**Project Summary**

**on**

**Predicting 30-Day**

**Hospital Readmissions**

**for Diabetic Patients**

**Group 12:**

* **Ruksar Lukade  - 37**
* **Alexandre Broggi - 12**
* **Mousami Biswas  - 11**

**Project Description**

Hospital readmission is an indicator of the quality of care and is a driver for the increasing cost of healthcare. Like other chronic diseases, Diabetes is associated with a higher risk of hospital readmission. In this research, we evaluate several machine learning approaches to predict the probability of hospital re-admissions for diabetic patients. We leverage several pre-processing techniques and investigate the performance of the various models. The significant variables contributing to the analysis are the number of in-patients, length of stay, number of medications, number of diagnoses, and age. The results demonstrate the viability of the techniques in providing a better understanding of factors influencing hospital readmission.

**Tools & Technologies used**

**Programming Language:** Python

**IDE:** Jupyter Notebook

**Visualization:** Python (Matplotlib and Seaborn)

**Models:** XGBoost, Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, Artificial Neural Network, K Nearest Neighbors,

**Problem Statement**

To identify the factors that lead to the high readmission rate of diabetic patients within 30 days post discharge and correspondingly to predict the high-risk diabetic-patients who are most likely to get readmitted within 30 days so that the quality of care can be improved along with improved patient’s experience, health of the population and reduce costs by lowering readmission rates. Also, to identify the medicines that are the most effective in treating diabetes.

**Dataset**

The data set used for this study contains information about clinical care for patient care over the years 1999 to 2008 which has 100k entries from 150 hospitals across the United States. The dataset contains 50 features, most of which can be used for our research. These features contain both ordinal and non-ordinal data.

The information was collected with the following five filters, as described by the original creators of the dataset:

* It is an inpatient encounter (a hospital admission).
* It is a diabetic encounter, that is, one during which any kind of diabetes was entered into the system as a diagnosis.
* The length of stay was at least 1 day and at most 14 days.
* Laboratory tests were performed during the encounter.
* Medications were administered during the encounter.

**Data Pre-processing**

In the dataset cleaning process, a 5-step approach was employed:

* Blank Values Definition:

Blank values were defined, with "?" considered as blank and "None" as a valid value.

* Ordinal Data Handling:

Dictionaries were created to map ordinal data to numerical values.

* ID Column Removal:

Identification (ID) columns were dropped from the dataset.

* Ordinal Value Mapping:

Ordinal values were mapped to their corresponding numerical representations.

* Non-Ordinal Data Transformation:

Pandas.get\_dummies() was used to create binary columns for non-ordinal data.

This process resulted in a dataset with 96 columns. The target feature was converted to binary during the ordinal-to-numerical mapping in step 5.

**Exploratory Data Analysis**

In the initial data exploration, various graphs and charts were employed to comprehend the dataset's relationship with the problem. This included:

* Correlation Matrix Heat Map:

A visual representation of the correlation between different variables using a heat map.

* Box and Whisker Chart:

A graphical display of the distribution of data, showing the median, quartiles, and potential outliers.

* Histogram:

A graphical representation of the distribution of a dataset, displaying the frequency of different values.

* Scatter Plot:

A visualization showing the relationship between two variables.

The analysis revealed dataset imbalance. To address this, under-sampling was chosen for algorithm analysis. For full deployment, oversampling methods would be considered to handle class imbalance more effectively.

**Undersampling**

A screenshot of a computer

Description automatically generated

**Feature Selection for Model Performance**

To optimize model performance, the focus is on selecting data that correlates with the target feature, specifically whether a patient returned within 30 days.

* Columns with minimal variation are removed. Those with fewer than 10 instances where they are not their most common value are dropped. This step reduces the dataset to 78 columns.
* The data is split into training and testing sets during the feature selection process.
* A validation set is omitted in this approach. The decision might be influenced by the dataset's characteristics or specific considerations of the modeling task.

By carefully selecting features and splitting the data, the goal is to enhance the model's ability to capture relevant patterns and improve its predictive performance on the target variable.

**Model Building**

**XGBoost Classifier:** XGBoost stands for extreme Gradient Boosting. XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solves many data science problems in a fast and accurate way.

**Logistic Regression:** Logistic regression is the classification counterpart to linear regression. Predictions are mapped to be between 0 and 1 through the logistic function, which means that predictions can be interpreted as class probabilities. The models themselves are still “linear,” so they work well when your classes are linearly separable (i.e. they can be separated by a single decision surface.

The logistic regression method assumes that:

• The outcome is a binary or dichotomous variable like yes vs no, positive vs negative or 1 vs 0.

• There is a linear relationship between the logit of the outcome and each predictor variables. (Recall that the logit function is logit(p) = log(p/(1-p)), where p is the probabilities of the outcome.)

• There are no influential values (extreme values or outliers) in the continuous predictors.

• There is no high intercorrelations (i.e. multicollinearity) among the predictors.

**Random Forest:** Random Forest is an ensemble learning algorithm used for both classification and regression tasks. It constructs multiple decision trees during training and merges their predictions to improve accuracy and control overfitting. Each tree is built on a random subset of the training data, and the final prediction is a combination of individual tree outputs. The randomness introduced in both data and features enhances robustness, making Random Forest a powerful and versatile machine learning algorithm.

**Decision Trees:** A Decision Tree is a simple representation for classifications and regression. It is Supervised Machine Learning where the data is continuously split according to a certain parameter. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches. Leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. In Decision Tree as we have no probabilistic model, but just binary split, we don’t need to make any assumption at all.

**Artificial Neural Network (ANN):** An Artificial Neural Network (ANN) is a machine learning model inspired by the structure and functioning of the human brain. It consists of interconnected nodes, or artificial neurons, organized in layers. Information is processed through these layers, with each connection having an associated weight. ANN learns by adjusting these weights during training, enabling it to recognize patterns, make predictions, and perform tasks such as classification or regression. The hidden layers in ANNs allow them to capture complex relationships in data, making them particularly effective for tasks with intricate patterns and large datasets.

**K Nearest Neighbors (KNN):** K Nearest Neighbors (KNN) is a simple and intuitive machine learning algorithm used for both classification and regression tasks. In KNN, predictions are made based on the majority class (for classification) or the average of neighboring points (for regression) among the k-nearest data points to the input sample in the feature space. The algorithm relies on the assumption that similar instances tend to exist in proximity in the feature space. The value of k, the number of neighbors considered, is a crucial parameter that influences the model's performance. KNN is non-parametric and instance-based, making it flexible but potentially sensitive to noise in the data.

**Model Evaluation**



**Conclusion**

Hospital readmission is an important contributor to total medical expenditure and is an emerging indicator of quality of care. It is disruptive to patients and costly to healthcare systems.

The objective of this project was to develop a predictive risk model to identify patients with diabetes who are at a high risk of hospital readmission. This is done by analyzing key factors using machine learning methods and through retrospective analysis of patients’ medical records which impact the all-purpose readmission of a patient with diabetes within 30 days of discharge and comparing different classification models that predict readmission and evaluating the best model.

In this project, the problem of predicting the risk of readmission was framed as a binary classification problem and several available prediction models were developed and evaluated. This study has assessed how various data preprocessing techniques such as feature selection, missing value imputation and class balancing techniques may impact the results of prediction modeling using readmission for patients with a diabetes diagnosis as the context for the analysis.

Then, various predictive models like Logistic Regression and Decision Tree were applied to this improved dataset (after pre-processing) to obtain risk of readmission predictions accuracy. The impact of different pre-processing choices was assessed on various performance.

This study offers empirical evidence that most proposed models with selected pre-processing techniques significantly outperform the baseline methods (without any pre-processing) with respect to selected evaluation criteria. In this study, we evaluated various machine learning models to predict readmissions of high-risk patients. Extending prior research, we performed class balancing considering the skewness of data. Some of the key features that drove readmissions are number of preceding year visits, length of stay, number of medications and amount of diagnosis. Extending this research, we plan to further investigate the performance of classifiers with the goal of improving the accuracy of prediction of readmission risk.

**References and Links**

* Beata Strack, Jonathan P. DeShazo, Chris Gennings, Juan L. Olmo, Sebastian Ventura, Krzysztof J. Cios, and John N. Clore, “Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records,” BioMed Research International, vol. 2014, Article ID 781670, 11 pages, 2014.

Website: https://www.hindawi.com/journals/bmri/2014/781670/

* The Hospital Readmissions Reduction (HRR) Program, Center for Medicare and Medicaid Services.

Website: <https://www.cms.gov/Medicare/Quality-Initiatives-Patient-AssessmentInstruments/Value-Based-Programs/HRRP/Hospital-Readmission-ReductionProgram>

* Diabetes 130-US hospitals for years 1999-2008 Data Set provided by Centre for Clinical and Translational Research, Virginia Commonwealth University.

Website: https://archive.ics.uci.edu/ml/datasets/diabetes+130- us+hospitals+for+years+1999-2008