STAT 564
Project 3: COVID-19

Group R-evolution

Dilay Özkan & Alper Şener

June 5, 2020

Starting from late 2019, a coronavirus started to be an issue; and due to inadequate actions of the Governments and related organizations, it is declared as a pandemic as of 11th of March 2020. In this project, we start with providing very brief information for the pandemic named Coronavirus Disease 2019 (COVID-19). Subsequently, we have analyzed the COVID-19's effects on the electricity consumption in Turkey. Finally, we have tested several statistical methods to classify the COVID-19 patients' data in South Korea.

Part A: COVID-19 Pandemic and Its Impact on Energy Consumption

As of 1st of June 2020, more than 6 million people have diagnosed to COVID 19. United States, Brazil and Russia are the leading countries in terms of cases, respectively. Figure 1 shows the most affected countries and their number of daily cases. Even though Brazil and Russia were not in top 5 countries as the end of April, cases in these two countries were boomed, since necessary precautions were not taken. Furthermore, top ten countries in terms of number of COVID-19 cases corresponds to two thirds of cases, while population of these countries are one-third of the world.

Figure 2 presents the countries where the total number of death is the highest because of COVID-19. The total number of deaths due to the COVID-19 is reached to 371,857 people, and therefore we observed that ca. 6% of diagnosed cases are resulted with death. When we consider the number of deaths, we observed that the leading countries are United States, United Kingdom and Italy. Dowd et al. (2020) posits that age and sex composition of the countries affect the mortality rate so that leading countries in amount of cases and death may differ due to demographic features. Moreover, we still need to gather more data to obtain healthy results since each country may have different characteristics and the coronavirus is still in an increasing trend in some of the countries. For instance, COVID-19 cases in Brazil has skyrocketed in the second half of May 2020, so that we can see the virus' effects on Brazil in the following months.

In this project we analyze Turkey and South Korea in different aspects and diagnosed cases in Turkey has surpassed 160,000 levels, while it is around 10,000 cases in South Korea. On the other hand, COVID-19 related deaths in Turkey and South Korea are 4,540 and 271, respectively. By looking the number of deaths, these two countries may seem less affected by the virus with respect to the countries that affected the most from the COVID-19. However, this pandemic has also significant side effects which we will analyze in the following part of this project.

Please note that in order to analyze the COVID-19 data visually, we created an R script named *Covid_Data.R* and an RShiny script named *Covid_Shiny.R* which includes diagnosed cases, deaths and per million case and death amounts for each country. We obtain data for this analysis from Roser et al's (2020) research.

Starting from this part, we have analyzed the effects of COVID-19 on electricity demand of Turkey. The electricity consumption of Turkey in hourly breakdown can be reached through the transparency platform of Energy Exchange Istanbul (EXIST) (2020). As the end of May 2020, there are 38,688 data, starting from January 1st, 2016 to May 31st, 2020. The data provider, EXIST, is the electricity market operator in Turkey and market participants are required to provide data to EXIST, therefore we assume that the provided data is reliable (Resmi Gazete, 2002).

To analyze the COVID-19 effects on electricity demand, we have determined to compare 2020 data with the data from previous years, since electricity consumption has seasonality, which is illustrated in following pages. We started with the understanding of historic data to obtain convenient outcomes. The historic annual demand shows that, electricity consumption in 2016 was less than the following years, while the annual consumption in 2017-2019 period were similar, displayed in Figure 3. Electricity demand in 2016 was ca. 275 TWh, while it increased to ca. 290 TWh levels after 2017, as shown in Figure 3. Thus, we have decided to discard 2016 data while performing our analyses and continue with remaining 29,904 data points.

Apart from the annual demand we observed that day of the week has a crucial effect on electricity demand. Fig-

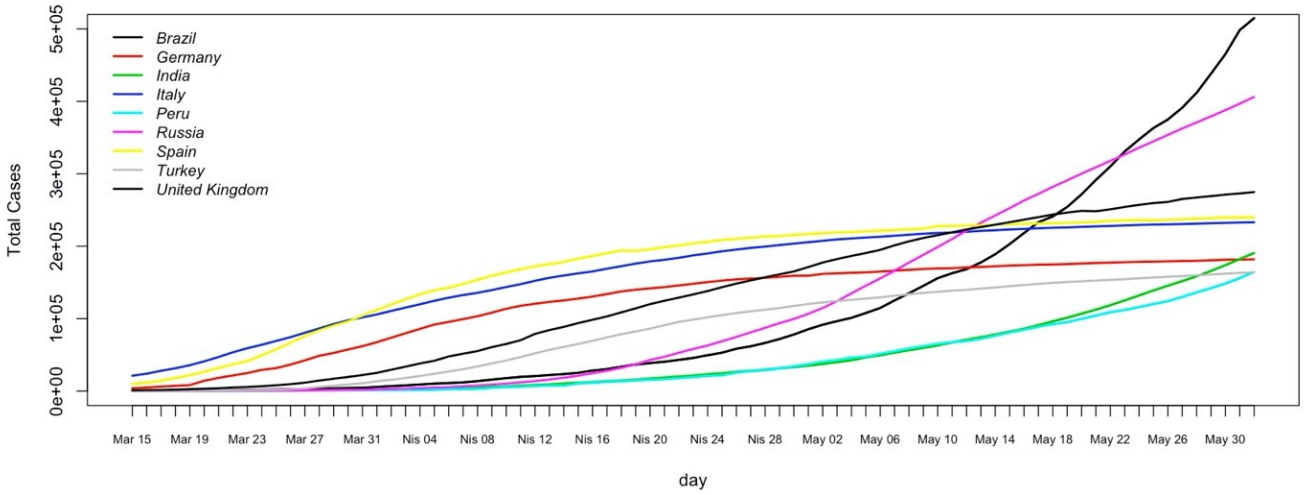

Figure 1: The most affected countries in terms of COVID-19 cases (except United States)

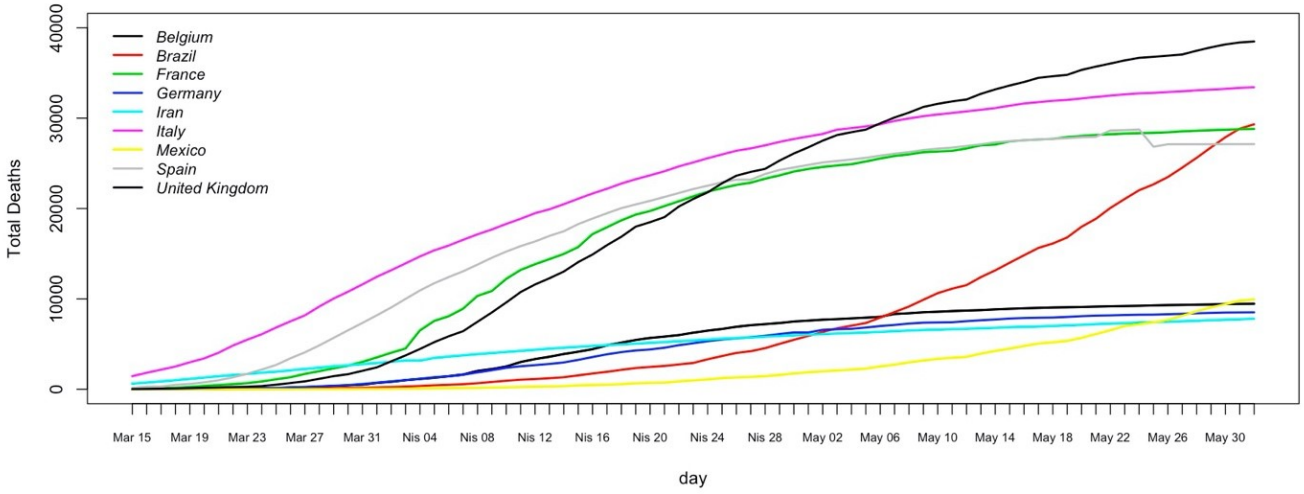

Figure 2: The Most Affected Countries in Terms of COVID-19 Deaths (except United States)

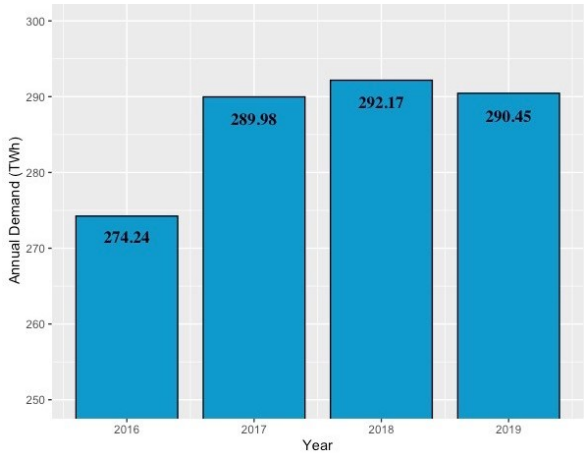

Figure 3: Electricity Demand in Turkey, 2016-2019

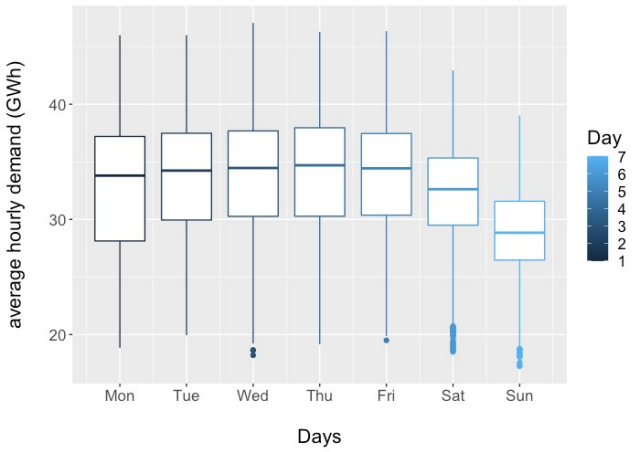

Figure 4: Average Hourly Electricity Demand in Turkey by Weekdays

ure 4 shows the average hourly consumption in weekdays are ca. 34 GWh while it decreases to ca. 30 GWh in weekends.

As illustrated in the Figre 4, consumption during Sundays are lower than the other days. In order to see whether there is a significant difference between a weekday (i.e. Wednesday) and Sunday, we utilized a Welch Two Sample t-test. The results are given in Figure 5, and they show that demand in a weekday is significantly greater than Sundays.

Since day of the week has a significant importance on electricity demand, we have compared the monthly demand amounts through following method. Firstly, we calculate average hourly demand in day of the week breakdown for each month and year, starting from January 2017. After that, we take the arithmetic average of these days to obtain a monthly average demand.

This method enables us to get rid of the imbalances of day types in a month. As an example, 1st day of July in 2017 is Saturday, while it is Monday in 2019. So, there are 4 and 5 weekend periods in July 2017 and July 2019, respectively. So, please note that *average hourly demand of a month* corresponds to the arithmetic average of each seven day's average demand (rather than hourly average of a month), throughout the Part A.

After analyzing the historic data, we started to analyze the COVID-19 effects. First of all, we have compared the average hourly demand in monthly breakdown. As shown in Figure 6, a slight increase was realized on hourly demand during the first two months of 2020. Subsequently, average hourly demand in March 2020 was decreased to the 2017-2019 demand levels. However, the effects of COVID-19 can be observed explicitly in April and May 2020. As it is illustrated obviously in the Figure 6, a dramatic decrease has experienced in April 2020. The mean hourly demand during April and May was around 31.0 GWh in 2017-2019 while it decreased to 26.5 GWh (ca. 14.5% decrease) in 2020. In consideration of the increasing trend on January and February 2020, we may posit that the electricity demand has shrunk more than 15%. Therefore, a dramatic decrease on electricity demand in Turkey is experienced due to COVID-19, which shows us the pandemic will affect the Turkish economy crucially, since energy usage is one of the leading indicators of the economic development.

Furthermore, we cannot obtain a healthy result for March 2020 by assessing the monthly data, since first case of this disease was diagnosed in mid-March. Therefore, we analyzed this month in a more detailed aspect. Although the first diagnose in Turkey announced in 11^{th} of March, precautions were not taken immediately. Therefore, we create a RShiny code to assess the demand decrease interactively. When we analyze the demand levels, we observed that a dramatic decrease on demand experienced within last one and a half weeks of March 2020.

As illustrated in the Figure 7, a slight increase on demand (2.5% with respect to 2017-2019 averages) can be observed in first three weeks of March 2020, with respect to previous years. So that, electricity demand was in an increasing trend in the first 22 days of March as experienced in January, and February. However, after 23th of March a significant shrinkage occurred on electricity consumption. The average demand decreased to 30.3 GWh, which is 5.5% less than the 2017-2019 average. If we consider the increasing trend in 2020, we may deduce that more than 7% demand has shrunk within the last two weeks of March 2020 due to the COVID-19 effect.

As a result, COVID-19 affects almost all kinds of businesses in the world, and most of them are affected negatively. We started our study with an assumption that electricity demand in Turkey is affected negatively because of the business shutdowns and changes on daily routines. From another perspective, residential usage will be increased since more people have stayed or worked from home which may increase the elec-

```
        Welch Two Sample t-test

data:  wednesday_demand$`Tüketim Miktarı (MWh)` and sunday_demand$`Tüketim Miktarı (MWh)`
t = 55.407, df = 7868.7, p-value < 2.2e-16
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 4.956425      Inf
sample estimates:
mean of x mean of y
 34.17391  29.06582
```

Figure 5: Results of Welch Two Sample t-test (Wednesday vs. Sunday)

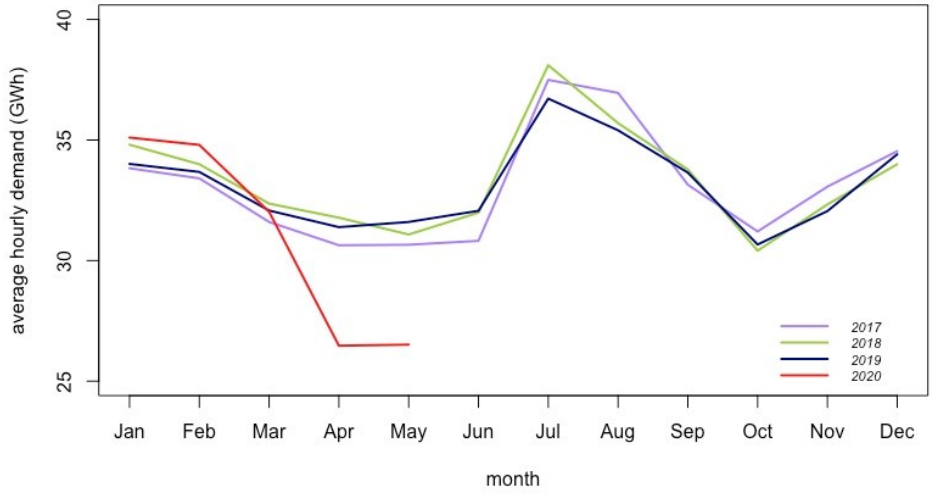

Figure 6: Monthly Average Electricity Demand (Jan 2017 – May 2020)

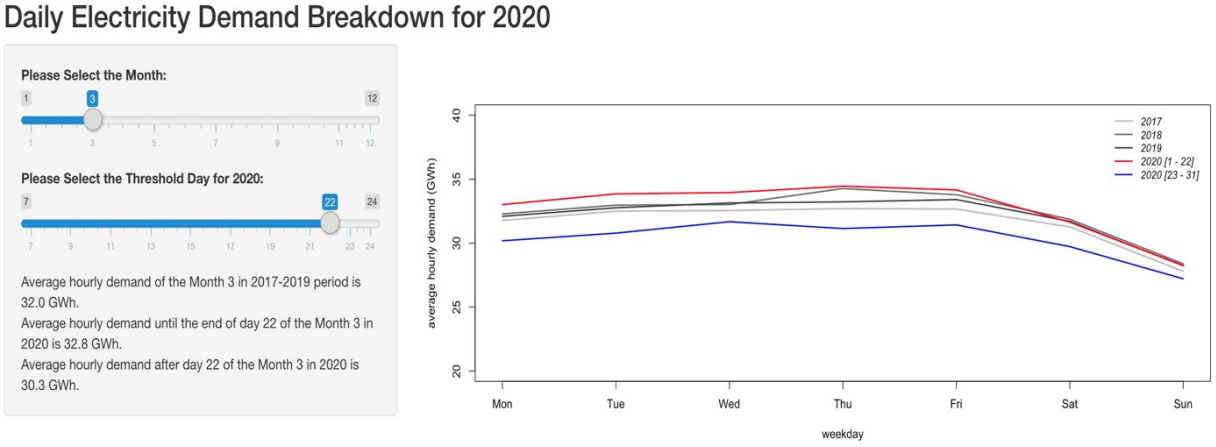

Figure 7: Average Electricity Demand of March in Daily Breakdown

tricity demand. But, in 2019, households consumed around one-third of electricity in Turkey, while small businesses and industries are responsible for ca. two-thirds of the consumption (EPDK, 2020), so that decrease seems more likely without undergoing any detailed analysis. Unfortunately, sectoral breakdown of electricity consumption for April and May 2020 has not disclosed yet. That is why, we made our analysis with respect to overall electricity consumption data.

We observed that, the decrease on demand has not realized suddenly since businesses continued their operations after the announcement of first case. The decreasing effect can be observed starting from the 4^{th} week of March and it still continues as the end of May. Moreover, shrinkage on electricity demand has strengthened starting from April, since precautions from the disease was hardened by the Government. In April and May 2020, more than 15% of decrease was realized, while it was around 10% in the last week of March 2020. By considering the relaxation on the disease precautions starting from 1^{st} of June, we expect that the COVID-19's effects on electricity consumption will be smaller after June 2020 with respect to April and May 2020.

Please note that the name of the related R script for electricity demand analysis are *Electricity_Data.R* and *Electricity_Shiny.R*.

PART B: Machine Learning Algorithms for Making Predictions

In this part, South Korea data set, prepared by Jihoo et al.(2020), which contains Covid-19 patients' information is used. It gives some information about the patients, and it classifies them according to their conditions. Since the data set has many irrelevant features, it is cleaned before using it. Data cleaning is mentioned in Section 1. Then, in Section 2, three different machine learning algorithms are applied to the data set, which are logistic regression, k-nearest neighbors, and decision tree (classification tree). The performance of each model is evaluated using 10 times 5-fold cross validation. However, the data set is imbalanced, i.e. most of the observations belongs to one class. To handle this situation Synthetic Minority Over-sampling Technique proposed by Bowyer et al. (2002) is used. The new data set is generated by using both oversampling and under-sampling in each algorithm. Then, all models are compared in Section 3.

1. Data Cleaning and Preparation

Originally, the South Korea data set has three different response variables which are 'released', 'deceased', and 'isolated'. We eliminate the observations of 'isolated' class because we would like to build a classification model with two different classes. In addition to that, the data set has so many irrelevant features. The columns of these features are eliminated. The remaining columns are the birth year, sex, confirmation of the disease, and the countries of the patients. Sex and the birth year are important features to determine whether a patient will be died. Therefore, the observations do not have birth year or sex information are deleted. The reason we choose to reduce these observations is that the total number of observations is still enough. The R script called *'DataCleaning'* takes the original data set and modifies the data set as mentioned above.

2. Prediction Algorithms

In this section, 3 different machine learning algorithms are applied to the data to make predictions about whether the patients will be released or deceased. Logistic regression, k-nearest neighbor, and classification tree models are constructed. Since the model is a classification model, we select the algorithms accordingly.

2.1. Logistic Regression

The main idea of the logistic regression is to build a regression model which classifies the data according to the features. Using logistic regression is suitable because we have two classes in the response variables. At first, the logistic regression model is run for all the data. Please note that when the logistic regression model is constructed, the 'country' feature is eliminated. Most of the observations' country feature is 'Korea'. That is why, when the data set is divided into two as training and test set using cross validation, sometimes training set does not contain the observations of rare countries. Therefore, when test set is run, it gives an error because the model did not learn that level. The related R script's name is *'PatientInfo_LogisticRegression'*.

The misclassification error rate is found as 0.02641934. However, the classes in the data set do not have the same ratio. The class 0, which is 'deceased' class, contains approximately 4% of the whole data whereas 96% of the observations belongs to the class 1, 'released' class. Therefore, it will be better to check the misclassification error rate of class 0 to evaluate the performance of the logistic regression model. The model gives the following Table 1.

Table 1: Error rates of class 0 and 1

	Correct Prediction	Total Observation	Error Rate
Class 0	23	66	0.6515
Class 1	1709	1713	0.0023
Total	1711	1779	0.0382

Table 1 shows that the model is not good at predicting the observations of Class 0. Although, when we apply cross validation, the overall average error rate of each cross validation is so small as in Table 2, it is just due to large number of observations in Class 1. In fact, we see that none of the cross-validation's average accuracy of Class 0 is not good enough, shown in Table 3.

Therefore, to observe the power of the model, the data set should be manipulated. The following explains how to change the data set so that we can make predictions accordingly.

Table 2: Average error rates in 10 times 5-fold CV in logistic regression

Cross Validation	Average Error Rate
1	0.028669
2	0.026979
3	0.027544
4	0.026987
5	0.027547
6	0.026417
7	0.026987
8	0.026982
9	0.027539
10	0.026412

Table 3: Average accuracy of Class 0 in 10 times 5-fold CV in logistic regression

Cross Validation	Average Accuracy of Class 0
1	0.36
2	0.35
3	0.33
4	0.35
5	0.37
6	0.36
7	0.36
8	0.33
9	0.33
10	0.36

- Dealing with the data set by using SMOTE

  As we mentioned before, the data set is imbalanced since the %4 of the observations belongs to Class 0 whereas the rest is in Class 1. Therefore, the data set should be manipulated in a way that the ratio of Class 0 should be large enough to make predictions. For that purpose, we can use under-sampling or oversampling methods. One of the most common sampling methods in the literature is Synthetic Minority Over-sampling Technique (SMOTE) proposed by Bowyer et al. (2002). This technique uses bootstrapping sampling by looking at the nearest neighbors of the observations. Moreover, SMOTE can be used for under-sampling the data set, which eliminates most of the observations that belong to the majority class. We will both make oversampling and under-sampling with SMOTE.

  a. Oversampling

  The total number of observations is increased from 1779 to 2046. The ratio of Class 0 becomes 35% and that of Class 1 becomes 65%. That is, 660 new observations of Class 0 are added to the data set. The logistic regression model is run for the new data set. Moreover, 10 times 5-fold cross validation is applied. The mean error rate of each cross validation is given in Table 4. This shows that the average error rate is approximately 10% in each cross validation. Although the error rate of oversampling seems higher than that of the original data set, we should also check the prediction accuracy of class 0 to evaluate the performance of the model. The average accuracy of each cross validation is given in Table 5. It shows that the average accuracy is 82% which indicates the model can predict Class 0 observations better.

Table 4: Average error rates of oversampling in logistic regression

Cross Validation	Average Error Rate
1	0.10459
2	0.10608
3	0.10847
4	0.108
5	0.10747
6	0.10752
7	0.10897
8	0.10607
9	0.10656
10	0.10902

Table 5: Average accuracy of Class 0 obtained by oversampling in logistic regression

Cross Validation	Average Accuracy of Class 0
1	0.82765
2	0.82943
3	0.82809
4	0.82933
5	0.8282
6	0.82888
7	0.82246
8	0.82933
9	0.82707
10	0.82508

  b. Under-sampling

  The under-sampling technique eliminates most of the observations in majority class. Total number of observations is reduced to 858 and 198 of them belong to Class 0. That is, Class 0 contains the 23% of the data set. When the logistic regression and 10 times 5-fold cross validation are applied to the new data set, the following mean error rates are obtained shown in Table 6. The average error rate is approximately 9% in all cross validations. When we check the mean accuracy of Class 0 in each cross validation, Table 7 is obtained. Table 7 indicates that almost three-fourths of the observations in Class 0 can be predicted correctly.

Table 6: Average error rates of under-sampling in logistic regression

Cross Validation	Average Error Rate
1	0.09085
2	0.09558
3	0.09086
4	0.09672
5	0.09211
6	0.09083
7	0.09218
8	0.09212
9	0.09204
10	0.09216

Table 7: Average accuracy of Class 0 obtained by under-sampling in logistic regression

Cross Validation	Average Accuracy of Class 0
1	0.75804
2	0.73785
3	0.75855
4	0.73433
5	0.74832
6	0.75337
7	0.75858
8	0.75419
9	0.74636
10	0.75148

When we compare those two techniques, we see that the error rates and accuracies of both techniques are similar. Under-sampling has lower error whereas oversampling has higher accuracy. Shortly, both oversampling and under-sampling techniques imply a significant improvement on the predictions when it is compared to the logistic regression model constructed by using original data set.

Please note that the name of the related R script for oversampling and under-sampling techniques is *'Sampling_LogisticRegression'*.

2.2. K Nearest Neighbors Algorithm

K-nearest neighbors (k-NN) algorithm predicts the class of an observation by looking at the classes of k closest data points of that observation. The most observed class in those closest data points is assigned as the prediction class of that observation. The algorithm is easy to apply but the value k, which determines the number of the nearest data points to be looked for, is not known. That is, it is an hyper-parameter whose value should be determined. One of the appropriate methods for determining the value of k is to try different k values and choose the one that gives the minimum test error rate. In fact, it will be more accurate when this method is repeated using cross validation.

The k-NN algorithm is run for the whole data set first using 10 times 5-fold cross validation. Moreover, k values starting from 1 to 50 are tried in each time. The k value with minimum test error rate is also stored, shown in Table 8. In fact, in each cross validation the test set is run for k values starting from 1 to 50 and the misclassification errors are kept. Then, the average of test error rate corresponding to each k value is taken in that cross validation. The k value which is related to the minimum average error rate is chosen as the optimal k value in that cross validation. This method is applied 10 times since we use 10 times cross validation. The optimal k values and minimum average error rates of each cross validation can be found in Table 9. It shows that the average error rates in each cross validation are so close. To determine the optimal k value of the model, the one which gives the minimum error rate in these cross validations can be chosen. In this case, k is chosen as 11.

Table 8: Misclassification error rate of data set in 10 times 5-fold CV in k-NN

Cross Validation	Average Error Rate
1	0.03484
2	0.03484
3	0.03204
4	0.03428
5	0.03429
6	0.03485
7	0.0343
8	0.0343
9	0.03486
10	0.03541

Table 9: k values and corresponding error rates in 10 times CV

Cross Validation	Optimal k value	Minimum Average Error rate
1	14	0.03708973
2	16	0.03653110
3	11	0.03597721
4	18	0.03709289
5	31	0.03709922
6	10	0.03653584
7	33	0.03711030
8	19	0.03710714
9	27	0.03654850
10	22	0.03653743

When we go back to the error rates in Table 8, the model seems like working well. However, it does not predict the observations in Class 0 properly. Table 10 shows the average accuracy of the model in Class 0. The accuracy of each fold in Table 10 is so low such that sometimes the model cannot predict almost any of the observations that belongs to Class 0. Accuracy of k-NN algorithm are worse than that of logistic regression. Therefore, the model cannot give proper results

in Class 0.

The R script's name used in this part is *'PatientInfo_KNN'*.

Table 10: Average accuracy of class 0 in 10 times 5-fold CV in k-NN

Cross Validation	Average Accuracy of Class 0
1	0.09744
2	0.10399
3	0.22156
4	0.18974
5	0.08889
6	0.18
7	0.09444
8	0.16494
9	0.19281
10	0.03539

Since the data is imbalanced, oversampling and under-sampling techniques are also applied.

a. Oversampling

The same data set created in Logistic Regression is used. The average misclassification error rate and mean accuracy in 10 times 5-fold cross validation are given in Table 11 and 12, respectively. The error rate is between 7-8%. When we check the accuracy of the model, values are between 81 to 84%. In all cross validations. The optimal k value is found as 1. Therefore, k can be selected as 1.

Table 11: Average error rates of oversampling in k-NN

Cross Validation	Average Error Rate
1	0.08114
2	0.0743
3	0.07432
4	0.07771
5	0.07431
6	0.07183
7	0.07672
8	0.07724
9	0.08016
10	0.07966

b. Under-sampling

The data set is reduced as it was done in under-sampling of logistic regression. When the model is built and run using 10 times 5-fold cross validation, the optimal k values are found as all 1 except for one cross validation. Therefore, the optimal value of k for the model is proposed as 1.

Table 13 and 14 presents the average error rates and mean accuracy of 10 times 5-fold cross validation, respectively. The error rate and accuracies are much better than the original model, but they are worse than those obtained using oversampling technique. Therefore, it will be useful to continue with oversampling technique.

Table 12: Average accuracy of Class 0 obtained by oversampling in k-NN

Cross Validation	Average Accuracy of Class 0
1	0.81644
2	0.84301
3	0.83956
4	0.83439
5	0.84281
6	0.84872
7	0.83325
8	0.8341
9	0.82928
10	0.82594

Table 13: Average error rates of under-sampling in k-NN

Cross Validation	Average Error Rate
1	0.10145
2	0.10142
3	0.10371
4	0.10133
5	0.10261
6	0.1072
7	0.09558
8	0.10487
9	0.10606
10	0.10019

Table 14: Average accuracy of Class 0 obtained by under-sampling in k-NN

Cross Validation	Average Accuracy of Class 0
1	0.74548
2	0.74315
3	0.69257
4	0.69751
5	0.67986
6	0.71151
7	0.71174
8	0.73364
9	0.73154
10	0.723

The codes of these two methods used in k-NN algorithm is available in the R script called *'Sampling_KNN'*.

2.3. Classification Tree

Classification tree is a type of decision trees where the response variable is discrete. A tree-like structure splits the data set according to the feature values of the ob-

servation. After the tree learns from the data set, it is tested using the test set. Please note that the name of the related R script is *'PatientInfo_DecisionTree'*.

We construct the classification tree model and test the set using 10 times 5-fold cross validation. The mean error rate of the cross validations and average accuracy of Class 0 in those cross validations can be shown in Table 15 and 16, respectively. Table 15 and 16 show that although the model has low average error rate, it does not predict the observations belonging to Class 0 well. As it is mentioned before, the reason is that the data set is imbalanced. To solve this problem, again we apply over and under sampling techniques to classification tree and the code is available in R script *'Sampling_DecisionTree'*.

Table 15: Average error rates in 10 times 5-fold CV in classification tree

Cross Validation	Average Error Rate
1	0.03091
2	0.0281
3	0.02811
4	0.03148
5	0.02979
6	0.02923
7	0.02756
8	0.02867
9	0.02811
10	0.02755

Table 16: Average accuracy of class 0 in 10 times 5-fold CV in classification tree

Cross Validation	Average Accuracy of Class 0
1	0.20271
2	0.25034
3	0.20364
4	0.18053
5	0.20544
6	0.23616
7	0.34091
8	0.25432
9	0.26976
10	0.26178

a. Oversampling

New data set constructed in both logistic regression and k-NN algorithm is used again. The model is evaluated using 10 times 5-fold cross validation. Table 17 and 18 show the mean error rate and Class 0 accuracies on average, respectively. The error rate becomes approximately 10% whereas the average accuracy of Class 0 is increased to 77-79%.

Table 17: Average error rates of oversampling in classification tree

Cross Validation	Average Error Rate
1	0.10749
2	0.10756
3	0.1095
4	0.10703
5	0.10458
6	0.10559
7	0.1095
8	0.10707
9	0.10708
10	0.10608

Table 18: Average accuracy of Class 0 obtained by oversampling in classification tree

Cross Validation	Average Accuracy of Class 0
1	0.78744
2	0.78808
3	0.77951
4	0.79019
5	0.79769
6	0.7959
7	0.77783
8	0.7906
9	0.78929
10	0.79119

b. Under-sampling

The data set is reduced so that the observations used in logistic regression and k-NN algorithm are obtained. Again 10 times 5-fold cross validation is run, and the corresponding performance measures are given in Table 19 and 20.

Table 19: Average error rates of under-sampling in classification tree

Cross Validation	Average Error Rate
1	0.10722
2	0.10603
3	0.10379
4	0.10264
5	0.10841
6	0.10956
7	0.10841
8	0.10605
9	0.09906
10	0.1061

The average error rate is nearly 10% and the mean accuracy of Class 0 is between 72-77%. In both the oversampling and under-sampling, the error rate is the same whereas the oversampling technique performs better on predicting Class 0 observations.

Table 20: Average accuracy of Class 0 obtained by under-sampling in classification tree

Cross Validation	Average Accuracy of Class 0
1	0.76162
2	0.77504
3	0.73655
4	0.72934
5	0.7579
6	0.75057
7	0.74331
8	0.70986
9	0.76076
10	0.7496

3. Comparison of the Algorithms

To compare the algorithms, we should check the average error rates and the accuracies of each method. The overall error rate and accuracy averages of 10 times cross validation for each method is provided in Table 21 and 22, respectively.

Table 21: Average error rates of all models in each algorithm

Model	Original Data	Over-sampling	Under-sampling
Logistic Regression	0.0272	0.10727	0.09254
k-NN Algorithm	0.0344	0.07674	0.10244
Classification Tree	0.0290	0.10715	0.10573

Table 22: Average accuracy of all models in each algorithm

Model	Original Data	Over-sampling	Under-sampling
Logistic Regression	0.35	0.82755	0.75011
k-NN Algorithm	0.1369	0.83475	0.717
Classification Tree	0.2406	0.78877	0.74745

These tables infer that when original data set is used, logistic regression is the best since it has the lowest average error and the highest mean accuracy of Class 0. However, when oversampling technique is applied, k-NN algorithm is improved the most. In under-sampling technique, logistic regression gives the best results.

References

[1] Chawla, N. V., Bowyer K.W., Hall, L.O., Kegelmeyer, W.P. SMOTE: synthetic minority over- sampling technique. *Journal of Artificial Intelligence Research*, 2002;16:321–57.

[2] Dowd, J. B., Andriano, L., Brazel, D. M., Rotondi, V., Block, P., Ding, X., ... & Mills, M. C. Demographic science aids in understanding the spread and fatality rates of COVID-19. *Proceedings of the National Academy of Sciences*, 2020;117(18), 9696-9698.

[3] Max Roser, Hannah Ritchie, Esteban Ortiz-Ospina and Joe Hasell,Coronavirus Pandemic (COVID-19). 2020. Published online at OurWorldInData.org. Retrieved from: *'https://ourworldindata.org/coronavirus'*[Online Resource]

[4] Gerçek Zamanlı Tüketim. (n.d.). Retrieved June 2, 2020, from *https://seffaflik.epias.com.tr/transparency/tuketim/gerceklesen-tuketim/gercek-zamanli-tuketim.xhtml*

[5] Resmi Gazete; E. P. D. Kurumundan (2002). Elektrik Piyasası Lisans Yönetmeliği

[6] EPDK. (2020). Elektrik Piyasası Aylık Sektör Raporları. June 2, 2020. Retrieved from *https://www.epdk.org.tr/Detay/Icerik/3-0-23/elektrikaylik-sektor-raporlar*

[7] Jihoo K. et al. Data Science for COVID-19 in South Korea. 2020. Published online at kaggle.com. Retrieved from *https://www.kaggle.com/kimjihoo/coronavirusdataset?select=PatientInfo.csv*

R Codes of Part B

- DataCleaning.R

```
#_____________________________Data Cleaning_____________________________________

#_____________Eliminating Third Class_____________

data <- read.csv('PatientInfo_original.csv')

data <- subset(data, select = c(3,4,6,9,18))

data$state <- factor(data$state)

data$state <- ifelse(data$state=='released',1,ifelse(data$state=='deceased',0,2))

ind <- which(data$state!=2)

data <- data[ind,]

#___________Handling with Missing Values_____________

cat('\nThere are', sum(is.na(data$sex)), 'missing value(s) in feature sex.')

cat('\nThere are', sum(is.na(data$birth_year)), 'missing value(s) in feature birth_year.')

ind2 <- which(is.na(data$birth_year))

data <- data[-ind2,]

cat('\nThere are', sum(is.na(data$country)), 'missing value(s) in feature country.')

data$disease <- ifelse(is.na(data$disease),0,1)

cat('\nThere are', sum(is.na(data$disease)), 'missing value(s) in feature disease')
```

- PatientInfo_LogisticRegression.R

```
data <- read.csv('PatientInfo.csv') #state:1 -> recovered, 0->deceased    disease:1 -> true

data <- data[,1:7]

data$sex <- factor(data$sex)

data$country <- factor(data$country)

data$province <- factor(data$province)

data$city <- factor(data$city)

data$disease <- factor(data$disease)

data$state <- factor(data$state)

logistic_model <- glm(state ~ sex+disease+birth_year, family = binomial(link = 'logit'), data =
data)

summary(logistic_model)

fitted.probabilities <- predict(logistic_model, data, type = 'response')

fitted.results <- ifelse(fitted.probabilities > 0.5, 1, 0)

missclassError <- mean(fitted.results != data$state)

accuracy <- 1 - missclassError

print(1- missclassError)


count <- 0

correct <- 0

for (i in 1:length(fitted.results)){

  if(data[i,7]==0){

    count <- count + 1

    if (fitted.results[i]==0){

      correct <- correct + 1

    }

  }


}

print(table(data$state, fitted.probabilities>0.5))


#_____________ 10 times 5-fold Cross Validation_____________________________________

library(caTools)

cv <- 1:10 # 5 times

cv2 <- 1:5 # 5-fold cv

missclassError <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))

accuracy <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))

count <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))

correct <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
```

```
length <- ceiling(dim(data)[1]/length(cv2))
for (j in cv){
  k <- 1
  ind <- sample(dim(data)[1])
  data <- data[ind,]
  cat('\n-------------- CV',j,'-------------- ')
  for (i in cv2){
    cat('\n Fold-', i)
    a <- i*length*(i!=length(cv2)) + dim(data)[1]*(i==length(cv2))
    cat('\nData is selected between', k, '-', a)
    index <- k:a
    final.test <- data[index,]
    final.train <- data[-index,]
    logistic_model <- glm(state ~ sex+disease+birth_year, family = binomial(link = 'logit'), data
= final.train)
    fitted.probabilities <- predict(logistic_model, final.test, type = 'response')
    fitted.results <- ifelse(fitted.probabilities > 0.5, 1, 0)
    missclassError[j,i] <- mean(fitted.results != final.test$state)
    accuracy[j,i] <- 1 - missclassError[j,i]
    cat('\nThe accuracy is', 1- missclassError[j,i])
    cat('\nThe missclassification error is', missclassError[j,i])

    k <- a + 1

    for (p in 1:length(fitted.results)){
      if(final.test[p,dim(final.test)[2]]==0){
        count[j,i] <- count[j,i] + 1
        if (fitted.results[p]==0){
          correct[j,i] <- correct[j,i] + 1
        }
      }

    }
    cat('\nOnly', correct[j,i], 'out of',count[j,i], 'class-0 are predicted correctly.')

  }
}
```

- Sampling_LogisticRegression.R

```
data <- read.csv('PatientInfo.csv') #state:1 -> recovered, 0->deceased    disease:1 -> true
data <- data[,1:7]
data$sex <- factor(data$sex)
data$country <- factor(data$country)
data$province <- factor(data$province)
data$city <- factor(data$city)
data$disease <- factor(data$disease)
data$state <- factor(data$state)
#_____________________Creating New Observations_____________________
#___________Oversampling_____________
data <- subset(data, select = c(1,2,6,7))
library(DMwR)
data_new <- SMOTE(state ~., data, perc.over = 1000)
count0 <- sum(ifelse(data_new$state==0,1,0))
count1 <- sum(ifelse(data_new$state==1,1,0))



#         _____________ 10 times 5-fold Cross Validation_____________
library(caTools)
set.seed(100)
cv <- 1:10 # 5 times
cv2 <- 1:5 # 5-fold cv
missclassError <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
accuracy <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
count <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
correct <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
length <- ceiling(dim(data_new)[1]/length(cv2))
for (j in cv){
  k <- 1
  ind <- sample(dim(data_new)[1])
  data <- data_new[ind,]
  cat('\n-------------- CV',j,'-------------- ')
  for (i in cv2){
    cat('\n Fold-', i)
    a <- i*length*(i!=length(cv2)) + dim(data)[1]*(i==length(cv2))
    cat('\nData is selected between', k, '-', a)
    index <- k:a
    final.test <- data[index,]
```

```
    final.train <- data[-index,]
    logistic_model <- glm(state ~ sex+disease+birth_year, family = binomial(link = 'logit'), data
= final.train)
    fitted.probabilities <- predict(logistic_model, final.test, type = 'response')
    fitted.results <- ifelse(fitted.probabilities > 0.5, 1, 0)
    missclassError[j,i] <- mean(fitted.results != final.test$state)
    accuracy[j,i] <- 1 - missclassError[j,i]
    cat('\nThe accuracy is', 1- missclassError[j,i])
    cat('\nThe missclassification error is', missclassError[j,i])

    k <- a + 1

    for (p in 1:length(fitted.results)){
      if(final.test[p,dim(final.test)[2]]==0){
        count[j,i] <- count[j,i] + 1
        if (fitted.results[p]==0){
          correct[j,i] <- correct[j,i] + 1
        }
      }

    }
    cat('\nOnly', correct[j,i], 'out of',count[j,i], 'class-0 are predicted correctly.')

  }
}



#_____________________Deleting Some Observations_____________________
#___________Undersampling_____________
data <- read.csv('PatientInfo.csv') #state:1 -> recovered, 0->deceased    disease:1 -> true
data <- data[,1:7]
data$sex <- factor(data$sex)
data$country <- factor(data$country)
data$province <- factor(data$province)
data$city <- factor(data$city)
data$disease <- factor(data$disease)
data$state <- factor(data$state)
data <- subset(data, select = c(1,2,6,7))
```

```
data_new <- SMOTE(state ~., data, perc.under = 500)

count0 <- sum(ifelse(data_new$state==0,1,0))

count1 <- sum(ifelse(data_new$state==1,1,0))

logistic_model <- glm(state ~ sex+disease+birth_year, family = binomial(link = 'logit'), data =
data_new)

summary(logistic_model)

fitted.probabilities <- predict(logistic_model, data_new, type = 'response')

fitted.results <- ifelse(fitted.probabilities > 0.5, 1, 0)

missclassError <- mean(fitted.results != data_new$state)

accuracy <- 1 - missclassError

print(1- missclassError)

print(table(data_new$state, fitted.probabilities>0.5))

#         _______________ 10 times 5-fold Cross Validation_______________

library(caTools)

cv <- 1:10 # 10 times

cv2 <- 1:5 # 5-fold cv

missclassError <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))

accuracy <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))

count <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))

correct <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))

length <- ceiling(dim(data_new)[1]/length(cv2))

for (j in cv){

  k <- 1

  ind <- sample(dim(data_new)[1])

  data <- data_new[ind,]

  cat('\n-------------- CV',j,'-------------- ')

  for (i in cv2){

    cat('\n Fold-', i)

    a <- i*length*(i!=length(cv2)) + dim(data)[1]*(i==length(cv2))

    cat('\nData is selected between', k, '-', a)

    index <- k:a

    final.test <- data[index,]

    final.train <- data[-index,]

    logistic_model <- glm(state ~ sex+disease+birth_year, family = binomial(link = 'logit'), data
= final.train)

    fitted.probabilities <- predict(logistic_model, final.test, type = 'response')

    fitted.results <- ifelse(fitted.probabilities > 0.5, 1, 0)

    missclassError[j,i] <- mean(fitted.results != final.test$state)

    accuracy[j,i] <- 1 - missclassError[j,i]

    cat('\nThe accuracy is', 1- missclassError[j,i])
```

```
    cat('\nThe missclassification error is', missclassError[j,i])

    k <- a + 1

    for (p in 1:length(fitted.results)){
      if(final.test[p,dim(final.test)[2]]==0){
        count[j,i] <- count[j,i] + 1
        if (fitted.results[p]==0){
          correct[j,i] <- correct[j,i] + 1
        }
      }

    }
    cat('\nOnly', correct[j,i], 'out of',count[j,i], 'class-0 are predicted correctly.')
  }
}
```

- PatientInfo_KNN.R

```
library(class)
data <- read.csv('PatientInfo.csv') #state:1 -> recovered, 0->deceased    disease:1 -> true
data <- data[,1:7]
data$sex <- as.integer(data$sex)
data$disease <- as.integer(data$disease)
data$country <- as.integer(data$country)
data <- subset(data, select = c(1,2,3,6,7)) #important features of dataset are taken


#_____________ 10 times 5-fold Cross Validation_____________________________
library(caTools)
set.seed(100)
cv <- 1:10 # 5 times
cv2 <- 1:5 # 5-fold cv
k_val <- 1:50
misclassError <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
accuracy <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
count <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
correct <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
length <- ceiling(dim(data)[1]/length(cv2))
k_opt <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
chosen_k <- rep(0,length(cv))
mean_error <- rep(0,length(cv))
for (j in cv){
  k <- 1
  ind <- sample(dim(data)[1])
  data <- data[ind,]
  cat('\n-------------- CV',j,'-------------- ')
  ErrorK <- matrix(rep(0,length(cv2)*length(k_val)),nrow = length(cv2), ncol = length(k_val))
  for (i in cv2){
    cat('\n Fold-', i)
    a <- i*length*(i!=length(cv2)) + dim(data)[1]*(i==length(cv2))
    cat('\nData is selected between', k, '-', a)
    index <- k:a
    train <- data[-index,]
    test <- data[index,]
    X.train <- subset(train, select = -c(dim(train)[2]))
    y.train <- train[,dim(train)[2]]
```

```
    X.test <- subset(test, select = -c(dim(test)[2]))
    y.test <- test[,dim(test)[2]]

    for (n in k_val) {
      prediction <- knn(train = X.train, test = X.test, cl = y.train, k = k_val[n])
      ErrorK[i,n] <- mean(prediction != y.test)
    }

    k_opt[j,i] <- which.min(ErrorK[i,])
    misclassError[j,i] <- min(ErrorK[i,])

    accuracy[j,i] <- 1 - misclassError[j, i]
    cat('\nk-value which gives the minimum error is', k_opt[j,i])
    cat('\nThe accuracy is', 1- misclassError[j,i])
    cat('\nThe misclassification error is', misclassError[j,i])


    prediction <- knn(train = X.train, test = X.test, cl = y.train, k = k_opt[j,i])
    for (p in 1:length(prediction)){
      if(y.test[p]==0){
        count[j,i] <- count[j,i] + 1
        if (prediction[p]==0){
          correct[j,i] <- correct[j,i] + 1
        }
      }

    }
    cat('\nOnly', correct[j,i], 'out of',count[j,i], 'class-0 are predicted correctly.')
    average_error <- rep(0,length(k_val))
    for (z in 1:length(average_error)){
      average_error[z] <- mean(ErrorK[,z])
    }
    mean_error[j] <- min(average_error)
    chosen_k[j] <- which.min(average_error)
    k <- a + 1
    cat('\nIn cross validation',j,'k is selected as',chosen_k[j])
  }
}
```

- Sampling_KNN.R

```
library(class)
data <- read.csv('PatientInfo.csv') #state:1 -> recovered, 0->deceased    disease:1 -> true
data <- data[,1:7]
data$sex <- factor(data$sex)
data$country <- factor(data$country)
data$province <- factor(data$province)
data$city <- factor(data$city)
data$disease <- factor(data$disease)
data$state <- factor(data$state)

#_____________________Creating New Observations_____________________
#___________Oversampling_____________
data <- subset(data, select = c(1,2,3,6,7))
library(DMwR)
data_new <- SMOTE(state ~., data, perc.over = 1000)
count0 <- sum(ifelse(data_new$state==0,1,0))
count1 <- sum(ifelse(data_new$state==1,1,0))

data_new$sex <- as.integer(data_new$sex)
data_new$disease <- as.integer(data_new$disease)
data_new$country <- as.integer(data_new$country)



#         _____________ 10 times 5-fold Cross Validation_____________
library(caTools)
set.seed(100)
cv <- 1:10 # 5 times
cv2 <- 1:5 # 5-fold cv
k_val <- 1:50
misclassError <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
accuracy <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
count <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
correct <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
length <- ceiling(dim(data_new)[1]/length(cv2))
k_opt <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
chosen_k <- rep(0,length(cv))
mean_error <- rep(0,length(cv))
for (j in cv){
```

```
k <- 1
ind <- sample(dim(data_new)[1])
data <- data_new[ind,]
cat('\n-------------- CV',j,'-------------- ')
ErrorK <- matrix(rep(0,length(cv2)*length(k_val)),nrow = length(cv2), ncol = length(k_val))
for (i in cv2){
  cat('\n Fold-', i)
  a <- i*length*(i!=length(cv2)) + dim(data)[1]*(i==length(cv2))
  cat('\nData is selected between', k, '-', a)
  index <- k:a
  train <- data[-index,]
  test <- data[index,]
  X.train <- subset(train, select = -c(dim(train)[2]))
  y.train <- train[,dim(train)[2]]
  X.test <- subset(test, select = -c(dim(test)[2]))
  y.test <- test[,dim(test)[2]]

  for (n in k_val) {
    prediction <- knn(train = X.train, test = X.test, cl = y.train, k = k_val[n])
    ErrorK[i,n] <- mean(prediction != y.test)
  }

  k_opt[j,i] <- which.min(ErrorK[i,])
  misclassError[j,i] <- min(ErrorK[i,])

  accuracy[j,i] <- 1 - misclassError[j, i]
  cat('\nk-value which gives the minimum error is', k_opt[j,i])
  cat('\nThe accuracy is', 1- misclassError[j,i])
  cat('\nThe misclassification error is', misclassError[j,i])


  prediction <- knn(train = X.train, test = X.test, cl = y.train, k = k_opt[j,i])
  for (p in 1:length(prediction)){
    if(y.test[p]==0){
      count[j,i] <- count[j,i] + 1
      if (prediction[p]==0){
        correct[j,i] <- correct[j,i] + 1
      }
    }
```

```
    }
    cat('\nOnly', correct[j,i], 'out of',count[j,i], 'class-0 are predicted correctly.')
    average_error <- rep(0,length(k_val))
    for (z in 1:length(average_error)){
      average_error[z] <- mean(ErrorK[,z])
    }
    mean_error[j] <- min(average_error)
    chosen_k[j] <- which.min(average_error)
    k <- a + 1
    cat('\nIn cross validation',j,'k is selected as',chosen_k[j])
  }
}

#_____________________Deleting Some Observations_________________
#___________Undersampling_____________
data <- read.csv('PatientInfo.csv') #state:1 -> recovered, 0->deceased    disease:1 -> true
data <- data[,1:7]
data$sex <- factor(data$sex)
data$country <- factor(data$country)
data$province <- factor(data$province)
data$city <- factor(data$city)
data$disease <- factor(data$disease)
data$state <- factor(data$state)
data <- subset(data, select = c(1,2,3,6,7))

data_new <- SMOTE(state ~., data, perc.under = 500)
count0 <- sum(ifelse(data_new$state==0,1,0))
count1 <- sum(ifelse(data_new$state==1,1,0))
data_new$sex <- as.integer(data_new$sex)
data_new$disease <- as.integer(data_new$disease)
data_new$country <- as.integer(data_new$country)

#         _____________ 10 times 5-fold Cross Validation_____________
library(caTools)
set.seed(100)
cv <- 1:10 # 5 times
cv2 <- 1:5 # 5-fold cv
k_val <- 1:50
```

```
misclassError <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
accuracy <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
count <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
correct <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
length <- ceiling(dim(data_new)[1]/length(cv2))
k_opt <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
chosen_k <- rep(0,length(cv))
mean_error <- rep(0,length(cv))
for (j in cv){
  k <- 1
  ind <- sample(dim(data_new)[1])
  data <- data_new[ind,]
  cat('\n-------------- CV',j,'-------------- ')
  ErrorK <- matrix(rep(0,length(cv2)*length(k_val)),nrow = length(cv2), ncol = length(k_val))
  for (i in cv2){
    cat('\n Fold-', i)
    a <- i*length*(i!=length(cv2)) + dim(data)[1]*(i==length(cv2))
    cat('\nData is selected between', k, '-', a)
    index <- k:a
    train <- data[-index,]
    test <- data[index,]
    X.train <- subset(train, select = -c(dim(train)[2]))
    y.train <- train[,dim(train)[2]]
    X.test <- subset(test, select = -c(dim(test)[2]))
    y.test <- test[,dim(test)[2]]

    for (n in k_val) {
      prediction <- knn(train = X.train, test = X.test, cl = y.train, k = k_val[n])
      ErrorK[i,n] <- mean(prediction != y.test)
    }

    k_opt[j,i] <- which.min(ErrorK[i,])
    misclassError[j,i] <- min(ErrorK[i,])

    accuracy[j,i] <- 1 - misclassError[j, i]
    cat('\nk-value which gives the minimum error is', k_opt[j,i])
    cat('\nThe accuracy is', 1- misclassError[j,i])
    cat('\nThe misclassification error is', misclassError[j,i])
```

```
    prediction <- knn(train = X.train, test = X.test, cl = y.train, k = k_opt[j,i])
    for (p in 1:length(prediction)){
      if(y.test[p]==0){
        count[j,i] <- count[j,i] + 1
        if (prediction[p]==0){
          correct[j,i] <- correct[j,i] + 1
        }
      }

    }
    cat('\nOnly', correct[j,i], 'out of',count[j,i], 'class-0 are predicted correctly.')
    average_error <- rep(0,length(k_val))
    for (z in 1:length(average_error)){
      average_error[z] <- mean(ErrorK[,z])
    }
    mean_error[j] <- min(average_error)
    chosen_k[j] <- which.min(average_error)
    k <- a + 1
    cat('\nIn cross validation',j,'k is selected as',chosen_k[j])
  }
}
```

- PatientInfo_DecisionTree

```
#_____________________________ Decision Tree Model ____________________________________________#
data <- read.csv('PatientInfo.csv') #state:1 -> recovered, 0->deceased    disease:1 -> true
data <- data[,1:7]
data$sex <- factor(data$sex)
data$country <- factor(data$country)
data$province <- factor(data$province)
data$city <- factor(data$city)
data$disease <- factor(data$disease)
data$state <- factor(data$state)
library(tree)
#tree.model <- tree(state ~ sex+disease+birth_year, data = data)
#summary(tree.model)



#_________________ 10 times 5-fold Cross Validation_____________________________________
library(caTools)
set.seed(100)
cv <- 1:10 # 10 times
cv2 <- 1:5 # 5-fold cv
missclassError <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
accuracy <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
count <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
correct <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
length <- ceiling(dim(data)[1]/length(cv2))
for (j in cv){
  k <- 1
  ind <- sample(dim(data)[1])
  data <- data[ind,]
  cat('\n-------------- CV',j,'-------------- ')
  for (i in cv2){
    cat('\n Fold-', i)
    a <- i*length*(i!=length(cv2)) + dim(data)[1]*(i==length(cv2))
    cat('\nData is selected between', k, '-', a)
    index <- k:a
    final.test <- data[index,]
    final.train <- data[-index,]
    tree.model <- tree(state ~ sex+disease+birth_year+country, data = final.train)
```

```
    predictions.prob <- predict(tree.model, final.test)
    predictions <- ifelse(predictions.prob[,1] > 0.5, 0, 1)
    missclassError[j,i] <- mean(predictions != final.test$state)
    accuracy[j,i] <- 1 - missclassError[j, i]
    cat('\nThe accuracy is', 1- missclassError[j,i])
    cat('\nThe missclassification error is', missclassError[j,i])

    k <- a + 1

    for (p in 1:length(predictions)){
      if(final.test[p,dim(final.test)[2]]==0){
        count[j,i] <- count[j,i] + 1
        if (predictions[p]==0){
          correct[j,i] <- correct[j,i] + 1
        }
      }

    }
    cat('\nOnly', correct[j,i], 'out of',count[j,i], 'class-0 are predicted correctly.')

  }
}
```

- Sampling_DecisionTree.R

```
#_____________________________ Decision Tree Model ____________________________________________#
data <- read.csv('PatientInfo.csv') #state:1 -> recovered, 0->deceased    disease:1 -> true
data <- data[,1:7]
data$sex <- factor(data$sex)
data$country <- factor(data$country)
data$province <- factor(data$province)
data$city <- factor(data$city)
data$disease <- factor(data$disease)
data$state <- factor(data$state)
#_____________________Creating New Observations_____________________
#___________Oversampling_________________
data <- subset(data, select = c(1,2,3,6,7))
library(DMwR)
data_new <- SMOTE(state ~., data, perc.over = 1000)
count0 <- sum(ifelse(data_new$state==0,1,0))
count1 <- sum(ifelse(data_new$state==1,1,0))



#         _________________ 10 times 5-fold Cross Validation_________________
library(caTools)
library(tree)
set.seed(100)
cv <- 1:10 # 5 times
cv2 <- 1:5 # 5-fold cv
missclassError <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
accuracy <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
count <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
correct <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
length <- ceiling(dim(data_new)[1]/length(cv2))
for (j in cv){
  k <- 1
  ind <- sample(dim(data_new)[1])
  data <- data_new[ind,]
  cat('\n-------------- CV',j,'-------------- ')
  for (i in cv2){
    cat('\n Fold-', i)
    a <- i*length*(i!=length(cv2)) + dim(data)[1]*(i==length(cv2))
    cat('\nData is selected between', k, '-', a)
```

```
    index <- k:a
    final.test <- data[index,]
    final.train <- data[-index,]
    tree.model <- tree(state ~ sex+disease+birth_year+country, data = final.train)
    predictions.prob <- predict(tree.model, final.test)
    predictions <- ifelse(predictions.prob[,1] > 0.5, 0, 1)
    missclassError[j,i] <- mean(predictions != final.test$state)
    accuracy[j,i] <- 1 - missclassError[j, i]
    cat('\nThe accuracy is', 1- missclassError[j,i])
    cat('\nThe missclassification error is', missclassError[j,i])

    k <- a + 1

    for (p in 1:length(predictions)){
      if(final.test[p,dim(final.test)[2]]==0){
        count[j,i] <- count[j,i] + 1
        if (predictions[p]==0){
          correct[j,i] <- correct[j,i] + 1
        }
      }

    }
    cat('\nOnly', correct[j,i], 'out of',count[j,i], 'class-0 are predicted correctly.')

  }
}



#_____________________Deleting Some Observations_____________________
#___________Undersampling_____________
data <- read.csv('PatientInfo.csv') #state:1 -> recovered, 0->deceased    disease:1 -> true
data <- data[,1:7]
data$sex <- factor(data$sex)
data$country <- factor(data$country)
data$province <- factor(data$province)
data$city <- factor(data$city)
data$disease <- factor(data$disease)
data$state <- factor(data$state)
data <- subset(data, select = c(1,2,3,6,7))
```

```
data_new <- SMOTE(state ~., data, perc.under = 500)
count0 <- sum(ifelse(data_new$state==0,1,0))
count1 <- sum(ifelse(data_new$state==1,1,0))

#         _______________ 10 times 5-fold Cross Validation_____________
library(caTools)
library(tree)
set.seed(100)
cv <- 1:10 # 5 times
cv2 <- 1:5 # 5-fold cv
missclassError <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
accuracy <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
count <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
correct <- matrix(rep(0,length(cv)*length(cv2)),nrow = length(cv), ncol = length(cv2))
length <- ceiling(dim(data_new)[1]/length(cv2))
for (j in cv){
  k <- 1
  ind <- sample(dim(data_new)[1])
  data <- data_new[ind,]
  cat('\n-------------- CV',j,'-------------- ')
  for (i in cv2){
    cat('\n Fold-', i)
    a <- i*length*(i!=length(cv2)) + dim(data)[1]*(i==length(cv2))
    cat('\nData is selected between', k, '-', a)
    index <- k:a
    final.test <- data[index,]
    final.train <- data[-index,]
    tree.model <- tree(state ~ sex+disease+birth_year+country, data = final.train)
    predictions.prob <- predict(tree.model, final.test)
    predictions <- ifelse(predictions.prob[,1] > 0.5, 0, 1)
    missclassError[j,i] <- mean(predictions != final.test$state)
    accuracy[j,i] <- 1 - missclassError[j, i]
    cat('\nThe accuracy is', 1- missclassError[j,i])
    cat('\nThe missclassification error is', missclassError[j,i])

    k <- a + 1

    for (p in 1:length(predictions)){
```

```
      if(final.test[p,dim(final.test)[2]]==0){
        count[j,i] <- count[j,i] + 1
        if (predictions[p]==0){
          correct[j,i] <- correct[j,i] + 1
        }
      }

    }
    cat('\nOnly', correct[j,i], 'out of',count[j,i], 'class-0 are predicted correctly.')

  }
}
```