Predicting Brake Pad Friction Coefficient

Yipeng Liu

Department of Aviation and Technology, San Jose State University

TECH 176, Machine Learning Technology and Applications

Armin Moghadam, Ph.D.

December 10th, 2023

Abstract

Brake pad machine learning models developed in this project can predict the friction coefficient μ with respect to its corresponding compound materials with an average error of 0.127 at best while the sample μ staying in between 0.146 and 1.161.

Table of Contents

I. Introduction

Machine learning (ML) has become a hot topic for the past few years. As compared to hard coded models, ML models are much more versatile and precise when dealing with complex problems with large datasets. For instance, there has been an ever-increasing need of ML models in the material science field as they would offer a great advantage to researchers by saving considerable amount of time from manual experiments. Recently, one challenge proposed by Brembo Inspiration Lab in Santa Clara, California, aimed at finding more brake pad compounds or recipes, as Brembo called them, that has not been synthesized or tested. The solution that Brembo was seeking was a collection of data generated by Artificial Intelligence (AI) with a desirable friction coefficient at 0.6 with an error of 0.1, so the ML specialists at Brembo Inspiration Lab could augment their existing dataset and optimize their current brake pad performances. The original dataset was provided by Brembo Inspiration Lab, and the data was obtained through actual experiments in labs. The models were built on the Google Colab Notebook.

II. Data

The data was provided by Brembo Inspiration Lab during a Hackathon in October, 2023. The data was desensitized, so there was no restriction on using this dataset for this paper, but it was not publicly available.

II.1. Data Types

In *Challenge 1: Gen AI & Brake Pad Recipe Creation* (Brembo, 2023), there were 1,295,428 rows of data consisting of 69 features in total with 67 of them being numerical and 2 of them being categorical. Even though there were only 337 compounds listed in the csv file, the dataset recorded 124 braking events, each measured 31 times (Brembo, 2023). To check the numbers, 337 compounds times 124 events times 31 measurement instances equals to 1,295,428.

Table II.1.1 Data Types (Brembo, 2023)

Column Name	Description	Data Type
time_index	incremental value for each compound characterizing the order of the test. For each compound_id there are 3844 points corresponding to consecutive time steps across all the test brakings	int
compound_id	compound identifier	str
braking_id	braking event identifier, constructed by concatenating the step and stop id	int
step	label of the section within the test. For details see 'Performance Test Specification'	str
time	progressive time [s] per braking_id	float
pressure_bar	pressure [bar] during performance test	float
temp_disc_c	disc temperature [°C]	float
speed_kph	speed [km/h] during performance test	float
mu	friction coefficient	float
A_* B_* C_* D_* E_* F_*	percentage raw material ingredients within the compound. The raw materials belong to one of five material classes encoded with the letters A, B, C, D, E or F. The composition is uniquely associated to the compound_id	float

Figure II.1.1

II.2. Data Visualization

Based on the following collection of graphs, the time_index attribute was approximately a uniform distribution, and mu was approximately a normal distribution. Some recipe materials were used more than the others, such as C_1, E_1, E_5, E_6, and F_9.

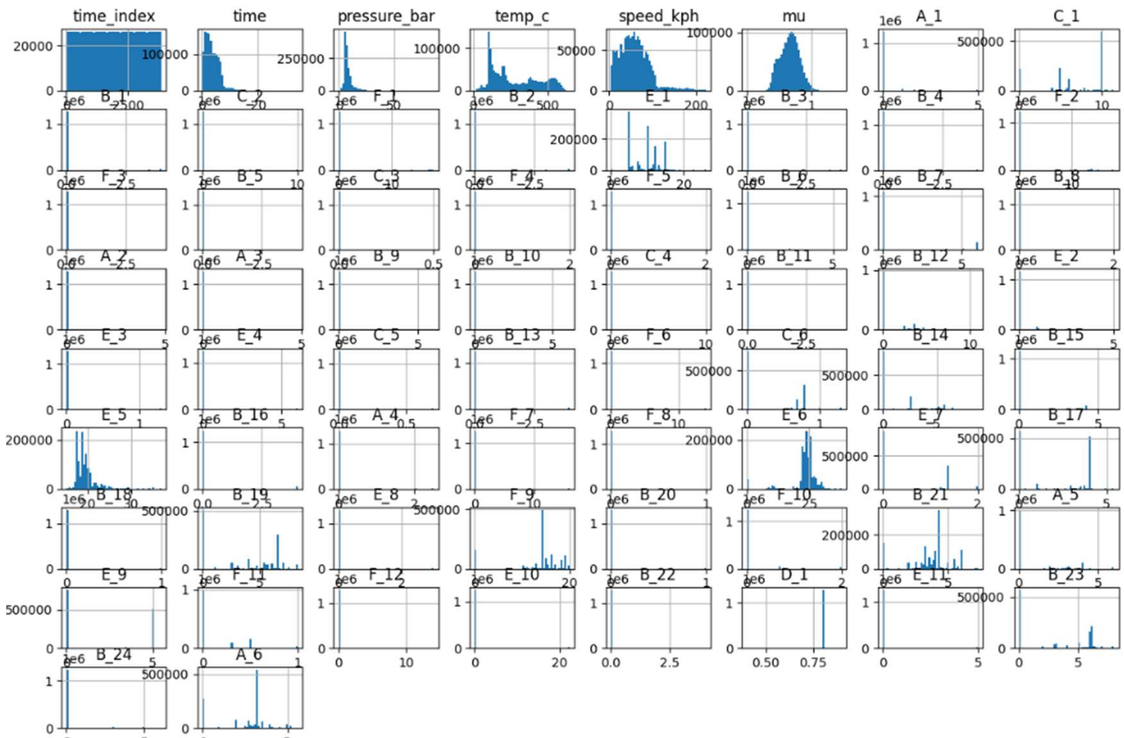

Figure II.2.1 Features Distribution Histogram

The following graph is a scatter matrix to see the correlation between mu and some of the components.

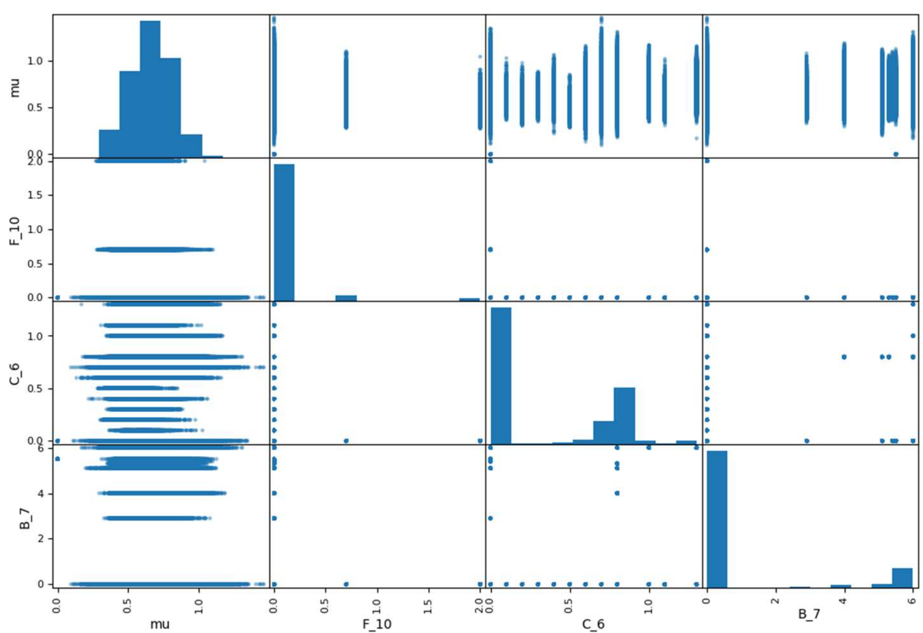

Figure II.2.2 Scatter Matrix

II.3. Statistical Information

Features time_index, compound_id, step, pressure_bar, time, and braking_id were dropped for this project, and the dataset has been down sampled to accommodate the limited computer power available. These 5 features: time_index, compound_id, step, pressure_bar, and time were dropped to isolate the effects of recipes on mu, and braking_id was dropped because of the limitations of one hot encoder. The one hot encoder encodes categorical values and create corresponding columns, adding more weight when building the 2^{nd} degree polynomial model using the LinearRegression class from Scikit Learn.

Furthermore, data entries with mu = 0 were dropped because friction cannot be 0 as far as the current laws of physics hold. For the rest of data used in constructing the models, the min of mu was 0.660, and the max of mu was at 1.161.

Additionally, to alleviate the load for the Google Colab Notebook ram, the dataset was down sampled to 3,562 entries of data using the stratified shuffle split class.

Finally, the dataset was separated into train set and test set with the test size being 10% of the 3,562 rows from the sample data before they were sent into a pipeline that scaled the datasets.

III. Methods and Materials

III.1 Problem Definition

Due to the time and resource constraints and the scope of TECH 176 class, the problem has been simplified as such: building a ML model based on the knowledge obtained in TECH 176 to predict the value of the friction coefficient. This was a regression problem because the output sought after was predicted values of the friction coefficient, and no classification was involved.

III.2. Machine Learning Approaches

To solve this regression problem, the following models were selected: linear regression, 2^{nd} and 3^{rd} degree polynomial regressions using polynomial features and the linear regression class, elastic net for overfitting, linear and polynomial regressions using the support vector machine, decision tree, and random forest.

III.3. Logics Behind Model Selections

All of the models listed above were covered in class, and due to the scope of this course, no other models were considered during model selection phase. The linear regression models served as bench marks to see if the hyperparameters needed to be optimized or not for the more advanced models such as decision tree and random forest.

IV. Evaluations and Discussions

The performance was evaluated based on the test root mean squared error (RMSE), and as stated by Brembo during the hackathon, the target mu error should be no more than 0.1. In other words, the metrics for test RMSE was 0.1. The following table was constructed based on the model performances.

Table IV.1 Performance Evaluation Table

Models	Train RMSE	Test RMSE	ElasticNet Train RMSE	ElasticNet Test RMSE	Prediction (actual label is 0.826)
LinearRegression	0.133	0.133	N/A	N/A	0.712
2nd Degree Polynomial	0.114	0.131	N/A	N/A	0.849
3rd Degree Polynomial	0.098	0.177	0.144	0.143	0.662
LinearSVR	0.142	0.141	N/A	N/A	0.702
2nd Degree SVR	0.120	0.133	N/A	N/A	0.758
3rd Degree SVR	0.120	0.133	N/A	N/A	0.737
Decision Tree	0.133	0.127	N/A	N/A	0.853
Random Forest	0.124	0.127	N/A	N/A	0.776

IV.1. Interpreting Machine Learning Models' Outputs

The random forest model and decision tree model have excelled in performance metrics based on their low test RMSEs. Moreover, they have also surpassed the linear regression models, both from the LinearRegression class and support vector machine class. Sadly, even the best performing models in this project could not suffice the Brembo requirement of error less than 0.1.

The last column in the table above was computed based on the first entry in the test set. It was used to show that even though some models have higher test RMSE, they could actually perform better in some instances, or in other words, "best case scenario." But such scenarios were not reliable, and the lowest Test RMSE was still more desirable because best case scenarios could not be guaranteed.

V. Conclusions and Future Recommendations

The random forest model and decision tree model have proven to be the best model for predicting mu based on the given inputs. However, the results could be further improved by having much higher ram sizes to handle the heavy computations caused by 1 million rows of data. Also, a better approach could be using time series in deep learning recurrent neural network (RNN) to better utilize the braking events from the original dataset that was dropped in this project.

Since ML models are available at hand, more recipes will be generated by future researchers using these models as guidelines to build more compounds. And through harnessing such power from ML, greater results can be achieved with less and less efforts and time.

References

Brembo, *Challenge 1: Gen AI & Brake Pad Recipe Creation*. Unpublished confidential document; 2023.