Predicting The Remaining Useful Lifetime of a Battery

Kristel Rochell Herrera
University of Houston-Victoria
HerreraKR@uhv.edu

Faculty Advisors:
Dr. Lu
University of Houston-Clear Lake
Dr. Gohel
University of Houston-Victoria
Dr. Wan
University of Houston-Victoria

Introduction

Our training data is obtained from Kaggle. The training data contains 7 features and a target column. This data is used to train the machine learning models. We tried five different machine learning models. The test data is also obtained from Kaggle, it has 7 features and does not have a target variable. We try to use this test data to predict the remaining useful battery life.

Objectives

This project's objective is to use the training data provided to train our machine learning model to predict the remaining useful battery life of 19 batteries in the test data set.

Results

We used the mean squared error metric to compare accuracy among the five different machine learning models; which were KNN(K-nearest neighbor), CNN (convolutional neural network), random forest, xgboost, and support vector machine.) The model with lowest mean squared error was KNN which was: 0.02302360388493143

Machine Learning Models

Random Forest- This machine learning model creates decision trees during training to output the average for regression tasks such as the one we are working with.

KNN- The k-nearest neighbor algorithm uses data clustering to predict the average from the surrounding data points in the area.

CNN- Convolutional neural networks is a kind of deep learning algorithm that is mostly used for image recognition Our model contained a neuron since it was a small data set.

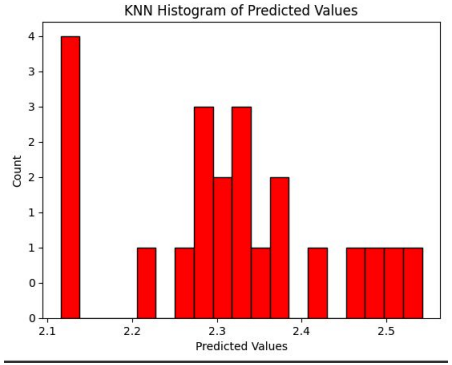
XG Boost- Extreme gradient boosting algorithm utilizes decision trees to make predictions. In this case, the decision trees used the training dataset to predict the target column in the test dataset.

Support Vector Machine- This is an algorithm that falls under the supervised category. It is best used for regression or classifying data.

Mean Squared Error	
Random Forest	0.06626329858789066
XGBoost	0.06604678941316999
Convolutional Neural Network	0.04032666673483938
K-Nearest Neighbor	0.02302360388493143
Support Vector Machine	0.06626329858789066

K-Nearest Neighbor Method

This machine learning model was coded in Google Colab and we used the programming language Python. We used libraries such as pandas, KNeighborsRegressor, and train_test_split for data manipulation, building and training the machine learning model, and for testing and validation of the model. The KNN model uses 7 neighbors to predict the test dataset target. Both training and test datasets are loaded to a panda dataframe for data manipulation. We then separate the training dataset by isolating the features from the target. The training dataset is split, where 80% of the data is used for training and the rest for validation. We train the KNN model on the training dataset. The model then predicts the target. We use mean square error to evaluate the predictions.



Conclusion

KNN provided the lowest mean squared error. The Kaggle private score received was of 181 and the public score of 214. The KNN algorithm trained on the training dataset. The predicted values where in the ones place, which differ from the sample submission which are in the hundreds place.