**Applied Data Analysis Final Project Report**

**Prediction of Readmission & Death of Patients**

1. **Introduction**

Physicians who care for critically ill patients often rely on subjective criteria to determine when a patient is ready to be discharged from the intensive care unit (ICU). Discharging patients “quicker and sicker” has the potential to increase the frequency of ICU readmissions as well as death, but discharging patients too slowly may provide little or no benefit and increase costs. Thus, it is critical to predict the outcome of an ICU discharge. In this project, we are given with a dataset which includes clinical and demographic information about patients. Moreover, we are asked to predict their future readmission & death situation.

1. **Data Evaluation & Preprocessing**

We have information of 16059 patients with 51 feature about them, such as their age, sex, duration of stay in ICU and some other clinical data, etc. First of all, we checked if there is any NA value on the data, and results showed that there isn’t NA data.

Then, we created histogram of the data to see how data is distributed. Results showed that data is totally imbalanced. Especially, ratio of positive and negative classes is far from 1 for each, readmission and death. While there are 13584 patients who are not readmitted, there are just 2475 patients who are readmitted. On the other hand, despite the fact that there is 12976 patients who are still alive after discharge, there are just 3083 patients who are dead. This unbalance is extremely important when classification is the case. If our model will just classify every patient as “Readmitted”, model’s accuracy will be almost 85%. Almost same conditions for “Death” classification. Hence, we decided to use recall score as our performance measure. Because, we want to be precise at predicting all patients who is going to be readmitted & death.

In the Part C, we checked data types in our dataset and found out that there are 3 types of data; integer, float and categorical data. For some algorithms, it is not possible to use categorical data, so, OneHotEncoding is used to let the algorithm use these categorical data. These categorical date were; 'ICU Type', 'Origin Level of Care', 'Sex' and 'Type'. All these features has more value type than 2, but ‘Sex’, so, we used dummy variable to represent values, such as 1 for “Female and 0 for “Male”. Then, we applied OneHotEncoder to encode our categorical data and had the final version of data set. After preprocessing, data is ready to be split as training and test data. At the end of these applications, we have ‘x\_train’, ‘y\_train\_readmitted’ and ‘y\_train\_death’ data and same groups for test set. Then as final of data evaluation, we did feature selection. Feature selection showed us there are features which are not related with target so we arranged our training test for readmission and death.

Data is ready to be processed by Machine Learning models.

1. **Training The Models**

We decided to use 7 different models and compare them.

1. **Decision Tree Classifier**

Readmission: We run a grid search with cross-validation to find out best parameters for Decision Tree Classifier. Best parameters gave us not the best results. We got 0.12 recall score.

Death: With grid search we got our best parameters and recall score was 0.62 which is quite high compared to readmission recall score. We believe the reason behind this difference between feature numbers.

1. **Random Forest Classifier**

Since Random Forest is quite time taking and need more computational power, we could not run the grid search for this approach.

Readmission: We used cross validation score function as see how random forest did on our training set of readmission and as the results came out we saw that recall score is quite low. So, random forest was not good at picking the readmitted patients.

Death: We used same approach for death training set and we got 0.63 recall score. However this result did not surprise us.

1. **Logistic Regression**

We used grid search for logistic regression to find out what kind of penalty we should use and what C value should get.

Readmission: Best parameters for logistic regression on readmission data gave us really bad recall score. Probably reason behind this is that our boundary line is not linear and that’s why logistic regression is not good at classifying correctly and picking up the readmitted people.

Death: Best parameters for death data set was l1 penalty and 10 for C value. Again like other, death prediction was quite better then readmission prediction. We gained 0.66 recall score.

1. **AdaBoost Classifier with Decision Tree Classifier**

We used Decision Tree with max depth of 1 as base estimator for AdaBoost Classifier and used cross validation score as criteria.

Readmission: AdaBoost is sensitive model for misclassified instances so our model did little bit better than previous models on readmitted data set and gave us 0.20 recall score as average of 5 fold cross validation.

Death: While AdaBoost gave us better results on readmission data, it did not give us the best result for death prediction. We got 0.64 recall score for death recall score.

1. **Multi-Layer Perceptron**

We used grid search for MLP and tried to see best number of hidden layers, alpha value and max iteration number with 0 as random state value.

Readmission: Since MLP is quite precise approach for non-linear dataset, MLP with best parameters gave us the best recall score on validation processes which is 0.40.

Death: As AdaBoost, MLP did not improve our death recall score with its best parameters and gave us 0.65 recall score.

1. **XGBoosting Classifier**

We gave the parameters, which we saw an example on the internet, to the XGBoosting classifier.

Readmission: XGBoosting was a disappointment for us since it did not give and good results on readmission data. We got average recall score of 5-fold cross validation 0.08.

Death: As other approaches, this approach did almost same recall score on death prediction.

1. **Testing Models**

After testing all the models which are mentioned above on test data after training on the full train dataset, some results was surprising. Our maximum recall score on readmission dataset was 0.61 by XGBoosting. We could not totally understand why it gave the best results on the test set while giving not the best, not even close, on training set. On the other hand maximum on death data set recall score was again XGBoosting, 0.85. However, other approaches was not that far from XGBoosting on death dataset.

1. **Conclusion**

In this project we got a dataset from a hospital which shows us the readmission and death situation of patients with bunch of features about them. The challenging part about this project was that dataset was not balanced and even if our models classifies each instance as 0 our accuracy could be around 85%. First of all we arrange the data set on the way that our models can be applied. After that we used simple and not-simple approaches on the data to predict our targets. Since the boundary was not linear, simple models were not doing quite good jobs. To find the best models we used grid search method and got our best parameters. After all this process XGBoosting classifier did a good job on data sets for classifying correct. We wanted to focus on recall score because human life is more important than hospitals profit. And, since the data was imbalanced it was not easy to get high recall scores, however for us, 0.61 was satisfying after seeing how bad other approaches were.