Udacity Machine Learning Engineer Nanodegree
Capstone Project Proposal

William Koehrsen

June 20, 2017

Domain Background

Safety and reliability testing is a crucial step in the automobile design process. Every novel vehicle concept must pass a thorough evaluation before it enters the consumer market. Testing can be time-consuming as a full check of vehicle systems requires placing the car in all situations it will encounter in its intended use. Predicting the overall time for a vehicle to pass testing is difficult because each model requires a different test stand configuration.[1] Mercedes-Bens has been a pioneer of numerous safety and technological vehicle features and offers a wide range of custom options to customers. Each and every possible combination must undergo the same rigorous testing to ensure the vehicle is robust enough to keep occupants safe and withstand the rigors of daily use. The large array of options means a large number of tests for Mercedes' engineers to conduct. More tests result in more time spent on the test stand which drives up costs for Mercedes and generates carbon dioxide, a polluting greenhouse gas.[1] Efforts by Mercedes Benz and other automakers to improve the efficiency of the testing procedure have mainly focusing on developing automated test systems.[2,3] An automatic test system eliminates the variability inherent in human behavior, especially between different drivers, and is therefore able to reduce the number of tests required. Moreover, testing can be dangerous for human drivers, and putting software in charge of vehicle testing makes for an overall more efficient validation process. The Mercedes-Benz "Greener Manufacturing" Competition on Kaggle[1] pursues a related approach by encouraging the development of a model that is able to predict the testing time for a particular vehicle combination.

The story of Mercedes-Benz indicates the technological foresight of the company. Mercedes-Benz is a German luxury automobile manufacturer that can trace its origins back to the first gasoline-powered automobile, Karl Benz's 1886 Benz Patent Motorwagen. The company was formed when Daimler Motors Corporation (DMG), an engine and automobile manufacturer, and Benz & Company merged in 1926 to form Daimler-Benz.[4] Today, Mercedes-Benz is a division of Daimler AG and sold 380,461 cars in 2015. [2] The Greener Manufacturing competition was posted on Kaggle in May 2017 with a July 10 deadline for submitting a model. Over 2500 teams are competing to help Mercedes improve the vehicle testing process (and the prize may also be a slight motivator). Ideally, reducing the total testing times will reduce vehicle carbon dioxide emissions.[1] Personally, any project that seeks to develop more environmentally-conscious systems is appealing to me. Even though the reduction in carbon dioxide may not be noteworthy on a global scale,[6] advances made in the Mercedes-Benz testing process could be passed along to other automakers or even to other industries which could result in a significant impact. Thousands of small efficiency gains do add up over time to create a more sustainable

society. Moreover, I believe that one of the fundamental goals of machine learning is to develop more efficient systems using the vast amounts of data now routinely collected. Gathering data by itself is meaningless unless that information is acted upon. As an engineer, I abhor inefficiency in any form, and I am excited to work to improve Mercedes-Benz's testing procedure. Finally, I have always enjoyed competitions, and I believe that competitions like Kaggle and the X-Prizes[7] hold great potential for spurring new developments and encouraging unique solutions to problems.[8]

Problem Statement

The copious number of vehicle customization options offered by Mercedes-Benz results in a large quantity of required vehicle tests because each configuration must be validated for safety and reliability. The more time spent by vehicles on the test bench, the higher the costs for Mercedes and the greater the quantity of carbon dioxide emitted. Mercedes Benz has therefore created the "Greener Testing" competition on Kaggle with the purpose of developing an algorithmic model that can predict the testing time for a vehicle given the vehicle configuration. [1] The idea is that being able to predict the duration of a test will allow Mercedes to reduce the total time vehicles spend on the test bench[9] by running cars with similar testing configurations successively. This situation is a regression machine learning problem because it requires predicting a continuous target variable (the duration of the test) based on a set of explanatory input variables (the customization options of the vehicle). Mercedes-Benz will implement the best-performing model into its design process to increase the overall efficiency of the testing procedure while maintaining high safety and reliability standards.

Datasets and Inputs

All of the testing and training for this project is provided by Mercedes-Benz through the Kaggle Competition Page.[10] The data is available for download as three comma separated values (CSV) files: a training dataset, a testing dataset, and an example submission file. In the training and testing files, the first row is the column headers and the remaining rows are the vehicle tests, with each test occupying one row. The columns contain the variables, or features, that define each test. The training dataset consists of 4209 tests. Each test has an identification number (ID) and the significance or non-significance of the ID is not provided by Mercedes. The testing dataset contains labels, the y column, for every test which represents the duration of the test in seconds. The remaining columns contain the descriptive characteristics of each testing configuration. The raw data contains a total of 376 variables. 7 of these are categorical variables with string values such as 'a', 'b', 'c', etc. 369 of the features are binary with values of either 0 or 1. Both the categorical and binary variables are anonymized meaning that Mercedes-Benz does not provide any physical descriptions of the variable. However, the description of the data[10] does indicate that each variable represents a custom feature in a Mercedes vehicle. These could be options such as suspension setting, engine type, adaptive cruise control, or a number of other possible additions. Each test therefore is composed of an

ID, a time in seconds for the test (the target), and a combination of variables (the features) that represent the vehicle arrangement.

The testing categorical variables contain a total of 195 unique categories. If all of the features are one-hot encoded, there will thus be a total of 564 features in the training data file. The testing data file also has each test represented by a row and the columns recording the vehicle features. However, the testing data does not include labels because the objective of the model is to predict those testing durations. There are 4209 tests in the testing set. Each testing vehicle test also has an ID with no stated physical representation. The same 375 raw features are in the testing set with 7 categorical and 369 binary features. The 7 categorical features have 201 unique categories in the testing set. This results in a total of 570 features in the testing set if the categorical variables are one-hot encoded. The sample submission file contains an example of the correct format of a submission to the competition. The correct format is a CSV file with the ID of the vehicle test from the testing set in the first column and the predicted testing time in seconds in the second column.

The entire dataset was obtained through download as a zip file from the Kaggle Mercedes-Benz competition page. The dataset has already been cleaned and is ready for use. The application of the data is appropriate for this project because the competition is designed specifically for this dataset. Mercedes-Benz has remained tight-lipped about the dataset[9] and will not provide any additional information about the significance of the features beyond saying that they represent different car configurations. An additional source of information that may be useful is the discussion page for the Greener Testing competition.[11] Competitors have posted example projects and engage in well-informed debates about the significance of variables and the best approach for the problem.

Solution Statement

The solution to the Mercedes-Benz Greener Manufacturing Competition is a model that predicts the testing time of a vehicle based on its configuration. The model should take in the features of the vehicle, and return a predicted testing time in seconds for the vehicle. A better model is therefore one that more accurately represents the testing time of a vehicle. This model will allow Mercedes' engineers to better prepare for each test and reduce the overall process time on the test bench through methods such as grouping cars with similar testing times together. This will minimize the necessary configuration changes to the test stand between vehicles. The quality of the solution will be judged based on the R^2 measure,[1] called the coefficient of determination. The coefficient of determination is a measure of how much of the variance in the dependent variable (in this case the testing time in seconds) is explained by the independent variables (in this case the configuration of the vehicle) and a better model will have a higher coefficient of determination. A useful model will not only perform well on the testing set, but can be implemented by Mercedes-Benz in the automotive design process.

Benchmark Model

A benchmark model for this regression task would be a linear regression. Linear regression is a method for modeling the relationship between a target variable (also called the response variable) and one or more independent, explanatory variables.[12] When there are multiple explanatory variables, this process is called multiple linear regression. In this situation, the target variable is the testing time of the car and the explanatory variables are the customization options for the car. A linear regression model assumes that the response variable can be represented as a linear combination of the model parameters (coefficients) and the explanatory variables. The explanatory variables can be transformed by being raised to higher powers, which is a form of linear regression known as polynomial regression. However, the benchmark model in this situation will be a basic linear regression and will predict the target for an instance based on the instance's features multiplied by the model's parameters (coefficients) and added to a bias term. Linear Regression can be implemented in Scikit-Learn using the Linear Regression class found in the Linear Model module.[13] The default method for Linear Regression is Ordinary Least Squares where the model minimizes the sum of the squares between the prediction and the known targets.[12]

The benchmark model can be compared to the final model by comparing the R^2 score or coefficient of determination of the Linear Regression to the final model. The Linear Regression class in Scikit-Learn includes a score method that returns the coefficient of determination between the features and the target variable. The benchmark model will use all of the raw features (after one-hot encoding of the categorical variables) in order to determine the effect of feature reduction/ feature engineering in the final model. The final model will need to perform significantly better in terms of the coefficient of determination than the benchmark model for the project to be a success.

Evaluation Metrics

The evaluation metric for the competition is the R^2 measure, known as the coefficient of determination. R^2 is used as a measure of the quality of a model that is used to predict one quantity from a number of other quantities.[14] It describes the amount of variation in the dependent variable, in this case the testing time of a vehicle in seconds, based on the independent variables, in this case the combination of vehicle custom features, that can be explained by the model. It is often interpreted as the percentage of the variation in the targets that is explained by the features, meaning that an R^2 value of 0.6 indicates that 60% of the variation in the testing time could be explained by the variation in the vehicle set-up. The remaining 40% of the variance is either not captured by the model, or is due to lurking variables that have not been included in the data.

The coefficient of determination is the appropriate metric for the problem because the goal as put forth by Mercedes-Benz is to create a model that is able to capture the testing time

of a vehicle. Mercedes is interested in why different vehicles take different times to test and how this can be represented in a machine learning model. Therefore, the algorithm that best explains the variation in testing times will be the optimal machine learning model for the task. Moreover, the evaluation metric used to determine the winner of the competition is the coefficient of determination.[1] The coefficient of determination is a common metric used in regression tasks and is implemented in Scikit-Learn, where it is the default evaluation score for a regressor. [15]

Project Design

A typical machine learning process consists of a number of well-defined steps: 1. Data Gathering; 2. Data Exploration and Cleaning; 3. Feature Manipulation and Engineering; 4. Dimensionality Reduction; 5. Model Selection; 6. Hyper parameter Optimization; 7. Model Evaluation and Testing; 8. Model Presentation.[16]

Step 1 is simple because the data is already provided as a series of downloadable CSV files on the Kaggle competition page.

Step 2 is also relatively easy in this situation because the data has already been cleaned and is in a usable format. However, I will be exploring the data to understand the different features and to search for outliers. As the features are anonymized by Mercedes-Benz, it may be difficult to contextualize the features, but I will still be able to at least see the distribution of the binary variables and examine the testing times. If there are outliers, defined as 1.5 times the interquartile range (IQR) below the first quartile or 1.5 times the IQR above the third quartile, in the testing times, these instances may need to be removed. Outliers can negatively affect the performance of the model even if the data is valid.

Step 3 will begin with one-hot encoding of the categorical variables. This can be accomplished in Scikit-learn using the Label Binarizer class or in Pandas with pandas.get_dummies. This will expand the number of features, but is necessary in order for the machine learning model to be able to process these variables. After the features have been transformed into a usable format, I will need to determine which features are most relevant to the task. I do not think I will be able to engineer any new features because the existing features do not have any physical meaning and cannot be intuitively combined.

Step 4 will require reducing the number of features in the dataset. Inevitably, there will be some features that are not useful in the model and can be removed. The explanatory power of each variable can be checked using the feature_importances_ method found in several Scikit-learn classes, including the DecisionTreeRegressor. From the feature importances, I can retain only the features that are most relevant for predicting the vehicle testing duration. Another dimensionality reduction technique is Principal Components Analysis (PCA) which finds the basis vectors with the greatest variance in the data.[17] The first principal component explains the greatest variance in the data, while the second explains the second most and is orthogonal to

the first and so on. I can then keep the number of principal components that explain a certain total percentage of the variability within the data. Independent Components Analysis (ICA) also finds a set of basis vectors, but these vectors represent the dimensions with the greatest independence from one another.[18] In other words, PCA is used to compress data while ICA is used to separate data into unique subsets. Scikit-learn provides implementations of both of these techniques in the decomposition module.[19] Feature reduction is necessary in this problem because of the high dimensionality of the data. As the amount of dimensions, or features, of the data increases, the amount of samples required to learn the underlying relationships within the data increases exponentially, in what is known as the "curse of dimensionality."

Step 5 involves testing several different algorithms to determine which one performs the best for the data. The model with the greatest evaluation metric will be selected for optimization. I have not decided on a model to use yet, but I will evaluate several candidate regressors including the Decision Tree Regressor, K-Nearest Neighbors Regressor, Stochastic Gradient Descent (SGD) Regressor, and ensemble methods such as Random Forest or Gradient Boosting Regressor. Based on the competition forums, I think that the best performing model will be an ensemble method. There are two classes of ensemble methods: averaging methods and boosting methods.[20] Average models train numerous simple models and essentially take an average of their predictions to determine a final answer. Boosting methods start out with a single simple estimator and then build on top of the estimator, with each iteration adding to the model by trying to reduce the bias of the combined model.[21] Ensemble methods are generally more accurate than a simple model and reduce the variance that can lead to overfitting with a single simple model.

Step 6 will involve tuning the hyperparameters of the selected model from step 5 to maximize performance on the evaluation metric. I like to think of this step as adjusting the settings of the algorithm to their optimal state. This can be done manually, by changing one or multiple hyperparameters at a time and then evaluating the model, or it can be done programmatically using a Grid Search. Scikit-learn provides the implementation of GridSearch with GridSearchCV. The idea is that a set of hyperparameter configurations are defined in a parameter grid and then each combination of settings is tested and scored using the evaluation metric. In this case the evaluation metric will be the coefficient of determination, and Grid Search will return the parameters from the grid that result in the highest R^2 value. The CV in GridSearchCV stands for cross-validation which is a technique that is used to prevent overfitting on the training data. Rather than testing each model against the same training data and then evaluating against one testing set, the training data is split into n number of subsets, called folds. The model is then trained on n-1 of these subsets and tested against the nth subset. This process is repeated iteratively n times. The final score of the model is the average score across the n validation tests. Then, after cross validation has been completed, the model is tested one final time on the testing set. Scikit-learn also provides another method for hyperparameter optimization called

RandomizedSearchCV. The principal is the same as GridSearch except that the user does not define the exact settings of the hyperparameters to evaluate. Rather, the user specifies the number of combinations to test and the hyperparameters to adjust, and the algorithm will test random values for the different hyperparameters and again return the configuration that scores the highest on cross validation. This is a good choice for evaluating a range of hyperparameters especially when the user is not sure what the scale of the hyperparameters should be. I may need to start off using Randomized Search to gauge the correct scale for the hyperparameters and then Grid Search to narrow down the optimal hyperparameters.

Step 7 will require running the final optimized model against the testing set. The testing set has also been provided by Mercedes-Benz, although without labels. This makes scoring the regressor difficult. However, once the testing set predictions have been submitted to the competition, Mercedes calculates the R^2 value based on the known correct times that they have recorded. I will then be able to observe my place on the leaderboard. The number 1 competitor on the public leaderboard currently has recorded a coefficient of determination of 0.57542 on the testing set.[22] The public leaderboard is based on approximately 19% of the test data while the final results will be based on the other 81% of the testing data. I will therefore be able to compare my models progressively against one another, as well as against a larger number of competitors. My goal is to make it into the top 5% of competitors on the public leaderboard.

Step 8 should be the exciting part! In an industry setting, at this point I would present my solution to the decision-makers at the company and they would choose whether the model should be implemented. A real-world implementation involves feeding the model new data where the answers are not known. Once a model has been released on real-world data, the data scientist's job is far from done. The model will require constant monitoring to ensure that its accuracy (or chosen evaluation metric) remains at an acceptable standard. Moreover, the model will require re-training as further data becomes available. Eventually, the model may need to be completely overhauled if a new technology or method comes along that offers increased performance. However, unless I win the competition, my model likely will not see use in the real world.

In this step, I will also need to document and record my process so others can follow along and understand the steps I took. The model needs to be presented in an understandable format to an audience that may not be technologically inclined. Machine learning may appear to be miraculous, but that belies the fact that it is in actually quite understandable in principle. Clearly communicating my ideas in writing not only gives me a chance to make sure that I comprehend what my model is doing, but it also allows the chance for others to constructively criticize my work. This feedback process will provide me the opportunity to improve my project and ultimately make me a better machine learning engineer.

References

[1]"Mercedes-Benz Greener Manufacturing Overview", Kaggle.com, 2017. [Online]. Available: https://www.kaggle.com/c/mercedes-benz-greener-manufacturing. [Accessed: 20- Jun- 2017].

[2]H. Schoner, S. Neads and N. Schretter, "Testing and Verification of Active Safety Systems with Coordinated Automated Driving", 2017.

[3]M. Tatar and R. Schaich, "Automated Test of the AMG Speedshift DCT Control Software", 2010.

[4]"Mercedes-Benz Tradition", daimler.com, 2017. [Online]. Available: https://www.daimler.com/company/tradition/mercedes-benz/. [Accessed: 20- Jun- 2017].

[5]M. USA, "Mercedes-Benz USA Reports Highest Year Ever With 2015 Sales Of 380,461", Prnewswire.com, 2016. [Online]. Available: http://www.prnewswire.com/news-releases/mercedes-benz-usa-reports-highest-year-ever-with-2015-sales-of-380461-300199502.html. [Accessed: 18- Jun- 2017].

[6]"Global Greenhouse Gas Emissions Data | US EPA", US EPA, 2017. [Online]. Available: https://www.epa.gov/ghgemissions/global-greenhouse-gas-emissions-data. [Accessed: 20- Jun- 2017].

[7]"About The Prize", XPRIZE, 2017. [Online]. Available: http://www.xprize.org/about. [Accessed: 20- Jun- 2017].

[8]L. Kay, "The effect of inducement prizes on innovation: evidence from the Ansari XPrize and the Northrop Grumman Lunar Lander Challenge", R&D Management, vol. 41, no. 4, pp. 360-377, 2011.

[9]"Mercedes-Benz Greener Manufacturing Discussion: Hello from Mercedes", Kaggle.com, 2017. [Online]. Available: https://www.kaggle.com/c/mercedes-benz-greener-manufacturing/discussion/34029. [Accessed: 20- Jun- 2017].

[10]"Mercedes-Benz Greener Manufacturing Data", Kaggle.com, 2017. [Online]. Available: https://www.kaggle.com/c/mercedes-benz-greener-manufacturing/data. [Accessed: 19- Jun- 2017].

[11]"Mercedes-Benz Greener Manufacturing Discussion", Kaggle.com, 2017. [Online]. Available: https://www.kaggle.com/c/mercedes-benz-greener-manufacturing/discussion. [Accessed: 20- Jun- 2017].

[12]D. Freedman and D. Freedman, Statistical Models: Theory and Practice, 2nd ed. Cambridge: Cambridge University Press, 2009.

[13]"sklearn.linear_model.LinearRegression — scikit-learn 0.18.1 documentation", Scikit-learn.org, 2017. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html#sklearn.linear_model.LinearRegression. [Accessed: 20- Jun- 2017].

[14]"Coefficient of Determination: Definition", Stattrek.com, 2017. [Online]. Available: http://stattrek.com/statistics/dictionary.aspx?definition=coefficient_of_determination. [Accessed: 20- Jun- 2017].

[15]"sklearn.metrics.r2_score — scikit-learn 0.18.1 documentation", Scikit-learn.org, 2017. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html. [Accessed: 20- Jun- 2017].

[16]A. Géron, Hands-On machine learning with Scikit-learn and TensorFlow. O'Reilly, 2017.

[17]L. Smith, A Tutorial on Principal Components Analysis. 2002. [Online]. Available: http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

[18]A. Ng, CS 229 Lecture Notes: Independent Components Analysis. 2017. [Online]. Available: http://cs229.stanford.edu/notes/cs229-notes11.pdf

[19]"2.5. Decomposing signals in components (matrix factorization problems) — scikit-learn 0.18.1 documentation", Scikit-learn.org, 2017. [Online]. Available: http://scikit-learn.org/stable/modules/decomposition.html. [Accessed: 20- Jun- 2017].

[20]"1.11. Ensemble methods — scikit-learn 0.18.1 documentation", Scikit-learn.org, 2017. [Online]. Available: http://scikit-learn.org/stable/modules/ensemble.html. [Accessed: 20- Jun- 2017].

[21]D. Opitz and R. Maclin, "Popular Ensemble Methods: An Empirical Study", Journal of Artificial Intelligence Research, 1999.

[22]"Mercedes-Benz Greener Manufacturing Leaderboard", Kaggle.com, 2017. [Online]. Available: https://www.kaggle.com/c/mercedes-benz-greener-manufacturing/leaderboard. [Accessed: 20- Jun- 2017].