INFO 501: Python & Data Mining

Enhancing Prehospital Triage Efficacy through Machine Learning

Team:
Vismay Parekh
Priyesh Chaurasia
Abdullah Mamun
Jay Desai

Introduction

Triage is the process of managing medical demand by assessing the symptoms of incoming patients and scheduling their treatments in order to get the best possible outcome from a flow of patients.

The **Korean Triage and Acuity Scale (KTAS)** is a standard scale used for classifying patients for triage.

The project aims to study the impact of various machine learning techniques to improve the accuracy of triage as compared to conventional methods.

Objective

- Evaluate **13 different machine learning algorithms** for predicting KTAS scores.
- Develop a custom accuracy metric.
- Improve prehospital triage efficiency.

Dataset

- The data used in this project was obtained from an anonymized dataset released by the authors of a research paper named **‘Triage accuracy and causes of mistriage using the Korean Triage and Acuity Scale’**

- The original dataset comprises 24 variables including basic patient info, vital signs (systolic blood pressure, diastolic blood pressure, heart rate, respiratory rate, and body temperature), chief complaints, mental state of the patient, pain description, etc.

Data Preprocessing

- Translation of Korean entries into English using Google Translate function in Google Sheets.
- Imputation of missing values and normalization of numerical data.
- One-hot encoding for categorical variables.
- Bag-of-words model and scalar vector decomposition for text data.

Methodology

KTAS scores are discrete integers ranging from 1 to 5.

Since the problem was a **hybrid of regression model and categorical target feature**, a custom accuracy metric was developed.

A predicted KTAS score is considered correct if it is found to be in the range of [(true_score + 1.6) to (true_score - 1.2)].

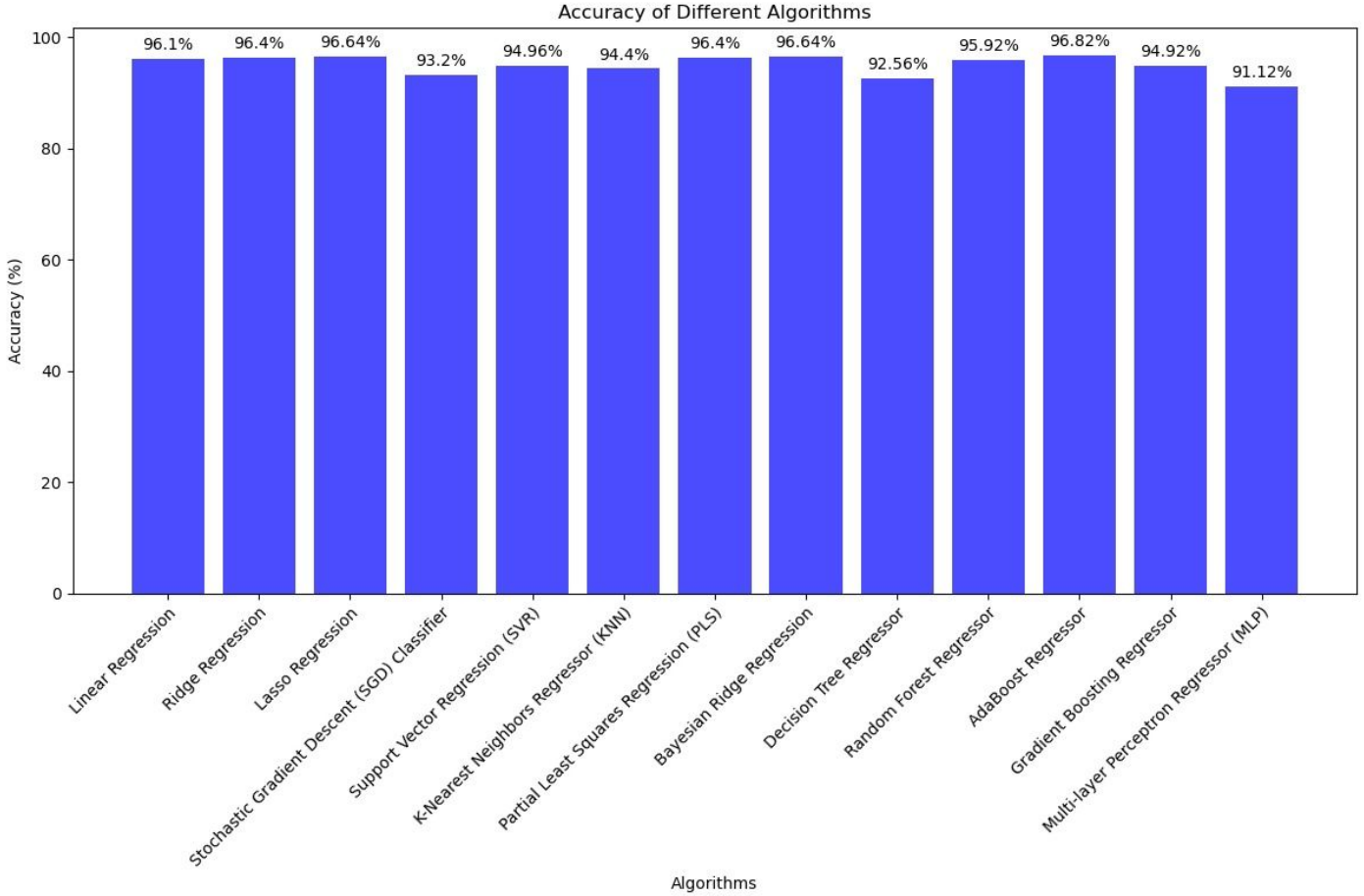
Asymmetry is to account to the fact that an up-triage is more acceptable than a down-triage.

Methodology

We trained on the following ML algorithms.

- Linear Regression
- Ridge Regression
- Lasso Regression
- Stochastic Gradient Descent (SGD) Classifier
- Support Vector Regression (SVR)
- K-Nearest Neighbors Regressor (KNN)
- Partial Least Squares Regression (PLS)
- Bayesian Ridge Regression
- Decision Tree Regressor
- Random Forest Regressor
- AdaBoost Regressor
- Gradient Boosting Regressor
- Multi-layer Perceptron Regressor (MLP)

Results and Analysis



Results and Analysis

Overview of Model Performances:

- The highest accuracy is achieved by the **AdaBoost Regressor (96.82%)**, followed closely by Lasso Regression and Bayesian Ridge Regression (both at 96.64%).

- Linear Regression, Ridge Regression, and Partial Least Squares Regression (PLS) also show high accuracy, all above 96%.

- On the lower end of the spectrum, the **Multi-layer Perceptron Regressor (MLP) has the least accuracy (91.12%)**, followed by the **Decision Tree Regressor (92.56%)** and Stochastic Gradient Descent (SGD) Classifier (93.2%)

Results and Analysis

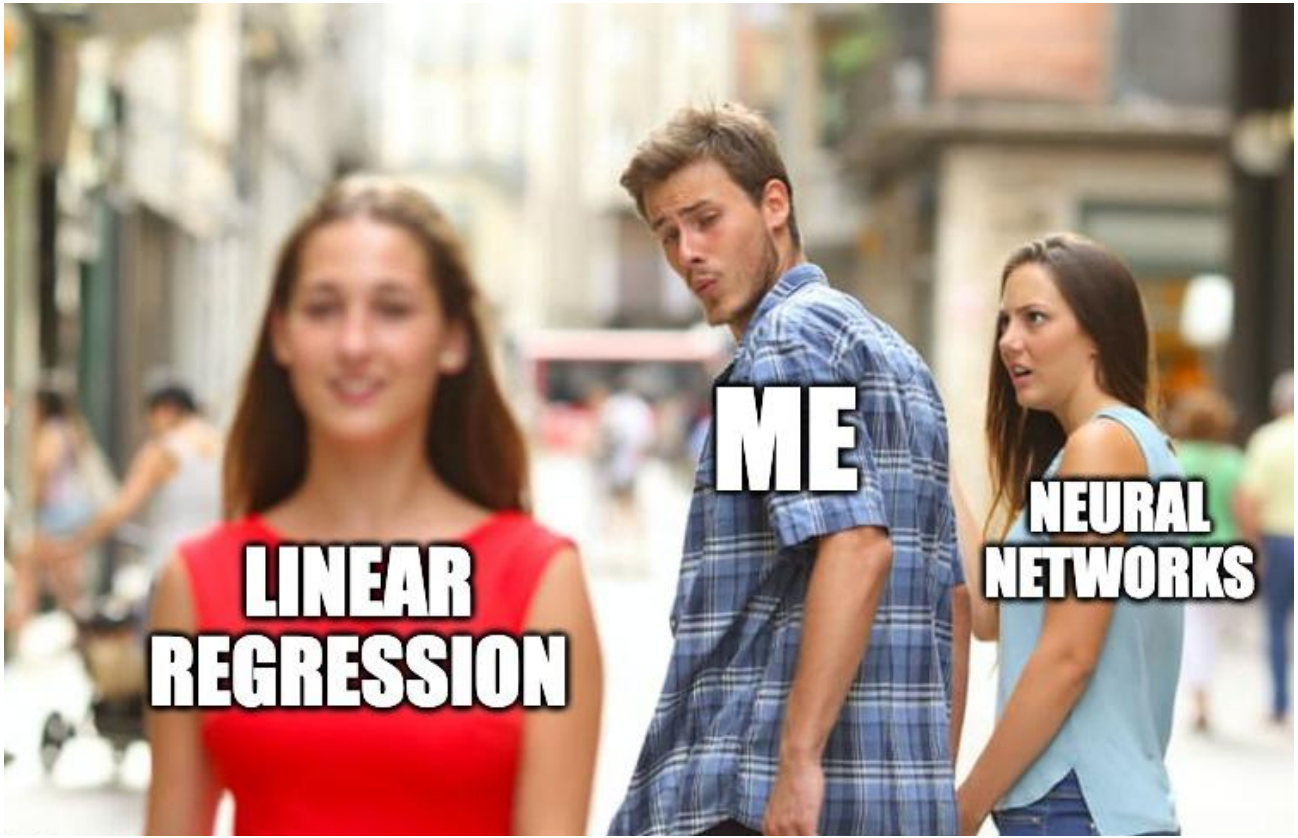
Reasons for Performance Variations:

- **AdaBoost's Success**: AdaBoost's leading performance could be attributed to **its ability to iteratively focus on the instances that previous models misclassified**. This adaptiveness makes it particularly effective for datasets with complex patterns and nuances, like medical data.

- **Effectiveness of Ensemble Methods**: Random Forest high accuracy underscores the advantage of ensemble methods in handling diverse and complex datasets. **By averaging multiple decision trees, it reduces the risk of overfitting and improves the overall prediction accuracy.**

- **Challenges with MLP and Decision Trees**: The lower accuracy of MLP and Decision Tree Regressor might be due to **overfitting**. **MLPs**, being complex neural networks, **can overfit if not properly regularized** or if the data is not sufficient to train such complex models. **Decision Trees are known for their tendency to overfit**, particularly if they are deep and not pruned properly.

Lesson Learned



Thank You!