CSC 215-01 Artificial Intelligence

Project 1: Predicting Acute Kidney Injury after Liver Cancer Resection

Name	**Meghana Rao Kanneganti, Nikitha Mandhana**
ID	**302804645, 303315350**
Course Title	**CSC- 215, Artificial Intelligence**
Assignment ID	**Project 1**
Due Date	**February 20, 2023**

Problem Statement :

The problem addressed in the research paper is the development of machine learning models, such as Logistic Regression, K-nearest neighbors (KNN) and support vector machines (SVM), to predict the risk of acute kidney injury (AKI) following liver cancer resection.

AKI is a common and potentially life-threatening complication following liver surgery, and early identification of patients at high risk for developing AKI is crucial for timely intervention and improved patient outcomes. The study aims to develop accurate predictive models using preoperative clinical and laboratory variables, and to compare the performance of Logistic, KNN, SVM and Neural networks in predicting AKI after liver cancer resection. The development of accurate predictive models could improve clinical decision-making and aid in the identification of patients who may benefit from more intensive monitoring or preventative interventions.

Methodology :

The research paper used a retrospective cohort of patients who underwent liver cancer resection at a single institution between January 2008 and December 2017. The study included 1,110 patients, of whom 180 developed acute kidney injury (AKI) within 7 days after surgery.

The study used different machine learning models - logistic regression, K-nearest neighbors (KNN), and support vector machines (SVM) etc.,- to predict the risk of AKI after liver cancer resection. The models were developed using a training dataset and were validated using a separate validation dataset. The performance of the models was evaluated using various metrics, including accuracy, F1 score, Recall and area under the receiver operating characteristic curve (AUC-ROC).

Experimental Results and Analysis :

On performing training and validation for all the models using some additional performance metrics, such as normalizing the values, performing one hot encoding, scaling the features and doing component analysis we could make out that different models give different accuracy results. The results and analyses are as follows :

Results :

Machine learning model	Accuracy
Logistic Regression	0.70
K Nearest Neighbor	0.78
Support Vector Machine	0.90
Fully Connected Neural networks	0.95

Analyses :

According to the metrics we used, we could see that best accuracy obtained was by neural networks when compared to all the other models. However, we cannot conclude this model is the best working as different machine learning models can vary depending on the specific dataset and features used.

Task Division and Project Reflection :

As a team we took up the responsibility of both of us working on all the models so that we learn from one another. We started working one by one on each model and comparing our accuracy results to check who achieves the highest accuracy and we learned the different techniques/metrices each one of us applied to get the best results.

We faced challenges while dealing with the accuracy values of neural networks and using tensor flow. Initially it was difficult for us to achieve a good accuracy, however we learned how to use the different metrices and assign parameters and train and fit the model in the best way possible to achieve the results.

Additional Features :

We created a more balanced dataset by using oversampling and under sampling techniques. In this paper, after implementing all the models we found out that we have an imbalanced dataset where the AKI class (class 1) has very few samples compared to the NON-AKI class (class 0).

We used Logistic Regression model in order to perform undersampling, oversampling and figure about the top 5 features. We trained and validated the model, and we observed that the accuracy of logistic regression model increased from 0.65 to ~0.70

Additional feature Name	Accuracy	Implementation
UnderSampling	0.71	Using sample() function
OverSampling	0.74	Using SMOTE
Feature Importance Analysis	0.70	Using SelectFromModel