DATA SCIENCE – I

Title: Mercedez-Benz Greener Manufacturing

Arvind Vaitheeswaran (670795302)

Ruchi Vazirani (652326908)

Sanket Waghmare (664947950)

Raunak Patil (673045203)

INDEX

PROBLEM DEFINITION

As one of the world's biggest manufacturers of premium cars, safety and efficiency are paramount on Daimler's production lines. Mercedes-Benz must ensure the safety and reliability of each unique car configuration before they hit the road. Quality testing is an integral part of any manufacturing facility. Daimler's objective is to achieve supreme quality in order to best prepare their vehicles for rigorous urban and rural operation. There are various quality checks that Daimler's vehicles need to pass through production, before they are declared as 'Conforming or 'Non-Conforming'. The quality checks include, Hardware in loop (HIL), component strength analysis, evaluation of vehicle vibrations, crash test simulations, etc. and many more. The speed of the testing system for so many possible feature combinations is complex and very time consuming. In order to adhere to standard production rules, safety and efficiency for Daimler's production lines, optimization of the testing procedure is required. Daimler has proposed Data Science/Big Data enthusiasts to optimizing an algorithm-based solution/model by encouraging the development of machine learning that is able to predict the testing to the above-mentioned problem.

STATEMENT OF OBJECTIVES

The primary challenge is to tackle the curse of dimensionality. The data set encompasses around 373 attributes and 8400 instances. The objective is to accurately predict the time a car will spend on the test bench based on vehicle configuration. The intention is that an accurate model will be able to reduce the total time spent testing vehicles by allowing cars with similar testing configurations to be run successfully. This is an example of a machine learning regression task since, it requires predicting a continuous target variable (the duration of test) based on the model of the car [X1-X9]. This is also an example of supervised task,

since the labels are available to us in the form of 'Y' which is in seconds. Model solutions will provide insight to what extent and how the car configuration affects the process time.

Moreover, one of the main aspects of machine learning is that the efficiency of current systems can be improved using massive quantities of data, which is routinely collected by companies. Reducing the process time via reducing the "standard deviation", carbon dioxide emissions can be lowered. Resulting in speedier testing, resulting in lower carbon dioxide emissions without reducing Daimler's standards.

SOURCE OF DATASET

There are 2 files available to use. 'train.csv' and 'test.csv' both containing 4209 rows of data. These rows essentially indicate the number of vehicles provided to us and their various configurations. The 'train.csv' also contains the target variable which is available as the first column in 'Y' which are numeric values in seconds. Both the test and train contain 8 different vehicle features with names such as X0 – X8 and 369 tests performed for the corresponding vehicle models. All the features have been anonymized and they do not come with any physical representation. The description of data does not indicate that vehicle features are configuration options. There are 8 categorical features with values encoded as strings such as 'A','B', 'C', etc. There are 368 tests which are integer values and only indicate 0 or 1 i.e it indicates whether the tests are performed or not. Upon, a quick inspection, there are no missing values or data in irregular format. It has been claimed by Mercedez – Benz that, the time measurement error is considered to be almost zero for time spent by each tests on the testing bench.

EVALUATION METRICS

The primary evaluation metric for this dataset is the co-efficient of determination (R^2 measure). R^2 is the measure of quality of a model that is used to predict one continuous variable from a number of other variables. It describes the amount of variation in the dependent variable, in this case the testing time of vehicles in seconds, based on independent variables which, in this case is the combination of vehicle custom feature.

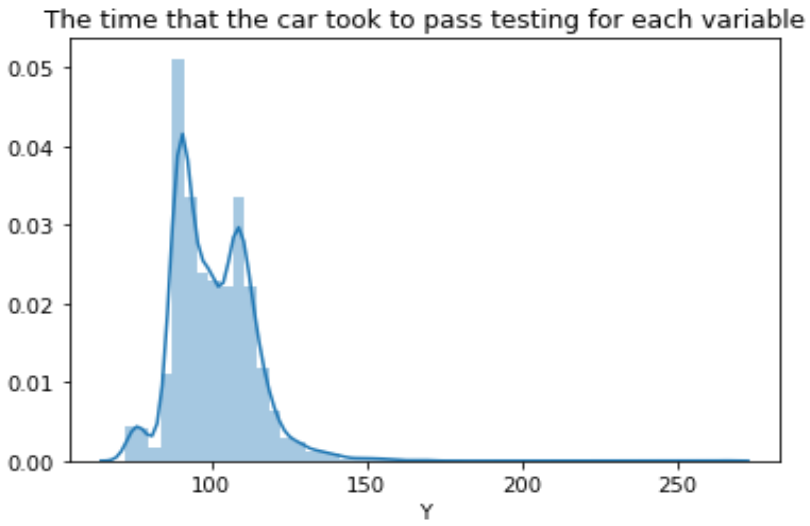
The co-efficient of determination is expressed as:

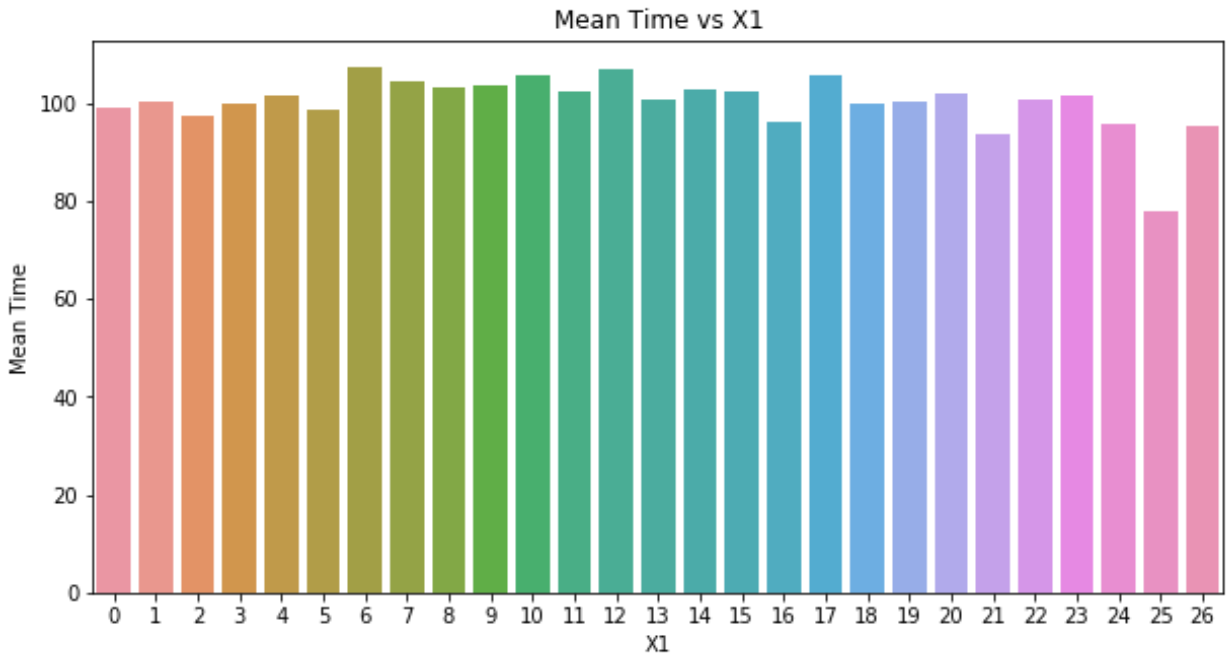
$$R^2 = \left(\frac{n * (\sum x * y) - (\sum x)(\sum y)}{\sqrt{n * [(\sum x^2) - (\sum x)^2] * [n * (\sum y^2) - (\sum y)^2]}} \right)^2$$

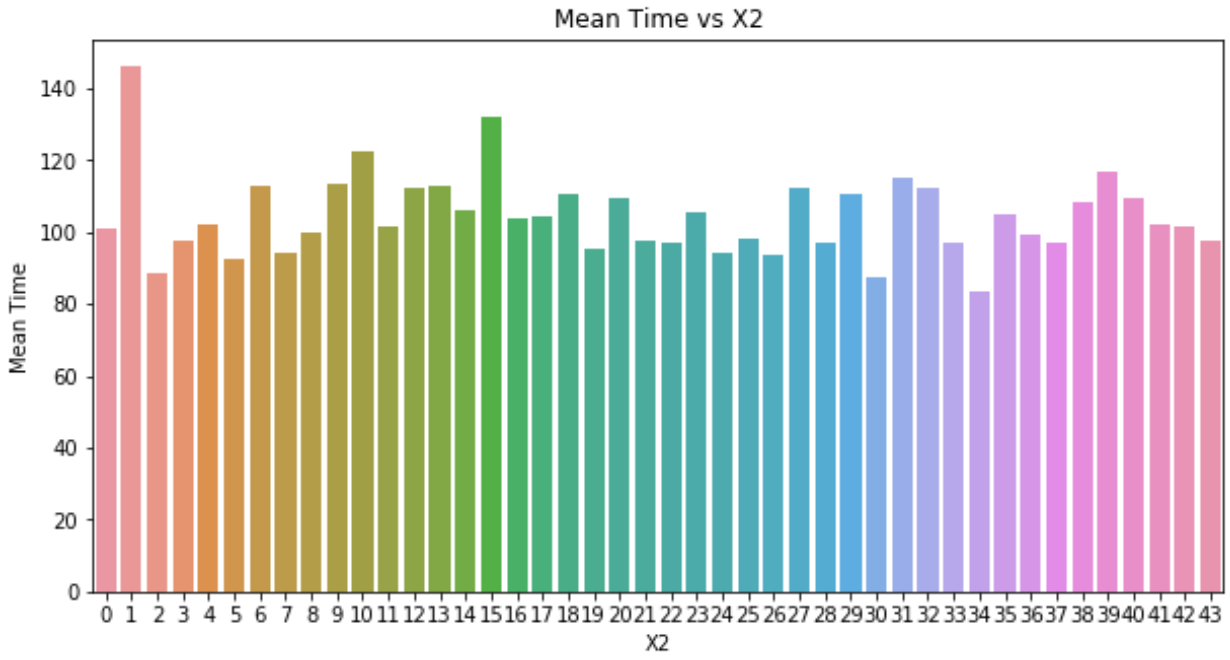
Equation 1: Coefficient of Determination

Where ‘n’, is the number of instances (which tests), ‘x’ is the prediction for the instance(the predicted time in seconds) and ‘y’ is the actual time (given in the dataset, in seconds).

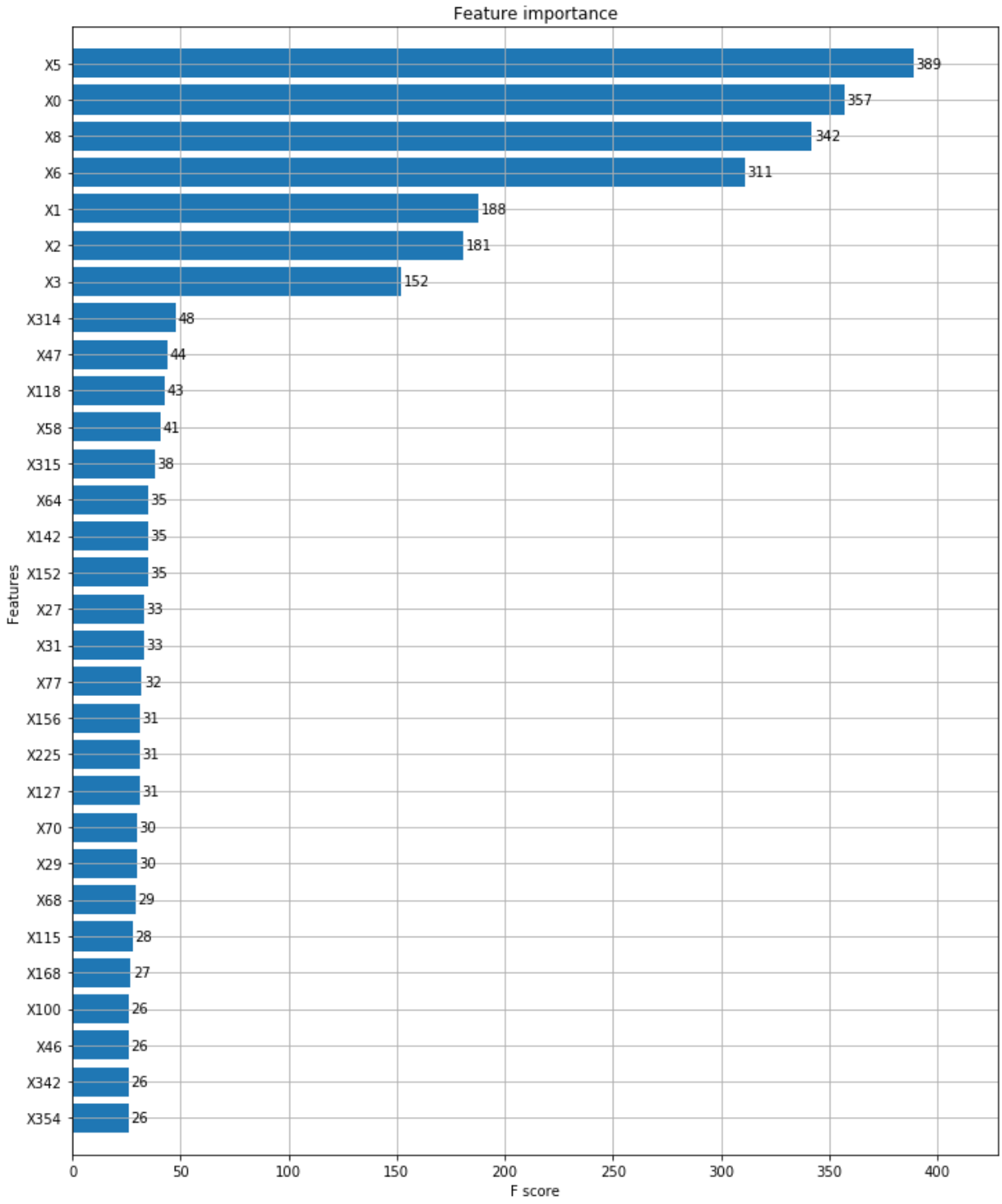
DATA VISUALIZATION







We plotted various different attributes for the mean time they contribute to and some of them had relatively high contribution to the total time and some of them had very low contribution. This led us to the analysis of PCA and ICA where attributes who contributed less would eventually be dropped or not considered for training.



The above graph shows us the important features selected by XGBoost. The features selected are categorical featuresas well as binary features. In our case, we will give more importance to the categorical features and hence, we go ahead and plot the mean times for all the categorical features. The categorical features are X5, X0, X8, X6, X1, X3, X2.

FEATURE SELECTION TECHNIQUES

As mentioned earlier, we have 8 categorical columns and 369 integer values. These 369 integer values are already in '0' and '1' format, so we will convert other 8 categorical columns into 'One Hot Encoded' columns to have entire train data in same format. The 'One Hot Encoding' is also applied to the 'Test' data. After having data in common format, following are various types of feature selection techniques used on the dataset:

1. **Principal Component Analysis (PCA):**

   Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components. It emphasizes variation and brings out strong patterns in a dataset. PCA is implemented as a transformer object that learns n-components in its 'fit' method, and can be used on new data to project it on these components.

2. **Singular Value Decomposition (SVD):**

   This transformer performs linear dimensionality reduction, contrary to PCA. This estimator does not center the data before computing the singular value decomposition. SVD makes it easy to eliminate the less important parts of the representation to produce the approx representation with any desired number of dimensions.

3. **Independent Component Analysis (ICA):**

Independent component analysis (ICA) is a statistical and computational technique for revealing hidden factors that underlie sets of random variables, measurements, or signals. ICA defines a generative model for the observed multivariate data, which is typically given as a large database of samples. In the model, the data variables are assumed to be linear or nonlinear mixtures of some unknown latent variables, and the mixing system is also unknown. ICA can be seen as an extension to principal component analysis and factor analysis. ICA is a much more powerful technique, however, capable of finding the underlying factors or sources when these classic methods fail completely.

METHODS

This project is a supervised Machine learning problem where the algorithm receives a set of inputs and the corresponding outputs, so the algorithm compares its outputs with the correct ones then finds errors. Accordingly, we would like to explore the following analytics methods of regression analysis.

REGRESSION

Regression is a method of modeling a target value based on independent predictors. This method is mostly used for forecasting and finding out cause and effect relationship between variables. Regression techniques mostly differ based on the number of independent variables and the type of relationship between the independent and dependent variables.

1. Linear Regression

2. Ridge Regression
3. Lasso Regression
4. Decision Tree Regression
5. KNN Regressor
6. Artificial Neural Network (ANN)

1. Linear Regression

Linear regression is a type of regression analysis where the number of independent variables is one and there is a linear relationship between the independent(x) and dependent(y) variable. This is usually a naïve method and used as a basis for reference to other models. Based on the given data points, we try to plot a line that models these points with linear best fit. The line can be modeled based on the linear equation, for Eg.: $y = a_0 + a_1 * x$, here the motive of the linear regression algorithm is to find the best values for a_0 and a_1.

2. Ridge Regression

Ridge regression works by penalizing the magnitude of coefficients of features along with minimizing the error between predicted and actual observations. It is called 'regularization'. Ridge regression performs '**L2 regularization**', i.e. it adds a factor of sum of squares of coefficients in the optimization objective. The main features of ridge regression are:

- Performs L2 regularization, i.e. adds penalty equivalent to square of the magnitude of coefficients
- Minimization objective = LS Obj + α * (sum of square of coefficients).

3. Lasso Regression

LASSO stands for 'Least Absolute Shrinkage and Selection Operator'. Lasso regression performs L1 regularization, i.e. it adds a factor of sum of absolute value of coefficients in the optimization objective.

- Performs L1 regularization, i.e. adds penalty equivalent to absolute value of the magnitude of coefficients.
- Minimization objective = LS Obj + α * (sum of absolute value of coefficients).

4. Decision Tree Regression

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be "learned" by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner. The recursion is completed when the subset at a node has all the same value of the target variable.

5. KNN Regressor

K nearest neighbors is a simple algorithm that stores all available cases and predicts the numerical target based on a similarity measure (e.g., distance functions). A simple implementation of KNN regression is to calculate the average of the numerical target of the K nearest neighbors. Another approach uses an inverse distance weighted average of the K nearest neighbors. KNN regression uses the same distance functions as KNN classification.

6. ANN (KERAS)

- The methodology using Keras is a way to organize layers. The simplest way to do this is using the 'Sequential' feature. We have created `model = Sequential ()` and added different layers of neurons and activation functions for each layers of model.
- There are many activation functions to choose from and we started with 'Relu' since, the time in seconds or the 'Y' columns doesn't have any negative values. However, with some trial and error, the best option was to select the 'Sigmoid' function.
- The 'Kernal Initializers' define the way, to set the initial random weights of keras layers. We have used '`Kernal initializers = normal`'.
- We have used 4 layers with 500, 300, 150 and 75 neurons each. The last layer is the output layer with 1 neuron which is to estimate the 'time in seconds' and consists of 1 column only.
- An objective function is required to compile the model and we have used '`mean_squared_error`' as an indication of score. The optimizer used is 'Adam' which is an algorithm for first-order gradient based optimization of stochastic objective function.
- The variable 'train_x' is used to store the training dataset on which the 'train_y' values are fit. This will be further used to predict values. The validation split is 0.05 which is 5% of the 4207 rows available.
- The number of epochs used were, found by trial and error. It was primarily set to be 100, but beyond 30, it is trying to overfit the train set. Hence we stopped it at 30.
- The above model created is used to predict the values on the test set and since they are values, we can calculate the R^2 values. The train_set R^2 value is 0.564.

- The final R^2 values using just this method were found to be 0.49 which is a big improvement as compared with the regression models with accuracy 0.42.

CLASSIFICATION

1. Logistic Regression

Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured where there are only two possible outcomes. The dependent variable is binary or dichotomous, i.e. it only contains data coded as 1 (TRUE, success, present, etc.) or 0 (FALSE, failure, non-present, etc.). The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest and a set of independent variables. Logistic regression generates the coefficients (and its standard errors and significance levels) of a formula to predict a logit transformation of the probability of presence of the characteristic of interest

RESULTS

The train scores after implementing different regression models:

USING PCA		
Model	**R^2 Values**	**MSE**
Linear Regression	0.5357	76.2435
Ridge Regression	0.5483	74.2217
Lasso Regression	0.5460	74.6026
Decision Tree Regressor	-0.1300	181.2991
KNN Regressor	-0.051	167.8394

USING TSVD		
Model	**R^2 Values**	**MSE**
Linear Regression	0.4927	83.06134
Ridge Regression	0.4928	86.06116
Lasso Regression	0.4927	83.06133
Decision Tree Regressor	0.0344	155.2708
KNN Regressor	-0.0517	167.2846

Model	**Accuracy**	**MSE**
Logistic Regression	0.3352	-
ANN	0.5639	59.1438

WHAT COULD WE HAVE DONE DIFFERENTLY?

- Implementation of ANN is really flexible, and hence we could have fine tuned, activation functions with different number of neurons and layers. This was very time consuming for 500 epochs and hence wasn't tried comprehensively.
- We could have implemented one or more variable reduction techniques to the ANN and logistic model. The variable reduction techniques weren't tried on ANN and Logistic regression as ANN is supposed to be learning at a better rate. We would like to think that with relevant attributes it could have worked well.
- The classification of logistic model could have been done in a more comprehensive way to increase sensitivity or 'classification' of data. Since the data is highly skewed to the right, the classification is biased.
- Also increasing the sensitivity of the less classified data was challenging. This could have been accomplished by oversampling and under sampling techniques.

FUTURE SCORE

- We could have used 'XGBoost' model for improving the optimizing process.
- Other models like Random Forest, Bagging, Boosting, and Gradient Boosting can be tested for accuracy.

CONCLUSION

- Since the data is anonymous and not a lot of information can be derived from it, it is difficult to relate features, even after visualization.

- The length to breadth ratio of data is very large. By encoding some attributes, we are adding more attributes to the dataset. This could add a lot of noise to the data as well. It gets more challenging to model the same dataset with more attributes.
- Most of the attributes are available in binary format. The model may be prone to 'overfitting'.
- Many features in the dataset are highly co-linear implying they don't contribute much.