Regression Challenge:

Using Linear Regression and KNN Model

MD IMRAN HOSSAIN
Statistical Learning & Data Mining
Erasmus Mundus Joint Master in Medical Imaging and Applications
University of Cassino and Southern Lazio, Italy

Methodology

Data Inspection

Data Pre-processing

Training Model

Prediction

1. Visualize the training dataset and observe the relation between input variables and the result.

2. Imputing outliers and removing highly correlated input variables from the train data. Selecting input variables using the Backward Feature Selection and Forward Feature Selection algorithms for training models.

3. Splitting the training dataset into train and test (80:20) and fitting the pre-processed training dataset into Linear Regression and KNN Regression model.

4. Predicting the result using the training dataset and calculating the Mean Average Error (MAE), Root Mean Square Error (RMSE) and R-Squared scores for each model. Fitting testing dataset in the best model and importing results.

Linear Regression: Data Inspection & Outliers Imputation

Data Inspection

The train data consist of

- Number of input variables, **X** = **9**
- Result, **Y** = **1**
- Total number of samples, **N** = **1000**

Outliers Imputation

The train data has

- Outliers containing input variables = **v8** and **v9**
- Number of outliers in **v8** = **7**
- Number of outliers in **v9** = **5**
- Total number of outliers = **12**

The **mean Imputation** method is used to impute the outliers instead of removing outliers consisting of samples from the dataset because these samples may contain important information.

Fig-2a & 2b show the boxplots of train data with outliers and without outliers (after mean imputation).

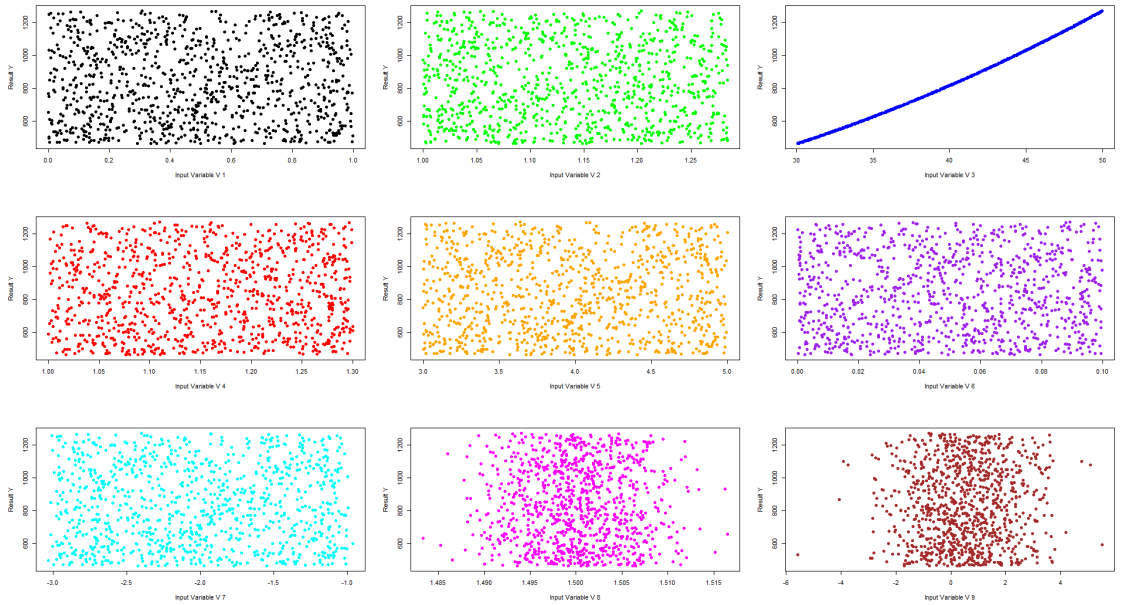

Fig-1: Visualization of the relation of input variables and results of the train data

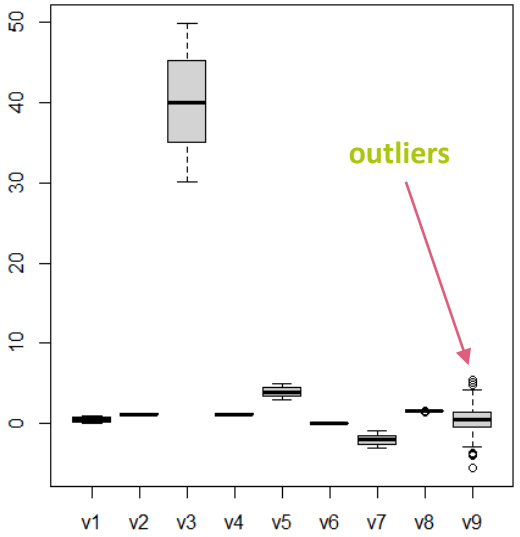

Fig-2a: Boxplot of train dataset with outliers

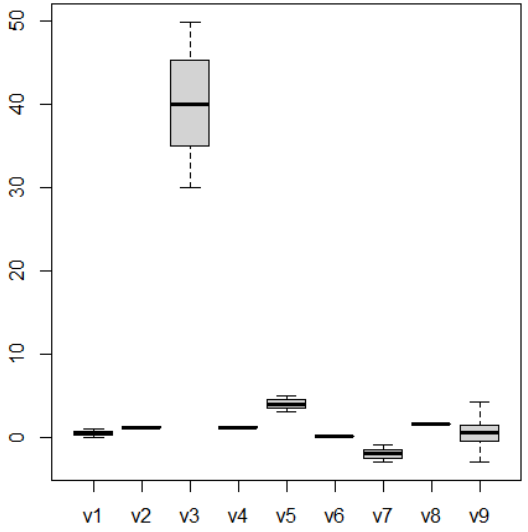

Fig-2b: Boxplot of train dataset without outliers

Linear Regression: Correlation & Training Model

Correlation

After observing the correlation among input variables of the train data, it is found that:

- Input variables **v1**, **v5**, and **v7** are **highly correlated (more than 0.9)** with each other.
- Hence, **v5** and **v7** are removed from the input variables of the train dataset.
- Total number of input variables, **X = 7**

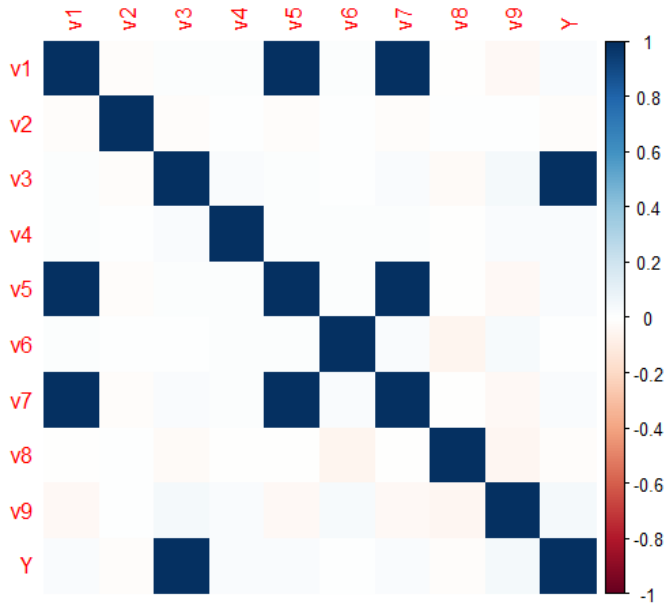

Fig-3: Correlation of the train data

Training Model

Backward Feature Selection:

- The linear regression model is trained after the elimination of input variables **(v5, v7)** of training data based on correlation.
- The input variables are eliminated one by one depending on the **p-value.**
- Finally, variables **(v1, v3)** are selected as they have the **lowest p-value** of **0.0422** and **2e-16** respectively.
- However, the training model with the combination of **v1** and **v3**, and only **v3** still provides high **Residual Standard Error (RSE)** of **14.88** and **14.90** respectively.

Forward Feature Selection:

- Using **Interactive terms, I(v1, v3)** and **Polynomial terms, I(v3, v3)** with the combination of input variables, the linear regression model is trained.
- It is observed that the combination of input variables and **Polynomial terms, I(v3, v3)** provides very **low p-value (< 1.20)** than the combination of input variables and **Interactive terms, I(v1, v3)**.
- Finally, the train model with **v1 + v2 + v3 + v4 + v6 + v8 + v9 + I(v3 * v3)** provides the minimum **p-value (0.1289)** and it is considered the **best model for prediction.**

Table-1: Backward Feature Selection

Model Name	Selected Variables (based on p-value)	R-Squared	Residual Standard Error
Model 1	v1+v2+v3+v4+v6+v8+v9	0.996	14.89
Model 2	v1+v3+v4+v6+v8+v9	0.996	14.88
Model 3	v1+v3+v4+v6+v8	0.996	14.88
Model 4	v1+v3+v6+v8	0.996	14.88
Model 5	v1+v3+v6	0.996	14.88
Model 6	v1+v3	0.996	14.88
Model 7	v3	0.996	14.90

Linear Regression: Prediction & Performance

Prediction & Performance

The entire train data is predicted using the best training **(Model 21)** and the prediction result provides the following scores:

Table-4: Train Data Prediction Performance

Regression Error Metrics	Scores
Mean Absolute Error (MAE)	**0.0096**
Root Mean Squared Error (RMSE)	**0.0128**
R-Squared	**1.0000**

- It is observed that prediction with **Model 21** provides the **minimum MAE** and **RMSE**, whereas the **maximum R-Squared**.
- Finally, the result of the test data is predicted using the best model **(Model 21)** in order to get accurate prediction results.

Table-2: Forward Feature Selection with Interactive Terms

Model Name	Selected Variables (based on p-value)	R-squared	Residual Standard Error
Model 8	v3+I(v1*v3)	0.996	14.88
Model 9	v1+v3+I(v1*v3)	0.996	14.88
Model 10	v1+v3+v6+I(v1*v3)	0.996	14.88
Model 11	v1+v3+v6+v8+I(v1*v3)	0.996	14.88
Model 12	v1+v3+v4+v6+v8+I(v1*v3)	0.996	14.89
Model 13	v1+v3+v4+v6+v8+v9+I(v1*v3)	0.996	14.89
Model 14	v1+v2+v3+v4+v6+v8+v9+I(v1*v3)	0.996	14.90

Table-3: Forward Feature Selection with Polynomial Terms

Model Name	Selected Variables (based on p-value)	R-squared	Residual Standard Error
Model 14	v3+I(v3*v3)	1	1.193
Model 16	v1+v3+I(v3*v3)	1	0.1065
Model 17	v1+v3+v6+I(v3*v3)	1	0.08375
Model 18	v1+v3+v6+v8+I(v3*v3)	1	0.08378
Model 19	v1+v3+v4+v6+v8+I(v3*v3)	1	0.08381
Model 20	v1+v3+v4+v6+v8+v9+I(v3*v3)	1	0.08385
Model 21	**v1+v2+v3+v4+v6+v8+v9+I(v3*v3)**	**1**	**0.01289**

KNN Regression: Data Normalization & Training Model

Data Normalization

The training data is normalized using the **min-max** normalization technique in order to reduce biasness of output values toward the k-value and input variables.

Model Training

Data Split:

The normalized training data is split into **train** and **test** with a ratio of **80** and **20** in order to train the **KNN** model using the **FNN** library.

Backward Feature Selection:

- At first, the **KNN** model is trained with all input variables (except **v5** and **v7** because of the high correlation factor) and the optimal **K** is selected based on the minimum **Root Mean Square Error (RMSE)** using the iteration technique.
- Input variables are eliminated one by one for training the model depending on the **p-value**.
- Finally, **KNN** is trained with the combination of **v1** and **v3** input variables only and it provides the minimum **RMSE** score **(0.0129)** with an **optimal K** value of **6**.

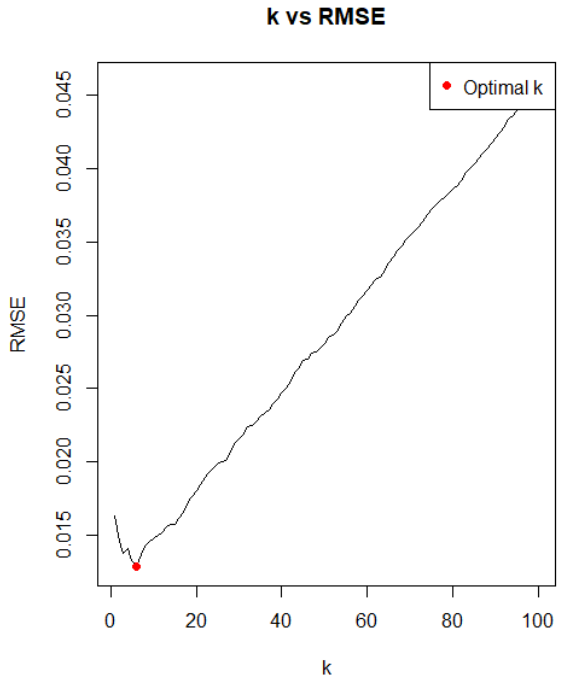

Fig-4: Plot of k vs RMSE

Table-5: Forward Feature Selection

Model Name	Selected Variables (based on p-value)	K (optimal)	RMSE
Model 1	v1+v2+v3+v4+v6+v8+v9	6	0.0650
Model 2	v1+v3+v4+v6+v8+v9	8	0.0580
Model 3	v1+v3+v4+v6+v8	9	0.0473
Model 4	v1+v3+v6+v8	4	0.0381
Model 5	v1+v3+v6	7	0.0295
Model 6	v1+v3	**6**	**0.0129**

KNN Regression: Prediction & Performance

Prediction & Performance

The training data is predicted using the best training **(Model 6)** and the prediction result provides the following scores:

- It is observed that prediction with **Model 6** provides the **minimum MAE** and **RMSE**, whereas the **maximum R-Squared**.
- Finally, the result of the test data is predicted using the best model **(Model 6)** in order to get accurate prediction results.

Table-6: Train Data Prediction Performance

Regression Error Metrics	Scores
Mean Absolute Error (MAE)	**0.00774**
Root Mean Squared Error (RMSE)	**0.01030**
R-Squared	**0.99788**

Linear Regression & KNN Regression

The **Table-7** indicates a compassion of performance between KNN model and Linear Regression model:

- In the case of **Mean Absolute Error (MAE)** and **Root Mean Squared Error (RMSE)**, the KNN model performed better (with less error) than the Linear Regression model.
- In the case of **R-Squared**, the KNN model performed worse than the Linear Regression model.

Table-7: Performance between KNN and Linear Model

Regression Error Metrics	KNN	Linear
Mean Absolute Error (MAE)	**0.0077**	**0.0096**
Root Mean Squared Error (RMSE)	**0.0103**	**0.0128**
R-Squared	**0.9978**	**1.0000**