirmed data. Therefore, the gaps between TP to FP and FN have increased. From the Confusion matrix, it can be learned that the identification of the countries of the African and Asian continents is particularly high. While there is serious confusion between the African continent and Oceania. In the graph of 'corona_in_each_continent' in the 'second statistical analysis' class, we saw that indeed these continents have relatively identical recovery numbers and this situation makes it difficult for the algorithm to identify.

Below is a Confusion matrix for Recovered data frame:


Confusion matrix for Recovered data frame

	0	1	2	3	4	5
0	15	0	1	1	8	0
1	4	13	5	3	6	1
2	5	0	7	1	8	1
3	4	0	2	0	6	0
4	0	0	0	0	6	0
5	1	0	1	0	1	2

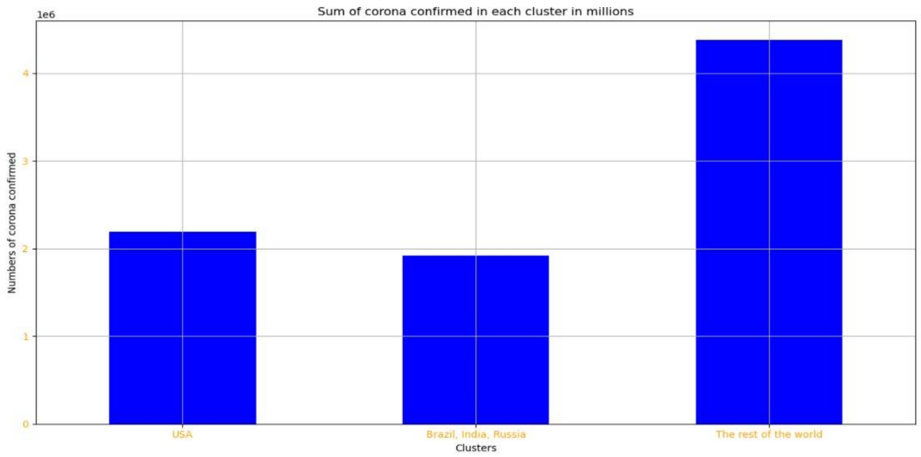
The last data we ran on Naive Bayes is the deaths data set. The best division, in this case, is when the test group is 40%. The accuracy that results from this division is 28%. Quite a low percentage compared to the previous two data sets. From the Confusion matrix, it can be understood that the low percentage of prediction is mainly due to the incorrect classification of countries from the African continent to the European continent. It can be seen from the graphs in 'second statistical analysis' class, that the division of the deaths by continents is relatively in pairs of continents, so it is difficult for the algorithm to identify the continent and this causes a relatively low prediction.

K-means

In the unsupervised part, we used "Kmeans" algorithm.

In the first data frame, we divided our data to 3 clusters. After we activated the algorithm, we got 3 interesting tables. In the first table, we had only one sample! In the second one, we had three samples and in the third, we had all the rest. After a deep exploring, we found out that the algorithm associate to cluster 1 the country which has the most confirmed cases of the "Covid-19" disease which is "USA". The second cluster contains the 3 countries which have the most confirmed cases of "Covid-19" disease except* "USA". Those countries are "Brazil", "India" and "Russia". From that, we can learn that the algorithm associates the cluster by the number of "Covid-19" cases, without connection to the continent the countries come from.

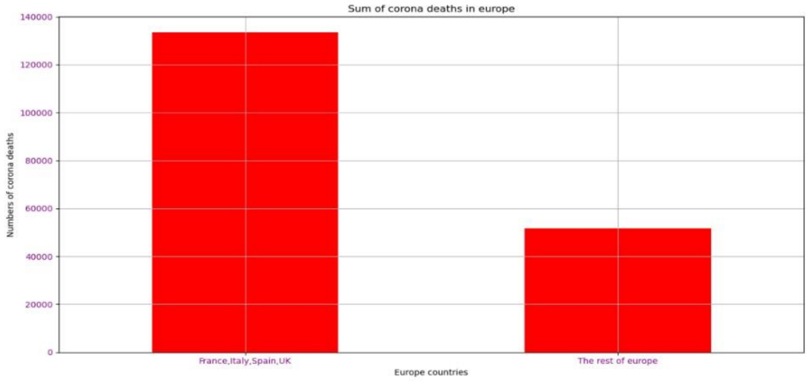
Below is a bar plot of the sum of each cluster of confirmed cases:



We can learn from the graph that the number of corona confirmed cases in "USA" is a bit more than a half (50.03%) compares to the rest of the world (except "Brazil", "India" and "Russia"). In addition, we can see that the other 3 countries which follow "USA" had a bit less confirmed cases, but it can be conclude that those 4 countries have a tremendous relative part of the confirmed cases around the world.

In the second data frame, which is the global death from the "Covid -19" disease, we divided our data to 4 clusters. The first cluster contains only one country which has the most death cases in the world, "USA". The second cluster contains 4 countries and all of them are from "Europe" continent. After exploring those countries, we found out that they are the countries with the most death cases in "Europe".

Below is a bar plot of the death cases in "Europe":



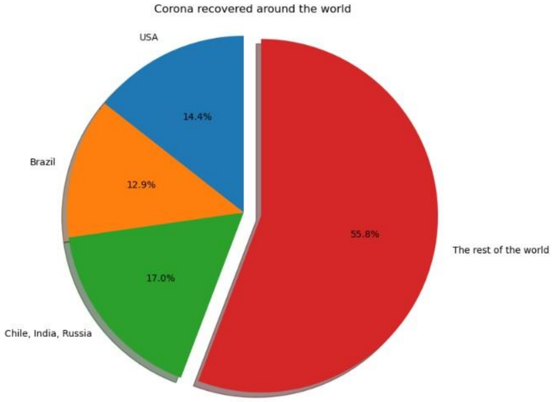
From this graph, we can learn that those 4 countries have the major impact of the number of deaths from "Covid-19" disease around "Europe". In other words, 72.05% of the deaths in "Europe" because of the "Covid-19" disease occurred in those 4 countries.

The third cluster contains 3 countries, those countries are "Brazil", "India", "Mexico" which have the most deaths cases around the world except "USA" and 4 countries in "Europe" from the second cluster.

Those 3 countries are from a different continent. Finally, the last cluster contains all the rest countries around the world.

In the last data frame of "Covid – 19" about the recovered from the disease, we divided the table to 4 clusters. 2 clusters contain 2 countries, which are "USA" and "Brazil". Those two countries have the most of recovered people from the disease. The third cluster contains 3 countries from a different continent "Chile", "India" and "Russia", those 3 countries have the most recovered cases after "USA" and "Brazil". The last cluster contains all the rest of the world recovered cases.

Below is a pie about the global recovered:



In this pie we can learn that 5 countries contain almost half of the recovered cases around the world 44.3%.

In conclusion, we can learn that the algorithm did not divide the countries by their continent/region. "USA" appeared alone in each cluster table, which can show us that this country has the biggest effect (significantly) about the "Covid – 19" numbers in comparison to the other countries around the world. Moreover, it can be seen from the results that 5-6 countries effect the most about the numbers of the "Covid-19" whether it is about the confirmed, death, or recovered cases. We can learn that if those specific countries take control on the disease, the numbers will grow much slower than they do these days.