



Supervised Learning: Regression

IBM Machine Learning - Project
Mohammed Qasim K.
October 2022



Main Objective

- The main objective of this analysis was to predict the Temperature of the permanent magnet synchronous motor (PMSM) using the Linear Regression Method and different Regularization Regression techniques.
- Both Train-Test-Split and Cross-Validation were used to understand how these methods can lead to different decisions in selecting the model.
- The data set was split into Training set (60%), Validation set (20%), and Test set (20%) for Cross-Validation purpose.



About the Data

- The data set comprises of several sensor data collected from a permanent magnet synchronous motor (PMSM) deployed on a test bench. Test bench measurements were collected by the LEA department at Paderborn University.
- This data set has 1330816 records and 13 variables. During the analysis, no duplicates were detected.

<https://www.kaggle.com/datasets/wkirgsn/electric-motor-temperature>

Variable name	Type	Description
u_q	float	Voltage q-component measurement in dq-coordinates (in V)
coolant	float	Coolant temperature (in °C)
stator_winding	float	Stator winding temperature (in °C) measured with thermocouples
u_d	float	Voltage d-component measurement in dq-coordinates
stator_tooth	float	Stator tooth temperature (in °C) measured with thermocouples
motor_speed	float	Motor speed (in rpm)
i_d	float	Current d-component measurement in dq-coordinates
stator_yoke	float	Stator yoke temperature (in °C) measured with thermocouples
i_q	float	Current q-component measurement in dq-coordinates
ambient	float	Ambient temperature (in °C)
torque	float	Motor torque (in Nm)
profile_id	int	Measurement session id. Each distinct measurement session can be identified through this integer id.
pm	float	Permanent magnet temperature (in °C) measured with thermocouples and transmitted wirelessly



Data Exploration

- After checking for duplicates, the EDA was conducted on the training set.
- The data description is as follows:

Data columns (total 13 columns):

#	Column	Non-Null Count	Dtype
0	u_q	798489 non-null	float64
1	coolant	798489 non-null	float64
2	stator_winding	798489 non-null	float64
3	u_d	798489 non-null	float64
4	stator_tooth	798489 non-null	float64
5	motor_speed	798489 non-null	float64
6	i_d	798489 non-null	float64
7	i_q	798489 non-null	float64
8	pm	798489 non-null	float64
9	stator_yoke	798489 non-null	float64
10	ambient	798489 non-null	float64
11	torque	798489 non-null	float64
12	profile_id	798489 non-null	int64

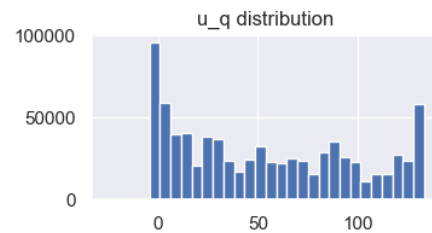
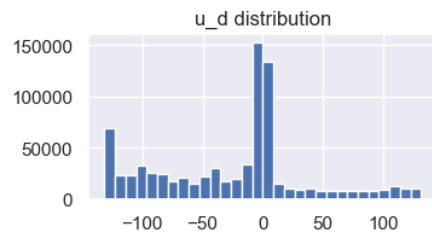
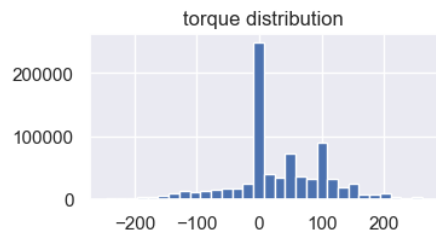
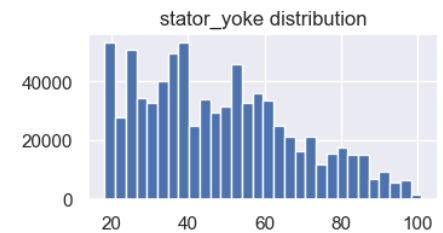
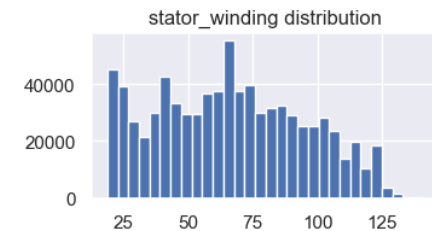
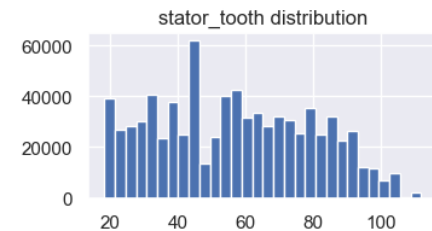
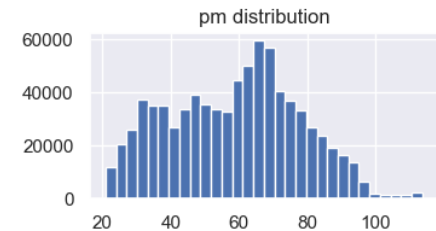
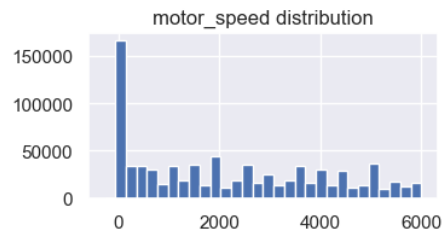
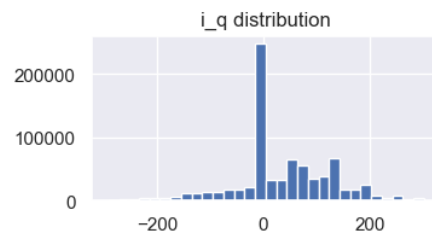
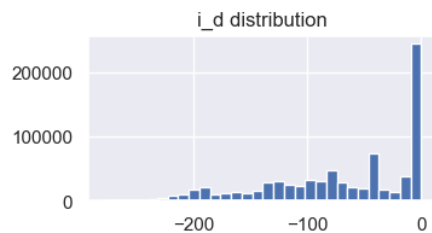
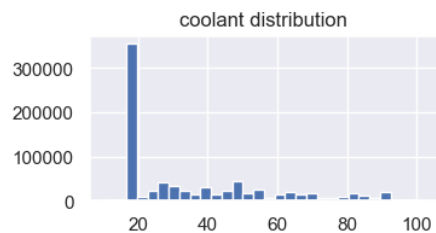
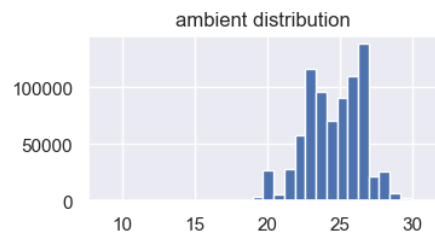
dtypes: float64(12), int64(1)



Data Exploration

- Dropped *profile_id* column.

	u_q	coolant	stator_winding	u_d	stator_tooth	motor_speed	i_d	i_q	pm	stator_yoke	ambient	torque	profile_id
count	798489.000000	798489.000000	798489.000000	798489.000000	798489.000000	798489.000000	798489.000000	798489.000000	798489.000000	798489.000000	798489.000000	798489.000000	798489.000000
mean	54.349174	36.245581	66.358853	-25.179385	56.895054	2204.901029	-68.751430	37.429517	58.527916	48.202727	24.565918	31.119940	40.803950
std	44.197834	21.792039	28.670377	63.095767	22.950999	1859.783391	64.937905	92.106282	18.997146	19.991096	1.930457	77.079573	25.040415
min	-25.290930	10.623751	18.585815	-131.530411	18.161402	-275.549144	-278.003632	-293.426793	20.856956	18.108742	8.783478	-246.451070	2.000000
25%	12.071892	18.698446	42.790879	-78.876443	38.423805	327.322631	-115.464500	1.095860	43.210197	32.003128	23.184975	-0.136527	17.000000
50%	48.957611	26.911880	65.163162	-7.515211	56.085577	1999.976807	-51.217891	15.774040	60.291509	45.659646	24.798018	10.865477	43.000000
75%	90.285187	49.857246	88.143259	1.469595	75.571545	3767.215100	-2.979726	100.494049	71.987605	61.452286	26.218862	91.211845	65.000000
max	133.036994	101.598512	141.362885	131.469788	111.946423	5999.991211	0.051897	301.707855	113.600159	101.114938	30.629913	261.005707	81.000000





Data Exploration

- Distributions of *coolant* and *motor_speed* were found to be right-skewed and the *i_d* distribution was found to be left-skewed.
- Square Soot Transformation was used to eliminate the skewness in these cases.
- The target variable (pm) was kept the unchanged.

Data Exploration

After applying Square Root Transformations of skewed features (only *coolant* skewness was above threshold value of 0.75)

The pair plot shows that:

- torque has a linear relationship with i_q .





Data Exploration

Severe Multicollinearity was found to exist in the data set based on the Variation Inflation Factor(VIF) values.

	variables	VIF
0	u_q	13.236383
1	coolant	354.571080
2	stator_winding	1263.515990
3	u_d	5.559511
4	stator_tooth	4766.619433
5	motor_speed	35.344533
6	i_d	22.014587
7	i_q	298.474640
8	stator_yoke	2086.045245
9	ambient	106.536704
10	torque	338.068029



Feature Engineering and Model Variations

Feature engineering was applied in order to create model variations. Each model was evaluated based on its Root Mean Square Error (RMSE).

- Applied Square Root Transformation to features that have a skew value greater than 0.75 (*coolant*)
- Scaled numerical features
- Added polynomial features

All these feature engineering steps were performed on training and validation sets, using a K-fold cross-validation with $k=5$.



Feature Engineering and Model Variations

- RMSEs of validation sets were slightly higher than training sets, which is expected but there were no signs of overfitting.
- The transformations improves all models.
- Scaling features is for regularization later. RMSEs of both training set and validation set will stay the same.




Feature Engineering and Model Variations

- Polynomial Features were added to the latest model (Square Root Transformed and Scaled) and the model was fit again.
- Only up to second polynomial degree transformation was possible due to system restrictions using such a large data set and is assumed to provide the best result as a greater degree transformation would lead to a higher number of features and complexity risking overfitting.



Cross-Validation and Regularization

- Using the same data pipeline: Square Root Transformation, Standard Scaling, and Polynomial Features Addition.
- Cross-validation is used to fit the linear regression model again, and then the hyperparameters are tuned to find a proper combination of alpha and polynomial degree for regularization. Regularized models include Lasso, Ridge, and Elastic Net.
- Each model was evaluated based on its average Root Mean Squared Error (from 5 folds).



Cross-Validation and Regularization

- Iterations over different polynomial degrees (1, 2,) and alphas.
- Results are sorted by RMSE in ascending order.

Linear

	Average RMSE
Degree = 3	1590.404635
Degree = 2	1612.824436
Degree = 1	1716.996101
Degree = 4	2080.145656
Degree = 5	8573.970563
Degree = 6	167125.925844

Ridge

	Average RMSE
Degree = 3, alpha = 0.005	1590.408087
Degree = 3, alpha = 0.01	1590.411623
Degree = 3, alpha = 0.05	1590.442776
Degree = 3, alpha = 0.1	1590.488293
Degree = 3, alpha = 0.3	1590.729348

Lasso

	Average RMSE
Degree = 3, alpha = 0.3	1589.662468
Degree = 3, alpha = 0.1	1589.918209
Degree = 3, alpha = 0.05	1590.094432
Degree = 3, alpha = 0.01	1590.336092
Degree = 3, alpha = 0.005	1590.369968

Elastic Net

	Average RMSE
Degree = 3, alpha = 0.005	1659.788222
Degree = 2, alpha = 0.005	1662.664297
Degree = 2, alpha = 0.01	1698.958361
Degree = 3, alpha = 0.01	1703.922403
Degree = 1, alpha = 0.005	1797.426905



Cross-Validation and Regularization

- The error metrics among Lasso, Ridge, and Linear regression were not significantly different.
- The best model found was Ridge Regression with polynomial degree of 3 and alpha equal to 0.005.
- Elastic Net had the highest RMSE value.

	Average RMSE	Average R2
Model		
Lasso	1589.662468	0.887138
Linear	1590.404635	0.887033
Ridge	1590.408087	0.887033
Elastic Net	1659.788222	0.876740

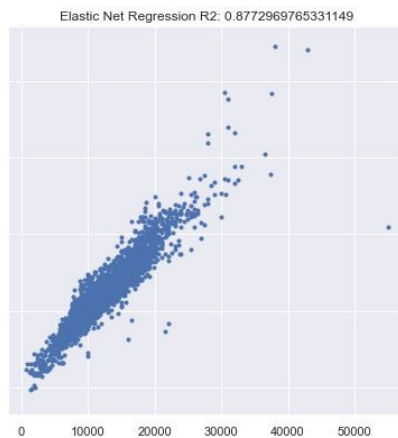
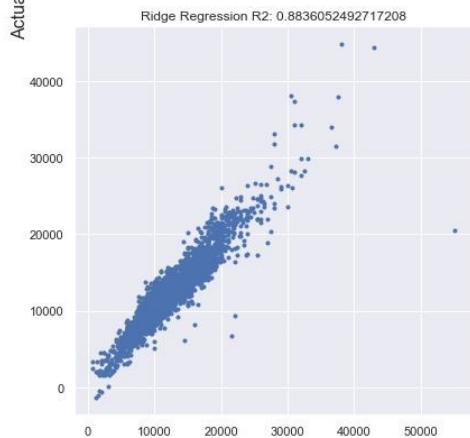
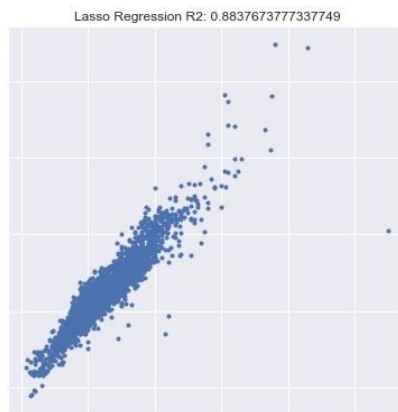
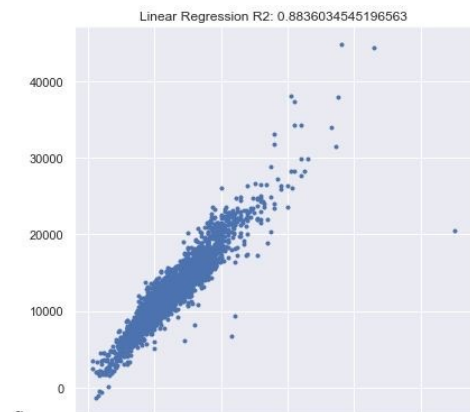


Predictions on the Test Set

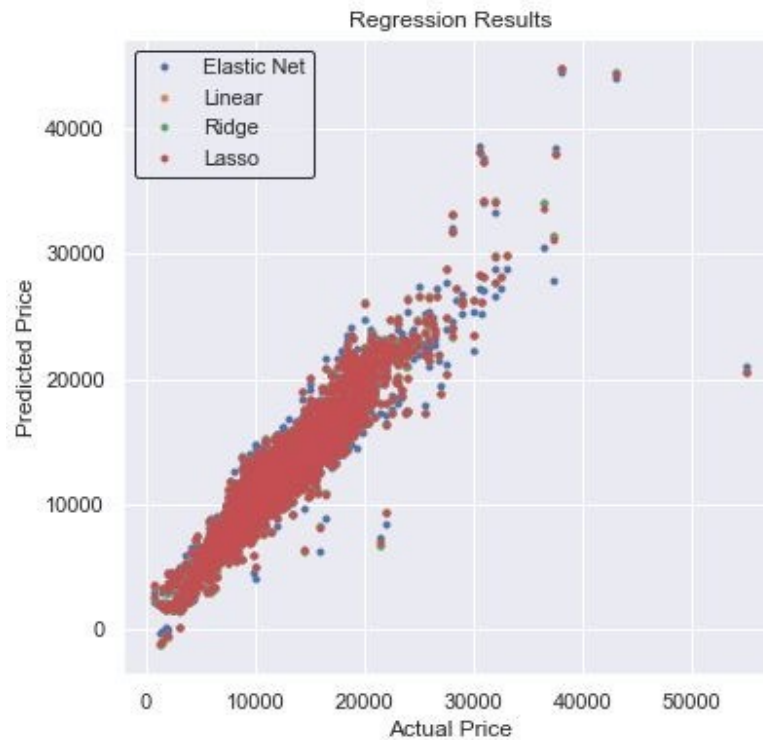
The four models were then fit on the unseen Test Set and the R2 score was calculated for each model.

- Linear Regression with third degree polynomial features
- Lasso Regression with third degree polynomial features and $\alpha = 0.3$
- Ridge Regression with third degree polynomial features and $\alpha = 0.005$
- Elastic Net Regression with third degree polynomial features and $\alpha = 0.005$

Next page shows the scatter plots (true vs predicted pm) and the R2 scores.



Predicted Price





Predictions on the Test Set

- The plots show that all the four models performed pretty well with no signs of overfitting. R2 scores among Lasso, Ridge, and Linear Regression were not significantly different.
- Lasso Regression was the best model ($R^2=0.8837$), and it also shrunk some of the coefficients.

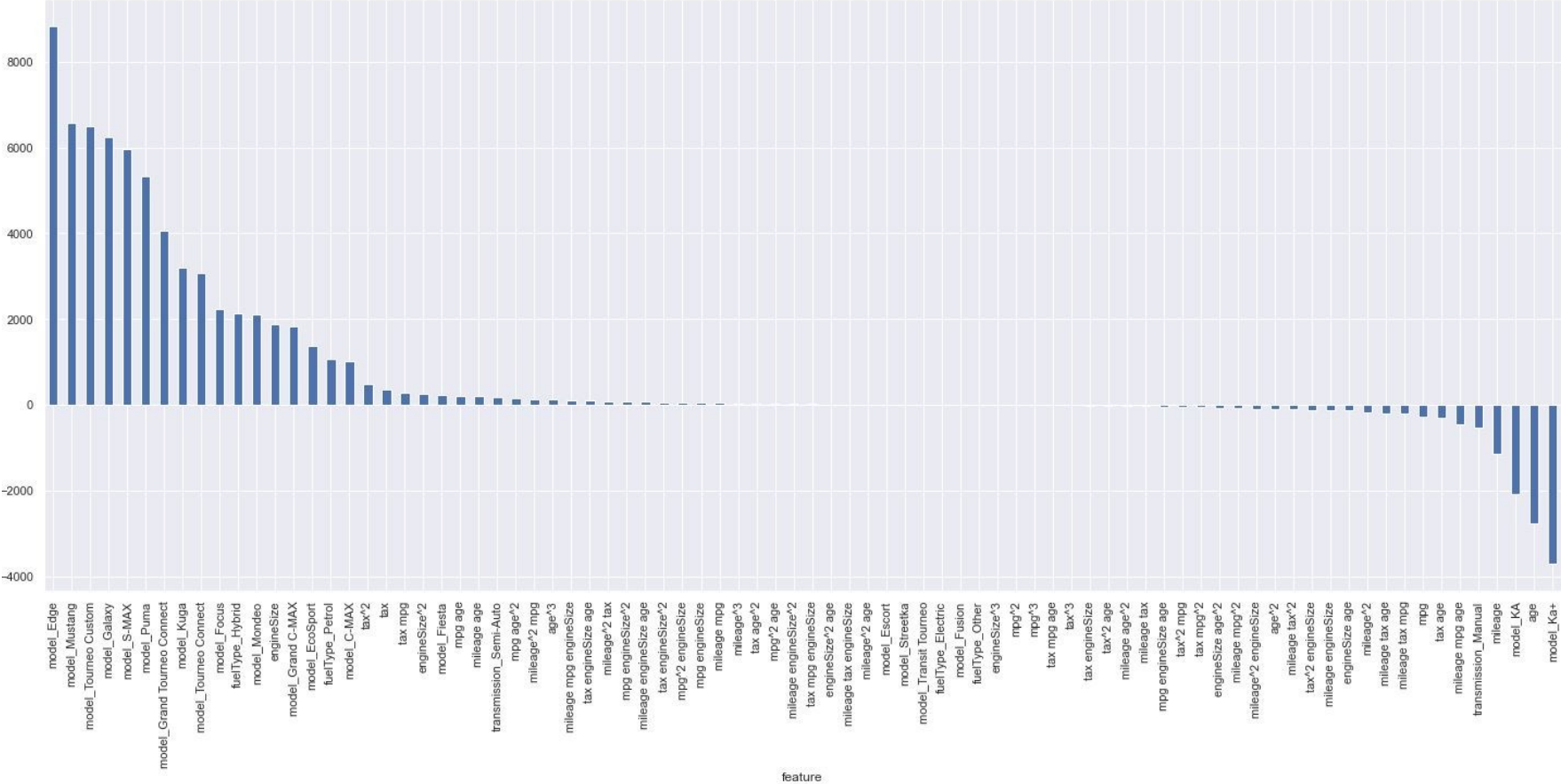
	feature	estimate
45	model_Escort	0.0
46	model_Streetka	0.0
47	model_Transit Tourneo	0.0
48	fuelType_Electric	0.0
49	model_Fusion	0.0
50	fuelType_Other	0.0



Predictions on the Test Set

The next page shows a plot of feature importance of the Lasso Regression. Among the different features, xxx and xxx have the strongest predictive power. Most interaction terms and polynomial features have low estimates in comparison to others.

Feature Importance





Conclusion

The analysis shows that Feature Engineering has a large effect on the model performance, and if the data is sufficiently large, Cross-Validation should be preferred over Train-Test-Split to perform the model evaluation. In this case, even though the predictors have high Multicollinearity, their coefficients were not shrunk by the Lasso Model, and it was shown that Regularization does not drastically improve a given model. The Lasso Regression had the highest R^2 value when predicting on the test set, and xxx appear to be the most important features to predict the motor temperature. Also, the Lasso model shrunk some of the features that do not contribute significantly to the prediction.

Jupyter Notebook can be found here:

<https://github.com/thuynh323/IBM-Machine-Learning/blob/master/2-Supervised-Learning-Regression/Project-2.ipynb>