Patient Survival Prediction

Final Project Report

1st Rana Demir
Artificial Intelligence Engineering
TOBB ETU
Ankara, Turkey
ranademir@etu.edu.tr

2nd Zeynep Meriç Aşık
Artificial Intelligence Engineering
TOBB ETU
Ankara, Turkey
zeynepmericasik@etu.edu.tr

***Abstract*—This document is a report for YAP470 Machine Learning course project. The subject of the project is the prediction of patient survival status by using machine learning. Different pre-processing techniques, models are used to obtain the most accurate prediction. Furthermore, the right metric is the concept that is focused on.**

***Index Terms*—Machine learning, feature selection, correlation, categorical feature, numerical feature, LightGBM, SVM, Streamlit**

I. Introduction

In this project, the problem that will be focused on is to predict survival state of a patient based on their health conditions. Estimating decease may prevent many negative outcomes. Looking through the diseases or other conditions a patient might have, early precautions can be taken to save lives. There are many researches on this issue since machine learning has started to be used in health care [1] in the last few decades. With this project, it is aimed to explore the most essential blood matters or illnesses which affect a patient significantly in terms of that patient's health and provide a support by predicting it beforehand by using a machine learning model. The approach that will be used is to train the data with several models and detect the best one for this case and for the dataset which is chosen.

II. Related Work

The idea of patient survival prediction [2] is leaning onto the same purpose in most of the projects which is detecting a catastrophe before happening. The aim of this project will be developing a system to provide early diagnosis for the patients who are hospitalized and whose blood values are examined. In this way, all types of health centres can make use of this utility. For the purpose of making this utility usable, the summary of the related works is discussed in this part. This process requires analysing the blood matters and physical conditions of the patient that are valid for that timestamp. As a result, this analysis is utilized to predict if the patient has a critical condition or not after this time interval. This operation requires training a model to guess the value according to all the calculated values, and mapping that value to 1 or 0. Since it is decided that the Binary Classification [3] is going to be used, there are many models which are applicable such as Naive Bayes [4], Stratified Logistic Regression [5], Support Vector Machine (SVM) [6], Random Forest [7], Least Square Support Vector Machine [8]. Before going through the model, the data should be manipulated to make it usable for the training model. Non-numerical values should be converted to numerical ones to make them computable. All parts of data may not be correlated with each other, meaning that overall values of the features may not be distinguishable when examined in a plot. In such a case, the features that are irrelevant should be eliminated [9]. The features that are significant to the dataset should be selected to be used. Afterwards, scaling procedure [10] may be done when the data contain scattered values. The most used scaling techniques are Standard Scaling and Min Max Scaling. There is another scaling method called Robust Scaler but, it is not used as frequent as others. In order to use Min Max Scaler, the data should be mostly non distorting. For Standard Scaler, the data should be relatively distributed normally. If there is inadequate information on specific features, bootstrap can be used as a solution as well. Then, the data will be divided into train and test sets in logical proportions. The most frequently used proportions are 0.8 and 0.7 for training, 0.2 and 0.3 for testing respectively. Whilst cross validation [11] can improve the results as it provides data diversity. After scaling operation, a classifier model is applied to the data. For this specific dataset, the commonly used models are Support Vector Machine, Stratified Logistic Regression, Linear Discriminant Analysis [12], and Random Forest. Cox Regression Model [13] is another frequently used one which is solely for survival analysis. Random Forest is preferred when the dataset is large and both classification and regression are needed. Aside from these, XGBoost [14], Logistic Regression [15], LightGBM [16] are also applied as popular. When the model is applied, training will take place by the fitting operation of the present model. The test data will be used for prediction and accuracy will be calculated by its results. Whilst calculation, recall, precision, p value, Gini impurity index, ROC AUC and f1 score are widely preferred metrics for evaluation. It depends on the main purpose. To improve the model performance, some changes can be made on the data such as selected features. Some of the features may be more important and beneficial than the others. Adding and extracting some of them may enhance or deteriorate the

algorithm performance. Another option may be choosing a model that combinates the successful models in the same structure.

III. Overview of Data

The dataset that will be processed is a data that is loaded from a popular website which is used for various competitions regarding tasks such as machine learning, deep learning on various datasets which are challenging in terms of observing and adjusting. This website is called Kaggle and the direct link can be found in [17]. Here, the main task is to make a classification prediction. The value that is aimed to be predict correctly is called "hospital_death". This feature is representing whether the death has occurred for a person or not. Since the required tools and devices in a hospital may be limited which allows for the patients who are in a serious condition to benefit from it, the priorities should be determined carefully. Since a priority given to a patient mistakenly instead of another patient who needs it may cause the loss of a life, this prediction is significant. It is not expected to only trust this prediction, however this model is built to be a supporting block to decide on such significant choices. Before going into detail regarding the condition of a patient, the dataset properties are examined. The dataset is in the form of a csv file and its size is approximately 31.41 MB. When the content of this dataset is investigated, there are a total of 85 features and 91713 rows of data. The type of content consists of integer, floating number, and object data. 7 of them are integer, 7 are object and the remaining 71 are of type floating number. The number of null values in each feature range from 0 (for features such as "encounter_id", "patient_id", "hospital_id") to 9585 (for features "d1_potassium_max" and "d1_potassium_min") except for the feature "Unnamed:83" which consists of only null values. Therefore, this feature will be eliminated. There are 2 features that have distinct values for each row of data and these values have nothing to do with the prediction result. These features are "encounter_id" and "patient_id." These will also be eliminated. Some of the categorical data are considered as numerical data since they have floating number values. When an investigation on these features is made, it is seen that 2 valued floating number data and all integer valued data are in fact categorical data. Hence, all these features mentioned will be considered as categorical data during the process. In order to obtain a better comprehension over the data, features are investigated. Firstly, the feature "hospital_death" has two distinct values which it can be equal to. First one is 1, which means that the death of a person has occurred in that hospital. Secondly, 0 means the death of a person has not occurred yet in that specific time. The aim to be reached is predicting this value based on the condition of a patient in terms of health which will be explained later in this section. After the target feature, it is seen that all the other features have mostly the same key phrases with each other. Depending on this property, others will be explained by using these phrases. A few of them are obvious since they have the explicit name which implies their actual meaning. For instance, "age", "gender", "ethnicity" and "height" are that kind of attributes. "bmi" stands for body mass index. One of the most common key phrases are "d1" and "h1" prefixes. These mean the first 24 hour and the first hour of the patient's unit stay respectively. Therefore, samples are obtained in a way that it provides a daily and an hourly information. Another common one is "apache" prefix or suffix depending on the feature. This implies the condition of a patient in intensive care unit. The features containing "bp" are in the interest of blood pressure. Other than these, "max" and "min" features are representing the minimum and maximum values in the current condition. An example to this is "d1_diasbp_min". This attribute is the patient's lowest diastolic blood pressure during the first 24 hours of their unit stay, either non-invasively or invasively measured. "icu" stands for intensive care unit. Some of the remaining ones are representing the specific diseases such as "cirrhosis." Lastly, some of them are showing the amount of matters in the blood such as potassium and glucose. 83798 samples of the data are tuples with "hospital_death" being 0 and the other 7915 are 1. Therefore, the dataset is obviously an imbalanced dataset which requires a balancing operation since it may be misleading. Another example showing that the data is imbalanced is when the feature "ethnicity" is observed, its most frequent value is approximately 78.26There are some features that are of no help in the process of predicting. The one that was already mentioned is "Unnamed:83". Aside from that, the features with "noninvasive" suffix have the same distribution with the feature that has no "noninvasive" in it. Therefore, these will be eliminated and the corresponding features will be used instead of using them both which will result an unnecessary memory usage and hence, a decrease in effectiveness.

IV. First Attempt

A. Pre-processing

Before splitting the dataset as training and test, some preparations will be made so that the data can be used in its most adaptive form and hence, the predictions are compatible with the actual values. As it was mentioned earlier, the unnecessary features are eliminated from the data. Then, 81 features are left. These are observed in two distinct headings. One is numeric and the other is categorical data. Categorical data is obtained by choosing only the object type of the whole dataset for the first attempt. Hence, the actual categorical data that are of type integer and floating number are missed during this section. Then, without making any elimination in this data, the next step that is taken is to continue with applying label encoding on this data. Afterwards, categorical data is ready to be processed since it has all numeric values as of that moment. For the numeric data, all the features' correlation with each other is computed by using Pearson Correlation [18]. After that, the features reflecting the similar data are detected and one of them is eliminated from each such pair. So, one of them is used instead of using both. The threshold that is decided in this process is 65 percent. When the distribution of the data is examined after the deletion, target value distributions are

similar to each other with target values being positive (1) for approximately 10 percent of the values which are negative (0). The next step to follow is to delete all the null valued rows. Then, 61593 rows are left. With this amount of data, splitting into train and test sets are actualized. The percentage for this operation is 80 percent for train and 20 percent for test set. Scaling transform and fitting are applied on the train set and only transform is applied on the test set. The final obtained data consists of 51 features and is used to proceed with the training and making prediction operations.

B. Experimental Results

In the training part, several machine learning models are trained such as logistic regression, XGBoost and Neural Network [**?**]. Also, there are some ensemble models that are decided to be used to see their advantages above popular machine learning models. These are Random Forest, Bagging [20], Adaboost [21], and LightGBM classifier. First part of the experiments is done using the default parameters of all models. In the end of each training, the test and train results are shown with accuracy, precision, recall, F1-score, and ROCAUC metrics. The models are also checked for overfitting with the display of the distribution of the actual and predicted values for both train and test to compare and evaluate. These processes should be evaluated by taking into consideration that since this is a classification model, the accumulation of the predictions will be always either on 1 or 0. As it is also visible there, the model with the highest accuracy in training is actualized with 93 percent. However, all test results came close to each other with a slight rise in LightGBM Classifier which is 0.5 to 1 percent higher than rest of the predictions. In the test set, there seems to be no lead since all Random Forest, XGBoost, AdaBoost, and LightGBM classifier again rounded on 93 percent on accuracy metric. Nevertheless, as this is a binary classification problem, the right metric might not be accuracy alone. If the distribution of the target data is examined, it can be seen that most of the results lie on 0, ten times of the positive results to be precise. In most of the models the recall value and thus the F1-score results poorly. The reasoning behind this might be that the models predict most of the values 0 as they are in the original dataset whilst positives (1) might be erased incorrectly. However, for the purpose of this project, the aim is to predict patient death which is crucial and represented by 1. Therefore, the recall metric might be telling something important here. As it is examining the quality of the positive rated results, this metric becomes the actual metric that must be paid careful attention to. Besides this issue, ROC-AUC values usually result between 85 to 90 percent on all models. Precision metric has also concluded mostly on 60 percent approximately. Within the unoptimized feature selection neural network resulted very lousy with 50 percent accuracy and 52 percent recall, loss is also very high with 75 percent within a dense algorithm. To check the overfit or underfit state of each model, distribution graphs are used to get clear curves on the issue. The target of the dataset used lies on 0 and 1 values which forms a binary classification problem, so that the lines on the graphs only alter on these points of the horizontal axis. As the data is highly imbalanced, the predicted line rises more than the actual when it encounters a negative (0) value, and acts just the opposite with positive (1) values. However, when the predicted and actual line comes too close or thoroughly on top of each other, that points to overfitting. When the used models are checked, Random Forest noticeably overfits the training data, thus performances way worse with test data. Beside of that, Bagging classifier can also be classified as overfitting since it predicts the training set too well and results poorly on test set. On the other hand, rest of the models, especially the ensemble models, reacts better on test data since training data does not seem to be memorized. To conclude these evaluations that are obtained from only using the default parameters of the models and an imbalanced dataset, still great accuracy is reached even though this is not the product that is aimed to be achieved. As it is crystal clear in Table I, the recall and f1-score metrics has not resulted within the scale expected and has remained low for all models. Only Naïve Bayes has resulted mildly better in recall with 52 percent but it is still not quite the standard anticipated. There are many ways for improvement to solve this problem better using the appropriate evaluation methods and they will be carried on followingly since the issue that is worked on is a very sensitive topic.

TABLE I
TEST RESULTS

Trained Models	**Metrics**			
	Accuracy	***Precision***	***Recall***	***f1-score***
Naive Bayes	0.85	0.29	0.52	0.37
Logistic Regression	0.92	0.65	0.23	0.34
Stratified LR	0.92	0.65	0.23	0.33
SVM	0.93	0.73	0.18	0.29
Linear Discriminant Analysis	0.92	0.54	0.34	0.42
XGBoost Classifier	0.93	0.63	0.29	0.39
Random Forest	0.93	0.74	0.23	0.35
Adaboost Classifier	0.93	0.65	0.30	0.41
Bagging Classifier	0.93	0.64	0.25	0.36
Neural Network	0.63	0.12	0.53	0.19
LightGBM Classifier	0.93	0.70	0.30	0.42

[a]First prediction evaluations for positive (1) values

C. Evaluating the First Attempt

For the first approach that has been applied, there are problems that cannot be underestimated. Since the dataset contains a large number of tuples, it is natural to believe that the predictions are going to be accordingly suitable for the real values. The deficits of this belief can be ordered as following: Firstly, even though the dataset is a large dataset, it does not imply that it will be as representative while actualizing the predictions and training processes. In this survival dataset, it is also the case. The reason that it is not representative and hence, does not work decent for the positive (1) values is because the dataset consists of mostly negative (0) values. These tuples are the information that belongs to the patients who are not in a serious condition as much as the other tuples that contain

the information of patients who died. For that reason, the present dataset states that it is an unbalanced data and it requires a process which can make it more balanced. In this way, predictions can be made more accurately. Furthermore, because of this drawback, the models that are trained such as LightGBM, XGBoost are prone to predict the result negative (0). Since the number of negative (0) values are approximately ten times bigger than the number of positive (1) values, its performance is naturally affected from this. To prevent this from happening, data should be balanced in a way that the numbers do not differ significantly from each other. This definition makes a connotation on down sampling process. Since down sampling enables any dataset to be more balanced by eliminating some rows which consist of the dominant values of that feature, it may be beneficial for this dataset. Even though it may come with its benefits, it may also bring drawbacks with it. These drawbacks are regarding the problem of the way that is chosen to apply the deletion operation on the specific parts of the dataset or doing it randomly all over the dataset. Another solution for an unbalanced dataset is the up-sampling process. By doing this, the number of the positive (1) values that are nearly one tenth of the negative (0) values would increase and these two would be much closer to each other in terms of the amount. Nevertheless, since the positive (1) values are representing the patients who are in a serious condition, augmenting their number would be an unwise solution. The reason is that after reproducing tuples accordingly, the data would be much larger than it already is which means approximately doubling its size. Moreover, a preferred solution that creates a whole model which works efficiently and fast may not be up-sampling. Secondly, even if the dataset were to be a balanced one, the technique that is utilized in order to select the required and sufficient features is not exhaustive since it considers only the features that contain numeric values. The remaining part of the dataset which is categorical dataset should be considered as well. As an addition to these, the property that is solely worked on is the numerical features' Pearson Correlation with each other. In this situation, the relation between the target value and the numerical features are not considered. The other weaknesses in this approach are the nonexistent inspection of the relation between at least two categorical data. What makes this more deficient is the lack of the investigation of the relation between categorical data and the target values. Without all these properties, the dataset cannot be comprehended completely. For instance, there may be categorical features that have the same distribution and hence, can be used in each other's place. Because of this, more data might be processed and therefore, the overall performance that includes the pace of the model training is affected negatively. For this attempt, SVM is the obvious example. Because of the dataset size, SVM works undoubtedly slow. So, an aim that should be reached is to make all the procedures much faster.

V. Second Attempt

A. Pre-processing

The same approach as the previous attempt is followed and the unnecessary features are eliminated from the data. Then, 81 features are left. These are again observed in two distinct headings. After the dataset description and details, categorical data is known to be consisting of all object type data and integer valued data. Also, some of the 2 valued floating number type of data are categorical data. After making this distinction correctly, the requirement of eliminating more features is realized since the data is still big. Two different techniques are used for each of numeric and categorical data. When deleting or adjusting the numeric features, the technique that is used is observing their correlation with each other. When a high percentage is obtained between two features, then one of them can be used by representing both. However, before moving in this direction, eliminating the features that have less impact on predicting the target feature is actualized. In other words, the relation between features and target feature is computed. This computation is based on the chi square values. Since the features are numeric and the target feature is a categorical value, correlation cannot be found. Hence, to obtain the relation between them, chi square value is observed. If it is high, then this tells the model that the difference between the features cannot be underestimated. Therefore, the preferred features are the ones with small chi square values. Others are deleted and will be used neither in training nor in test data. Afterwards, according to the relation between each remaining feature, it is decided to combine them into one feature. The ones with a high correlation which is over 70 percent, are combined after checking their significance to whole model. Here, the combined features are as following: "d1_temp_min" with "temp_apache", "h1_heartrate_max" with "heart_rate_apache", "h1_mbp_max" with "h1_diasbp_max", "h1_mbp_min" with "h1_diasbp_min". The combination process is actualized by giving the both features 11 weights ranging from 0 to 1 by the value 0.1. Each of these 11 combinations' chi square value with the target feature is computed and the one with the lowest score is chosen as the new feature. Then, this is used instead of the 2 component features. The obtained new features from this process are "temperature", "heartrate", "mbp_diasbp_max" and "mbp_diasbp_min" respectively. Moving onto the categorical data, before making any updates, label encoding is applied on this data to be able to make any computation. After obtaining the numeric values out of the categorical data, polychoric correlation between them is computed and one of the high correlated ones are eliminated. As a result, "apache_3j_bodysystem" and "apache_post_operative" are dropped. Then, the remaining attributes' polychoric correlation with the target feature is computed and a threshold of 0.01 is used to do further elimination. After completing all the operations on the both data, there are 47 features left.

Then this dataset is down sampled, splitted into training and test data and Standard Scaling is applied.

B. Experimental Results

In the training part, several machine learning models are trained such as logistic regression, XGBoost and Neural Network. Also, there are some ensemble models that are decided to be used to see their advantages above popular machine learning models. These are Random Forest, Adaboost, and LightGBM classifier. Second part of the experiments is done using again the default parameters of all models. In the end of each training, the test and train results are shown with accuracy, precision, recall, F1-score, and ROCAUC metrics. The models are also checked for overfitting with the display of the distribution of the actual and predicted values for both train and test to compare and evaluate. These processes should be evaluated by taking into consideration that since this is a classification model, the accumulation of the predictions will be always either on 1 or 0. As it is seen on Table II, the accuracy dropped to 80-82 percent since the last experiments. The model with the highest accuracy in training is actualized with 82 percent. However, all train results came close to each other with a slight rise in XGBoost Classifier which is 1 to 2 percent higher than rest of the predictions. In the test set, there seems to be no lead since all Random Forest, SVM, XGBoost, and LightGBM classifier again rounded on 82 percent on accuracy metric. Nevertheless, because most of the results lie on 0, the dataset is down sampled in the preprocessing part. In most of the models while the accuracy has decreased, the recall value and thus the F1-score resulted better than before with around 70 percent which was the aim of the second attempt. Since the proportion of 0 and 1 values are closer hence data is more balanced, the positives are predicted more accurately. Besides this, ROC-AUC values have not changed drastically, so they are still between 85 to 90 percent on all models. Precision metric has also concluded better between 70 to 75 percent. Within the optimized feature selection neural network resulted very lousy again with 43 percent accuracy but 71 percent recall which is as decent as the other models, loss has also dropped to 70 percent within a dense algorithm. To check the overfit or underfit state of each model, distribution graphs are used to get clear curves on the issue. The data is more balanced now, however, the predicted line still rises more than the actual when it encounters a negative (0) value, and acts just the opposite with positive (1) values. But, when the predicted and actual line comes too close or thoroughly on top of each other, that points to overfitting. When the used models are checked, Random Forest noticeably overfits the training data, thus performances way worse with test data. Beside of that, Adaboost classifier can also be classified as overfitting since it predicts the training set too well and results worse on test set than the other ensemble models. On the other hand, rest of the models, especially the ensemble models, reacts better on test data since training data does not seem to be memorized. To conclude these evaluations that are obtained from using a different method for preprocessing and a better-balanced dataset, not a great accuracy is reached but there is no problem because this is not the product that is aimed to be achieved. Recall has increased as it was the reason a second attempt has been made. There is still room for improvement so in the next attempt the best parameters will be checked to get even more successful outcomes. In Table II, the test results for all models has been compared and best models will be chosen to be used for hyperparameter tuning.

TABLE II
TEST RESULTS

Trained Models	Metrics			
	Accuracy	***Precision***	***Recall***	***f1-score***
Naive Bayes	0.76	0.66	0.60	0.63
Logistic Regression	0.80	0.77	0.58	0.67
Stratified LR	0.80	0.77	0.58	0.67
SVM	0.82	0.79	0.62	0.69
Linear Disc Analysis	0.80	0.78	0.57	0.66
XGBoost Classifier	0.82	0.77	0.66	0.71
Random Forest	0.82	0.78	0.64	0.70
Adaboost Classifier	0.81	0.74	0.63	0.68
Neural Network	0.57	0.33	0.23	0.25
LightGBM Classifier	0.82	0.76	0.66	0.71

[a]Second prediction evaluations for positive (1) values

VI. Evaluating the Second Attempt

As it was told in the first attempt's evaluation part, a more comprehensive feature selection approach is assimilated. It includes computing chi square values and Polychoric Correlation as addition to Pearson Correlation. Down sampling operation is applied to the dataset and processing times of the distinct models are reduced. These two add up to generate a more successful model which produces healthier recall and precision scores. By virtue of this, the result that is aimed to be reached is nearer.

VII. Hyperparameter Tuning

Some of the best models has been tried to optimized with RandomizedSearch such as LightGBM, XGBoost, AdaBoost, and Random Forest classifier. Even though Random Forest overfits, it does not perform too lousy so it is still in the best models. When the test results are evaluated, it is indicated in the graphs that Adaboost classifier has overfitted the train data, hence has the lowest recall value. On the contrary, XGBoost has come on lead with 68 percent and LightGBM followed with 67 percent on recall metric. Random Forest also performed well with 65 percent recall value. LightGBM's best parameters are 'binary' on objective, 100 on num_leaves, 50 on n_estimators, 50 on min_child_samples, 0.05 on learning_rate, and 'goss' on boosting_type. As this model also seems to do better on the other metrics, it is chosen for best 5 features experiment. SVM's best parameters are 'kernel' on linear, 'rbf' on poly, 'degree' on rbf, and 2 on sigmoid. In Table III, all of the models' evaluation metrics can be compared to notice that Support Vector Machine is the best model to use for best features evaluation and then for the web app. However, the model is very time consuming than all the

best models and hence will not be used going forward. Instead, it is decided to train LightGBM classifier. Its evaluation metric calculations are a bit lower than SVM, but still has very good results and most importantly produces a high recall value.

TABLE III
TEST RESULTS

Trained Models	Metrics			
	Accuracy	***Precision***	***Recall***	***f1-score***
XGBoost Classifier	0.82	0.77	0.66	0.71
SVM	0.82	0.79	0.62	0.69
Adaboost Classifier	0.75	0.62	0.62	0.62
LightGBM Classifier	0.81	0.77	0.64	0.70
Random Forest	0.81	0.77	0.61	0.68

[a]Evaluations after optimization for positive (1) values

VIII. BEST FEATURES OF LIGHTGBM WITH SHAP

SHAP is a mathematical method to explain the predictions of machine learning models by calculating the contribution of each feature to the prediction. To find the best features with the optimized parameters of LightGBM classifier model, SHAP is used. The five best features obtained by this method are listed as 'apache_4a_icu_death_prob', 'apache_4a_hospital_death_prob', 'd1_spo2_min', 'd1_sysbp_min', and 'age'. The metrics of prediction has not appeared to alter noticeably with 80 percent of accuracy, 76 percent of precision, 60 percent of recall, and 67 percent of f1-score, however, these five features are very important for the next steps of the project. The model is dumped with Pickle to load in the front-end script and be used for real-time prediction.

IX. DEPLOYMENT

In the deployment stage, Streamlit is the module that has been used. The goal was to provide a practical way for doctors to sense danger solely based on the blood test results even before examining the patient. With the reduced feature number by best 5 features found by SHAP of LightGBM classifier, this app only requires the age, APACHE IVa score of in ICU mortality, APACHE IVa score of in-hospital mortality, the lowest peripheral oxygen saturation during the first 24 hours, and the lowest systolic blood pressure during the first 24 hours. The results can also be uploaded as a csv file in the given space, also an example file is dropped to show the appropriate format. The values can also be filled in the form below the CSV space one by one. Then, the filled values can be seen on the Patient Health Information data frame. And just below that there are the prediction results which is either 'Survival' or 'Mortality'. As the audience of this app is doctors and not patients, this piece of information can be displayed frankly which would not be very ethical for patients to see directly.

X. DISCUSSION

Comparing the two approaches in terms of memory usage and performance quality will be discussed. For the first approach which was based on only using Pearson Correlation, the feature selection is only for the numeric values. However, for the second case, along with the Pearson Correlation, chi square values and Polychoric Correlation are also computed to make it possible to select features from categorical features as addition to the numeric ones. This improves the feature selection approach. Another difference in two approaches is down sampling process. By courtesy of this, both all processes until obtaining the results accelerate and the imbalance of the dataset is eliminated compared to the previous version. Since the dataset mainly consists of negative (0) values, the predictions are inclined to be negative (0). Therefore, the effect of this weakness is reduced. Aside from the comparison between two approaches there are also improvements that can be beneficial. As it was mentioned for a few times already, recall score is the main score to focus on. In order to increase the recall score, correctly predicted positive (1) values must increase. So, in order to improve the model further, different feature selection techniques can be considered to use. More data with much more positive (1) values can be collected in order to have a balanced data. An idea that is open to discussion is to adjust the models' threshold value which they use in the classification process. Aside from the prediction results, prediction probabilities are also accessible for all models, and when it is examined, two results exist for any target value. First one is the probability that the target value is negative (0) and the second is the probability of the target value being positive (1). So, these two probabilities sum up to 1. The default threshold value for these probabilities can be adjusted to obtain a higher recall value. Even though it would escalate, a drawback would also occur. By arranging the threshold in this way, the tendency of predicting the positive (1) values rises and this causes for the precision score to reduce. Accepting such a solution is arguable depends on the problem. Here, both recall and precision scores are significant for the model. Therefore, none of them to be eliminated is the decision that is made here.

XI. RESULT

The key findings of this project can be arranged as following: The dataset which is worked on consists of large amount of data which means 91713 rows of information. Therefore, the representativity of it is much more when compared to a lower sized dataset. On the other hand, because of the same reason, there are problems in terms of efficiency. So, the dataset should be examined in a lower sized version of it. For that reason, features are eliminated and some of them are combined into a single feature using Pearson Correlation and Polychoric Correlation. Then, Label Encoding is applied on the categorical data, then for all the data, Standard Scaling is applied. In order to deal with a lower size, down sampling is applied on the dataset. It is ready for training. It is mainly trained by using ensemble models such as XGBoost, LightGBM and AdaBoost. The performance of recall and precision scores are approximately 67With this result obtained, good scores that could be beneficial for patients and hospital system are provided. Without focusing only on these numbers and tak-

ing them as a supportive tool, the patient in serious condition can be detected and treatments can be done accordingly.

REFERENCES

[1] Irene Y. Chen, Emma Pierson, Sherri Rose, Shalmali Joshi, Kadija Ferryman, and Marzyeh Ghassemi. Ethical Machine Learning in Healthcare. Annual Review of Biomedical Data Science. Vol. 4:123-144.

[2] Hardy JR, Turner R, Saunders M, A'Hern R. Prediction of survival in a hospital-based continuing care unit. Eur J Cancer A 1994;30(3):284–8.

[3] Roshan Kumari, Saurabh Kr. Srivastava. Machine Learning: A Review on Binary Classification. International Journal of Computer Applications (0975 – 8887). Volume 160 – No 7, February 2017.

[4] International Journal of Applied Mathematics and Computer Science. 2013. Vol. 23, no. 4. 787-795.

[5] A. J. Scott and C. J. Wild. Fitting Logistic Regression Models in Stratified Case-Control Studies. Vol. 47, No. 2 (Jun., 1991), pp. 497-510.

[6] A. Mathur and G. M. Foody. Multiclass and Binary SVM Classification: Implications for Training and Classification Users. IEEE Geoscience and Remote Sensing Letters, Vol. 5, No. 2, April 2008.

[7] Breiman L: Random Forests. Machine Learning 2001, 45:5-32.

[8] Suykens, J., Vandewalle, J. Least Squares Support Vector Machine Classifiers. Neural Processing Letters 9, 293–300 (1999).

[9] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu. 2017. Feature Selection: A Data Perspective. ACM Comput. Surv. 50, 6, Article 94 (December 2017), 45 pages.

[10] Ahsan, M.M.; Mahmud, M.A.P.; Saha, P.K.; Gupta, K.D.; Siddique, Z. Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance. Technologies 2021, 9, 52.

[11] Arlot S, Celisse A. A survey of cross-validation procedures for model selection. Stat Surv 2010;4:40.

[12] S. J. D. Prince and J. H. Elder, "Probabilistic Linear Discriminant Analysis for Inferences About Identity," 2007 IEEE 11th International Conference on Computer Vision, 2007, pp. 1-8, doi: 10.1109/ICCV.2007.4409052.

[13] Altman, D.G. and Andersen, P.K. (1989), Bootstrap investigation of the stability of a cox regression model. Statist. Med., 8: 771-783.

[14] Z. Chen, F. Jiang, Y. Cheng, X. Gu, W. Liu and J. Peng, "XGBoost Classifier for DDoS Attack Detection and Analysis in SDN-Based Cloud," 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), 2018, pp. 251-256, doi: 10.1109/BigComp.2018.00044.

[15] Ana M Bianco and V´ıctor J Yohai. Robust estimation in the logistic regression model. Springer, 1996.

[16] Dehua Wang, Yang Zhang, and Yi Zhao. 2017. LightGBM: An Effective miRNA Classification Method in Breast Cancer Patients. In Proceedings of the 2017 International Conference on Computational Biology and Bioinformatics (ICCBB 2017). Association for Computing Machinery, New York, NY, USA, 7–11.

[17] https://www.kaggle.com/datasets/mitishaagarwal/patient

[18] Benesty, J., Chen, J., Huang, Y., Cohen, I. (2009). Pearson Correlation Coefficient. In: Noise Reduction in Speech Processing. Springer Topics in Signal Processing, vol 2. Springer, Berlin, Heidelberg.

[19] Ahmed, F.E. Artificial neural networks for diagnosis and survival prediction in colon cancer. Mol Cancer 4, 29 (2005).

[20] Hothorn, T., Lausen, B., Benner, A. and Radespiel-Tröger, M. (2004), Bagging survival trees. Statist. Med., 23: 77-91.

[21] A. Safiyari and R. Javidan, "Predicting lung cancer survivability using ensemble learning methods," 2017 Intelligent Systems Conference (IntelliSys), 2017, pp. 684-688, doi: 10.1109/IntelliSys.2017.8324368.