©2020 Published in 8th International Symposium on Innovative Technologies in Engineering and Science 23-25 October 2020 (ISITES2020 Bursa - Turkey)
https://doi.org/10.33793/acperpro.03.01.31

On-Line Spot Welding Quality Prediction with Analysis of Welding Parameters

[1]Emin Cantez, [1]İsmail Atalay, [1]Oğuz Alper İsen, [1]Serkan Aydın
[1]Coşkunöz Kalıp Makina Sanayi ve Ticaret A.Ş., Bursa, Turkey

Abstract

Spot welding is one of the metal joining technologies and has an important place especially in the automotive industry. A passenger car has average 5000 spots. Destructive inspection is carried out at certain periods to check these spots. However, not all parts can be checked. In this work, welding parameters were collected and analyzed. By applying different machine learning methods, the quality of the spot welding was tried to be estimated and the results were compared.

Key words: Data Acquisition, Quality Prediction, Manufacturing Science, Intelligent Manufacturing

1. Introduction

Resistance spot welding (RSW) is a simple and cost-effective manufacturing process which is extensively used for joining sheet steel in the automobile industry due to its high speed and adaptability for automation. The number of RSW joints per vehicle is very high (usually 5000 spots)[1] and the tendency in the highly competitive automobile industry is to reduce it as much as possible.

In RSW, the two electrode caps of the welding gun (see Figure 1) press two or three worksheets between the electrodes with force. An electric current then flows from one electrode, through the worksheets, to the other electrode, generating a substantial amount of heat as a result of electric resistance. The materials in a small area between the two worksheets, known as the welding spot, will melt, and form a weld nugget connecting the worksheets. The electrode caps directly touching the worksheets wear out easily due to high thermo-mechanical loads and oxidation, and need to be changed on a regular basis.

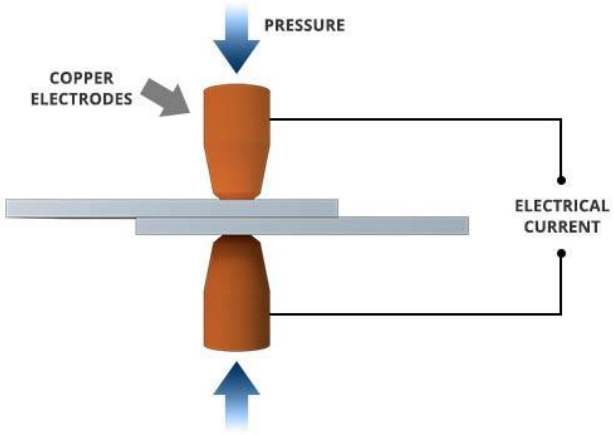

Figure 1: Schematic diagram of resistance spot welding

The inclusion of Industry 4.0 in factories, these important processes can be predicted with machine learning models and errors that may occur during production can be prevented. The inclusion of Industry 4.0 in factories, these important processes can be predicted with machine learning models and errors that may occur during production can be prevented. Using the BOS6000 program, the data collected through the BOSCH branded spot welding system can be labeled and the desired and unwanted situations can be reported. With this set of data, results can be observed using regression modeling.

2. Materials and Method

The quality of the welding spots is typically quantified by the weld nugget diameter as defined in the international standards (ISO, 2004) and German standard (DVS, 2016). The welding quality is then determined as accepted or rejected according to specific tolerance bands. The data collected in this study (see table 1) are written to the SQL database created with a communication method established over MQTT[7]. In this study 450 samples were collected and all were labeled from manufacturing line.

The approach to discover answers during the spot welding process starts with applying machine learning models. As it seems obvious the dataset of spot welding is a regression problem. It is planned to make a real-time experiment with the model that evaluates different solutions for this problem and gives the most optimistic result.

First, some simple features are extracted from the time series. Extracted properties are properties that are not directly related to the resource diameter, these are "TWLD1", "TWLD3", "CAP1", "TIP1", "IWLD3", "TIP3", "RON3", "WLDMTR", "IWLD1", "QualityClassification". Other of them evaluated along with all other single attributes and selected using the forward step selection, starting with the evaluation of each feature and progressively to add more features. Properties that result in models with the best regression Accuracy (RMSE) is chosen to estimate the point diameters. After feature selection, some machine learning methods different data-driven models. These models are compiled in the Python[6] language with the Scikit-Learn library[3].

The feature extraction and machine learning models[2] are selected with given best performance, for better generalization and automation in other similar applications in automatic manufacturing. The characteristics of the eight machine learning methods and the reasons for choosing them are explained below.

•**Linear Regression** fits a linear model with coefficients w=(w1,...,wp) to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation. Mathematically it solves a problem of the form:

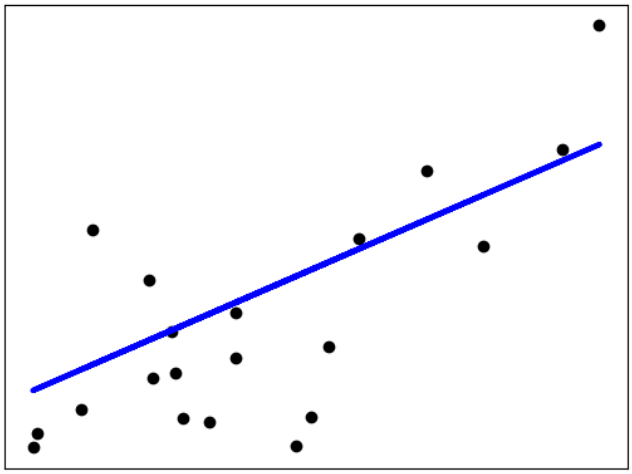

Figure 2: Linear Regression graph example

• **Ridge regression** addresses some of the problems of Ordinary Least Squares by imposing a penalty on the size of the coefficients. The ridge coefficients minimize a penalized residual sum of squares:

The complexity parameter $\alpha \geq 0$ controls the amount of shrinkage: the larger the value of α, the greater the amount of shrinkage and thus the coefficients become more robust to collinearity.

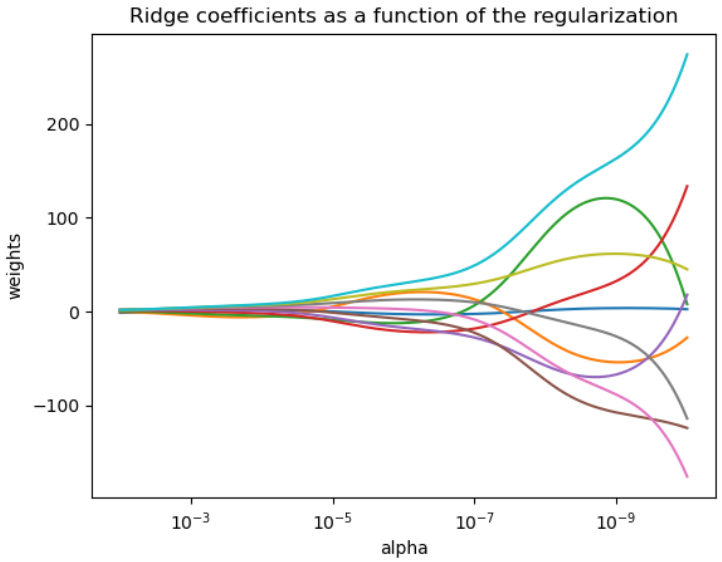

Figure 3: Rigde Regression graph

• **Lasso Regression** is a linear model that estimates sparse coefficients. It is useful in some contexts due to its tendency to prefer solutions with fewer non-zero coefficients, effectively reducing the number of features upon which the given solution is dependent. For this reason Lasso and its variants are fundamental to the field of compressed sensing. Under certain conditions, it can recover the exact set of non-zero coefficients (see Compressive sensing: tomography reconstruction with L1 prior (Lasso)).

Mathematically, it consists of a linear model with an added regularization term. The objective function to minimize is:
The lasso estimate thus solves the minimization of the least-squares penalty with added, where is a constant and is the -norm of the coefficient vector.

• **KNeighborsRegressor** implements learning based on the nearest neighbors of each query point, where is an integer value specified by the user.

The basic nearest neighbors regression uses uniform weights: that is, each point in the local neighborhood contributes uniformly to the classification of a query point. Under some circumstances, it can be advantageous to weight points such that nearby points contribute more to the regression than faraway points. This can be accomplished through the weights keyword. The default value, weights = 'uniform', assigns equal weights to all points. weights = 'distance' assigns weights proportional to the inverse of the distance from the query point. Alternatively, a user-defined function of the distance can be supplied, which will be used to compute the weights.

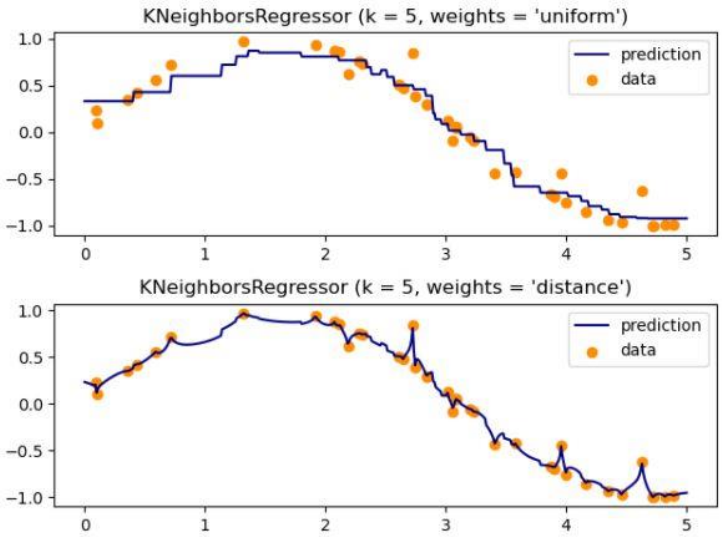

Figure 4:
KNeighbors graph

• **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.

For instance, in the example below, decision trees learn from data to approximate a sine curve with a set of if-then-else decision rules. The deeper the tree, the more complex the decision rules and the fitter the model.

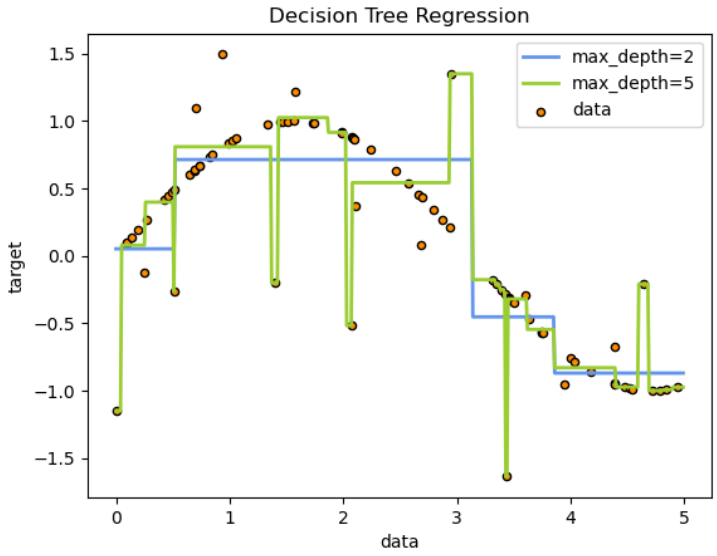

Figure 5: Decision Tree graph

• **Random forest** is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the max_samples parameter the whole dataset is used to build each tree

• **GradientBoostingRegressor** supports a number of different loss functions for regression which can be specified via the argument loss; the default loss function for regression is least squares ('ls').

The figure below shows the results of applying GradientBoostingRegressor with least squares loss and 500 base learners to the Boston house price dataset (sklearn.datasets.load_boston). The plot on the left shows the train and test error at each iteration. The train error at each iteration is stored in the train_score_ attribute of the gradient boosting model. The test error at each iterations can be obtained via the staged_predict method which returns a generator that yields the predictions at each stage. Plots like these can be used to determine the optimal number of trees (i.e. n_estimators) by early stopping. The plot on the right shows the impurity-based feature importances which can be obtained via the feature_importances_

property.stage. Plots like these can be used to determine the optimal number of trees (i.e. n_estimators) by early stopping. The plot on the right shows the impurity-based feature importances which can be obtained via the feature importance property.

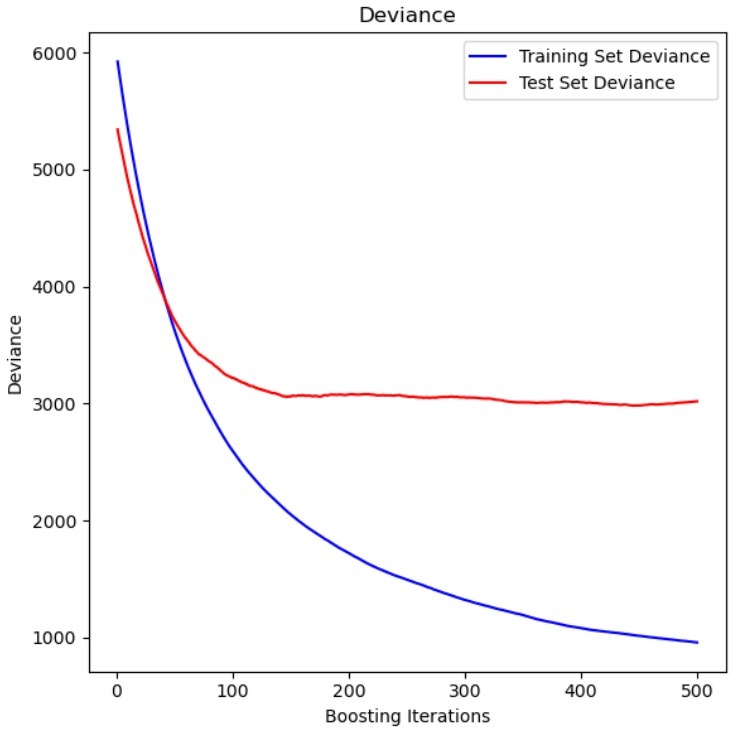

Figure 6: Gradient Boosting graph

• The core principle of AdaBoost is to fit a sequence of weak learners (i.e., models that are only slightly better than random guessing, such as small decision trees) on repeatedly modified versions of the data. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction. The data modifications at each so-called boosting iteration consist of applying weights to each of the training samples. Initially, those weights are all set to , so that the first step simply trains a weak learner on the original data. For each successive iteration, the sample weights are individually modified and the learning algorithm is reapplied to the reweighted data. At a given step, those training examples that were incorrectly predicted by the boosted model induced at the previous step have their weights increased, whereas the weights are decreased for those that were predicted correctly. As iterations proceed, examples that are difficult to predict receive ever-increasing influence.

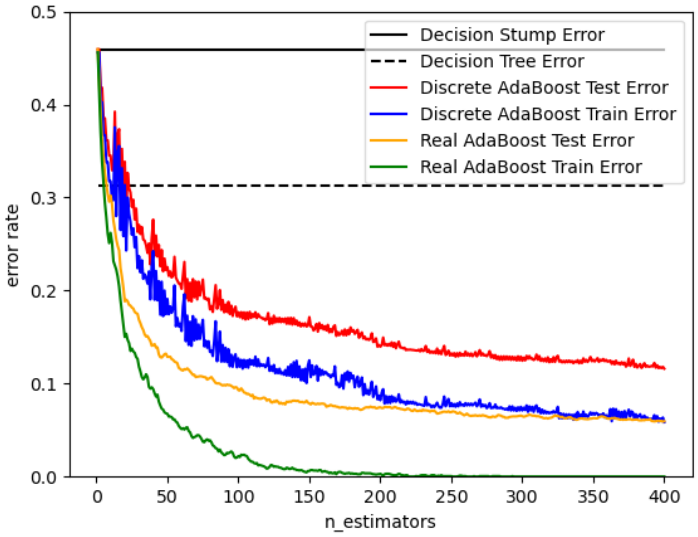

Figure 7: Adaboots graph

2.1. Theory/calculation

The hyper-parameters of these models are chosen with 3-fold cross validation with exemplary training datasets. For all machine learning models trained on subsets without noise, the hyper-parameters are chosen with the non-noisy subset of all training data (with 414 training data points) and the Production Feature Set. For all machine learning models trained with subsets with noise, this is performed again with the noisy subset of all training data (with 414 training data points).

Machine Learning Model	Hyper-parameter	Value
Decision Tree Regressor	n_estimators	5
Ridge Regression	Alpha	0.5
Lasso Regression	Alpha	0.1
K Neighbors	k	5
Random Forest Regressor	n_estimators	100
Gradient Boosting Regressor	n_estimators	500
Adaboots Regressor	n_estimators	100

Table 1: Hyperparameters

3. Results

Python is preferred as the programming language during the development process in this paper. Scikit-learn [4] package are used for regression.

Seven regression models are trained which are ;
DecisionTreeRegressor ,
Ridge Regression,
Lasso Regression,
K Neighbors,
Random Forest Regressor,
Gradient Boosting Regressor,
Adaboots Regressor

All data set splits into random train and test subsets[5]. Test size is determined as 0.25 which represents proportion of dataset to include in the test split. Seven models compared over the RMSE and R Squared.Both of them is based on following structure as showing figure [8].

FOLLOWING ARE THE TRAINING SCORES:

	Model	RMSE	R Squared
0	Linear Regression	0.835027	0.860501
1	Ridge Regression	0.835027	0.860529
2	Lasso Regression	1.022843	0.801818
3	K Neighbors Regressor	0.828517	0.867357
4	Decision Tree Regressor	0.729453	0.880250
5	Random Forest Regressor	0.585789	0.932029
6	Gradient Boosting Regressor	0.553696	0.937680
7	Adaboost Regressor	0.666221	0.918190

Figure 8: Regression models result

In addition, we have reached a heat map that separates the variables that affect the weld diameter from this information. With this map, we can determine which variable has an effect on the result and we can adjust our model accordingly. This table is important for the improvements to be made in this study. Feature extraction is done at this step.

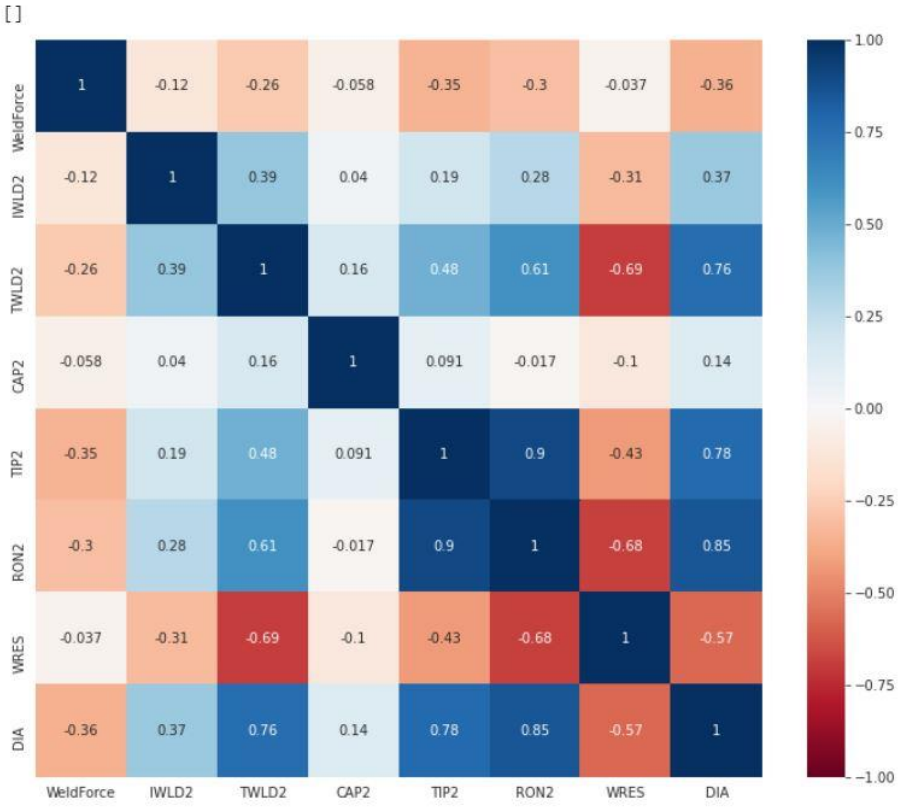

Figure 9: Heatmap graph

The difference between predicted values and actual values are enough for our gap which means the model can be deployed in real time with spot welding process.

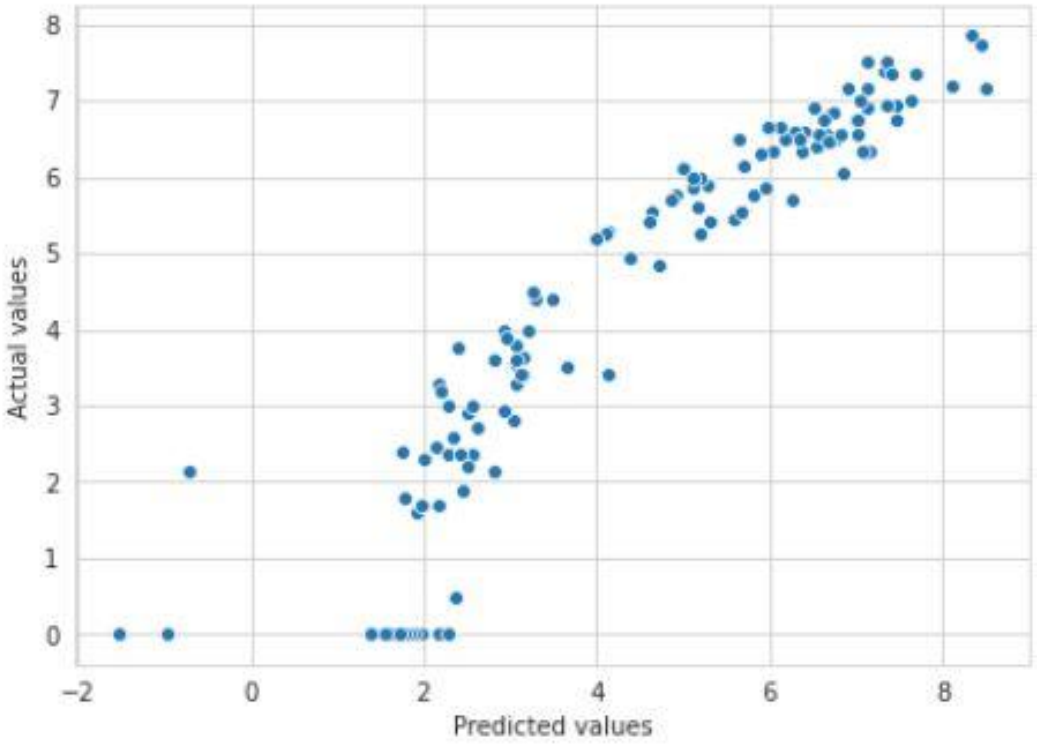

Figure 10: Difference between values

3.3. Math formulae

Linear Regression:

The following are a set of methods intended for regression in which the target value is expected to be a linear combination of the features. In mathematical notation, if $\hat{y}$ is the predicted value.

$$\hat{y}(w, x) = w_0 + w_1 x_1 + \ldots + w_p x_p$$

Figure 11: Regression formula

Across the module, Designate the vector **w=(w1,...,wp)** as **coef** and **w0** as intercept.

Rigde Regression:

$$\min_w ||Xw - y||_2^2 + \alpha ||w||_2^2$$

Figure 12: Ridge regression formula

Lasso Regression:

$$\min_w \frac{1}{2n_{\text{samples}}} ||Xw - y||_2^2 + \alpha ||w||_1$$

Figure 13: Lasso regression formula

Root Mean Squared Error:

$$\text{MSE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2.$$

Figure 14: RMSE formula

R Squared Error:

$$\text{R}^2 = 1 - \frac{\text{Unexplained Variation}}{\text{Total Variation}}$$

Figure 15: R Squared formula

4.Discussion

With these results, the data received from the field were classified correctly and incorrectly according to the entered parameter values. If the regression values obtained are greater or less than the expected diameter value at a certain precision, we can make this classification. However, this study depends on the accuracy of the data received from the field. The model created during real time operation must be retried. Or when working with data that has not been added to the data before, deviations may occur, and more data should be collected to prevent them.

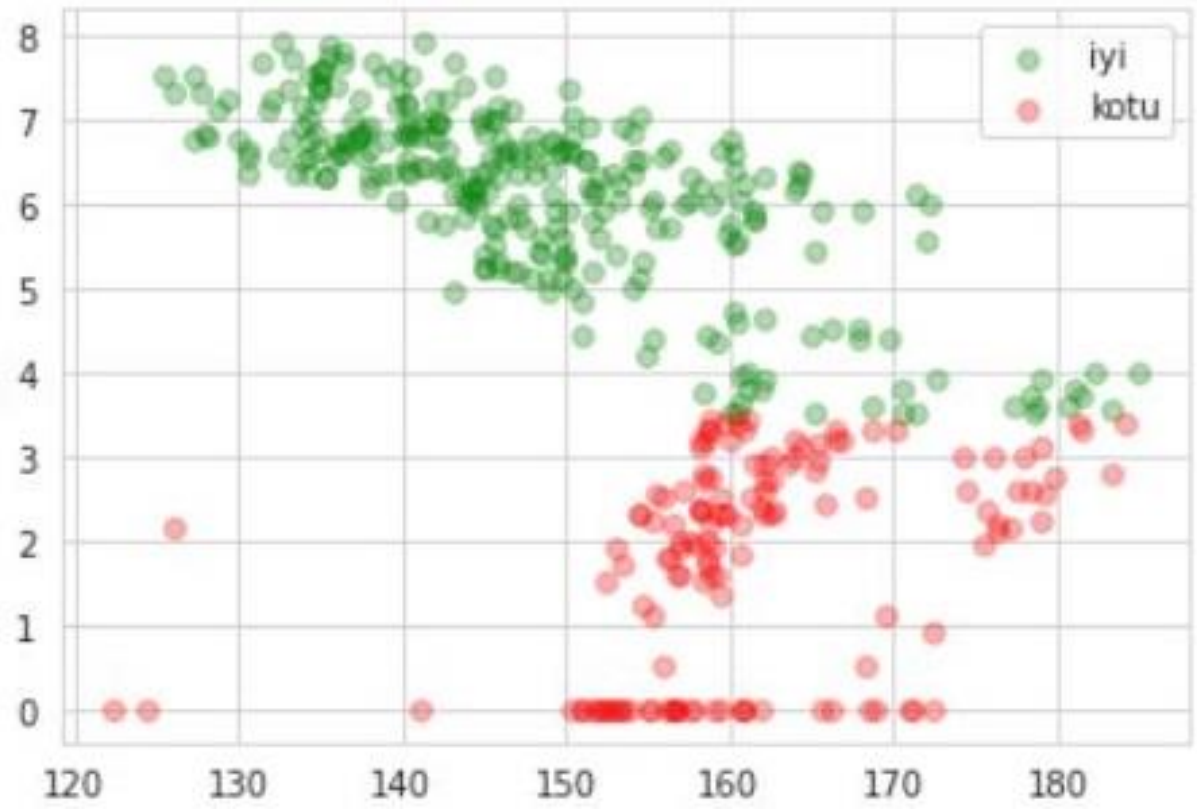

Figure 16: Classification graph

Conclusions

The results of the studies focused on spot welding quality detection, which are important topics for quality control processes, predictive maintenance and intelligent manufacturing in automotive sector, are presented. Regression methods are trained by using machine learning algorithms among eight regression model, Gradient Boosting model has the highest R Square scores with 0.937. Random Forest and Adaboots models produced close scores with 0.931 and 0.912 respectively. According importance of the feature which is shown in Figure [x], contrast stands out as the most effective feature among all features on results.

Acknowledgements

This research is supported Coşkunöz Kalıp Makina Sanayi ve Ticaret A.Ş . Coşkunöz technicians supported to collect data from robots and financial support were given by R&D department.

REFERENCES

[1] https://learnmech.com/introduction-to-spot-welding-working/
[2] http://www.sthda.com/english/wiki/regression-analysis-essentials-for-machine-learning
[3] https://scikit-learn.org/stable/user_guide.html
[4] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
[5] Cassidy, A. P.; Deviney, F. A. Calculating Feature Importance in Data Streams with Concept Drift Using Online Random Forest. 2014 IEEE International Conference on Big Data (Big Data), 2014.
[6] https://docs.python.org/3/
[7] https://mosquitto.org/documentation/