ESTIMATION OF THE SURFACE SHRINKAGE RATE BASED ON THE EFFECT OF DIFFERENT WASHING ALGORITHMS VIA MACHINE LEARNING APPROACH

SEN MÜHENDİSSİN BİZİMLESİN PROJECT

Prepared by: Belce KAYA

Project Mentors: Zehra YALÇIN

Gökhan SIR

EXECUTIVE SUMMARY

The aim of the project is to analyze first input features with EDA (Explotary Data Analysis) methodology by using Excel, Tableau and Python programs (such as descriptive analysis, correlation computation between inputs and also output, detecting outliers with statistic tests, OLS regression for backward selection to be able to analyze features' p-values, r^2 score, feature importance scores after fitting the ML algorithms…) and then to install the model for estimation on surface area shrinkage rate of woolen textiles.

For the best model, bagging and boosting ensemble learning algorithms and linear regression will be used, also to avoid high variance and overfitting/underfitting, grid search method will be used to optimize algos' hyper params.

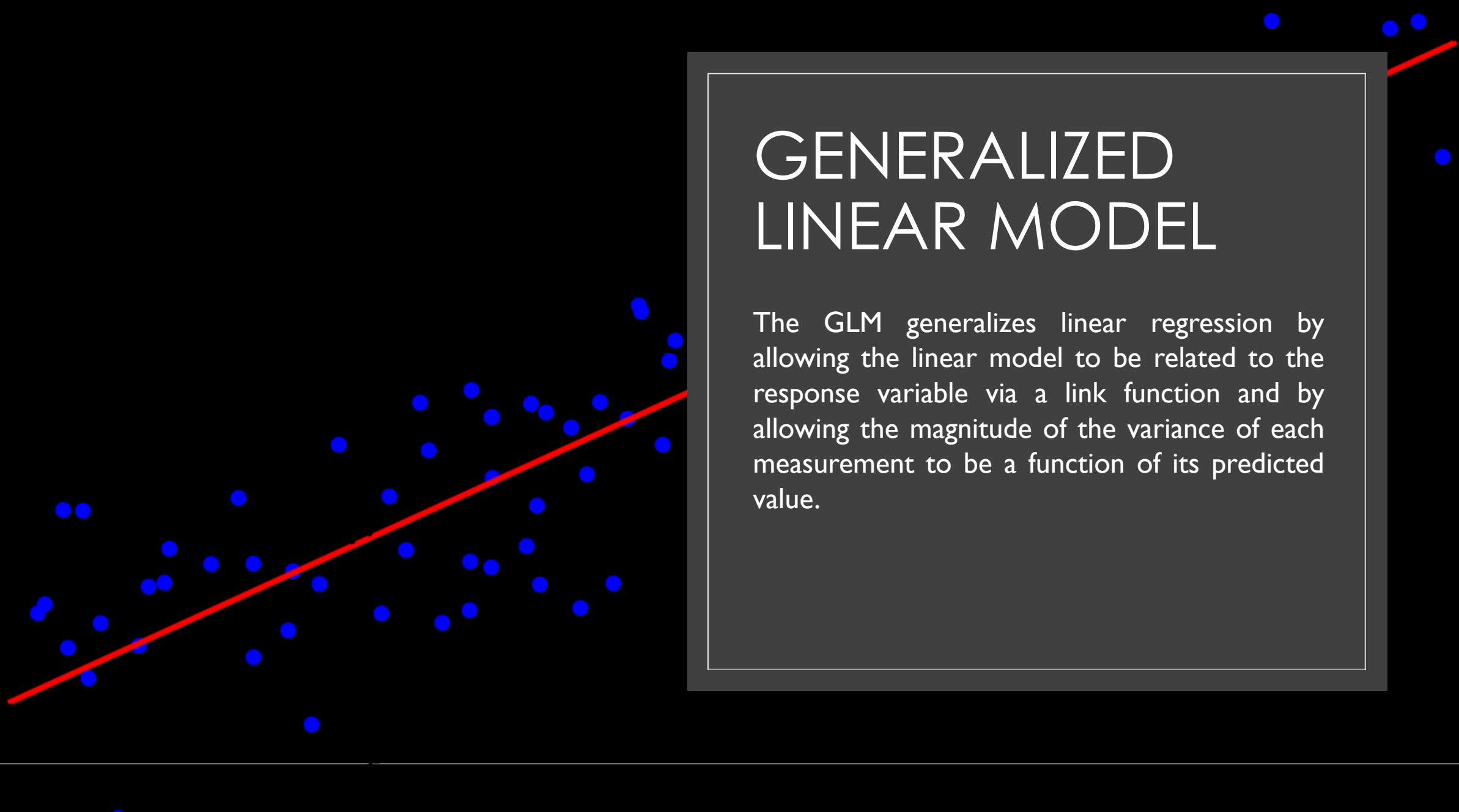
INTRODUCTION
Predictive Modeling

Generally used to feed predictive models:

- Transaction data
- CRM data
- Customer service data
- Survey or polling data
- Digital marketing and advertising data
- Economic data
- Demographic data
- **Machine-generated data (for example, telemetric data or data from sensors)**
- Geographical data
- Web traffic data

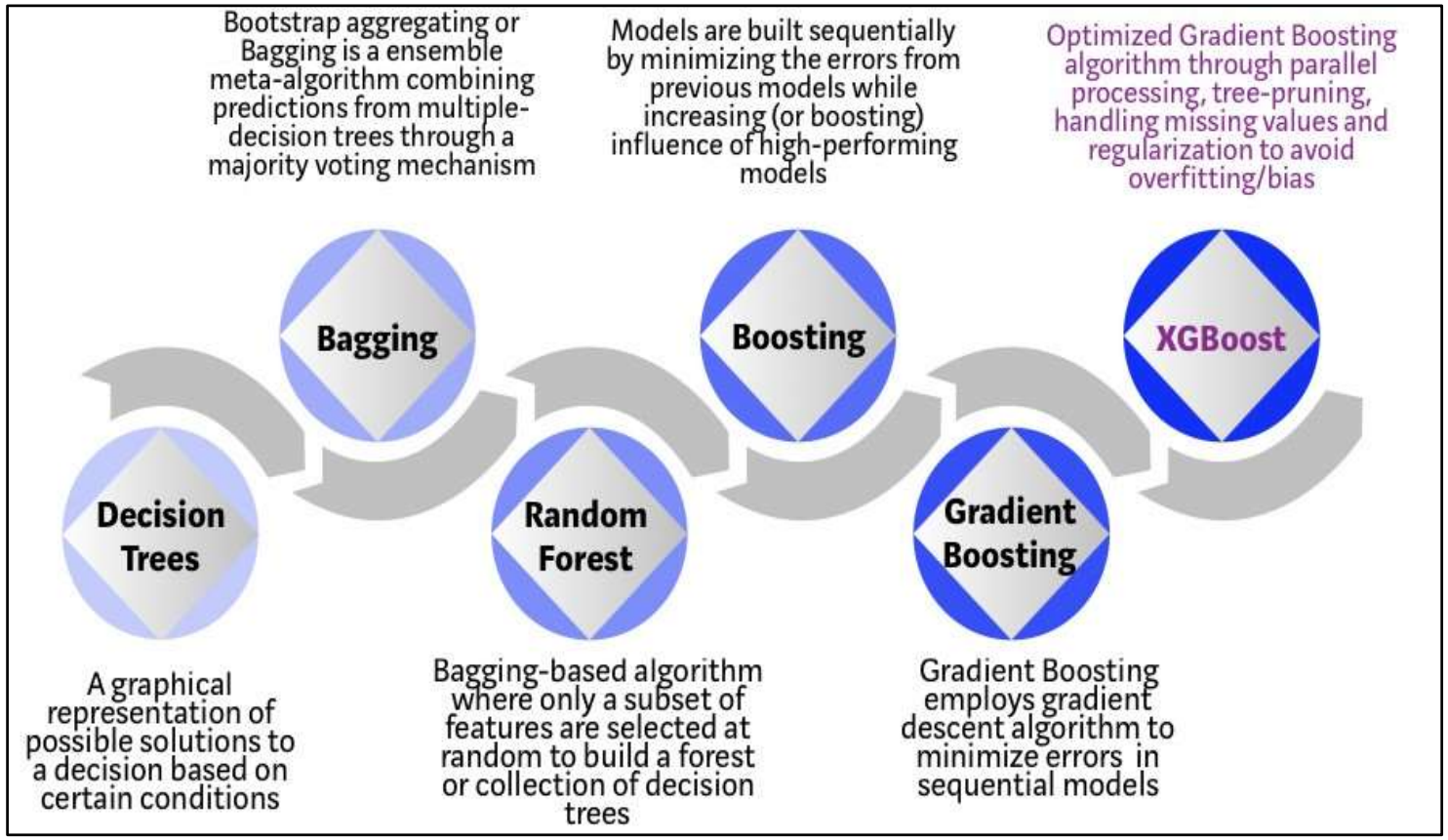
INTRODUCTION
Types of Predictive Models

- Generalized Linear Models (GLM)
- Random Forests
- Decision Trees
- Neural Networks
  - In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks.
- Gradient Boosting Models (GBM)
- Support vector machines (Support Vector Regression-SVR)
- Extended Gradient Boosting Models (XGBoost)
  - Small-to-medium structured /tabular data, decision tree-based algorithms are considered best-in-class.



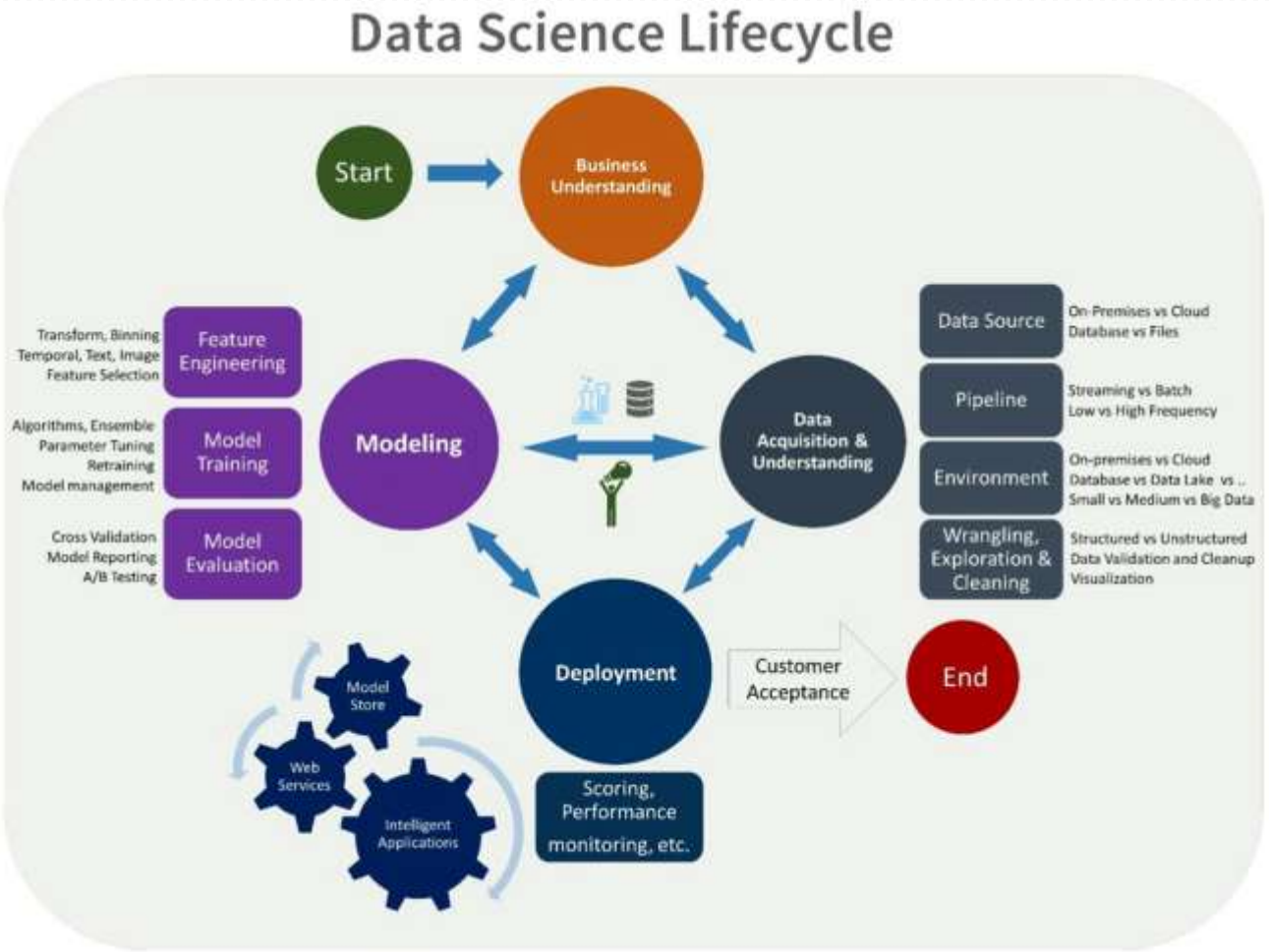
GENERALIZED LINEAR MODEL

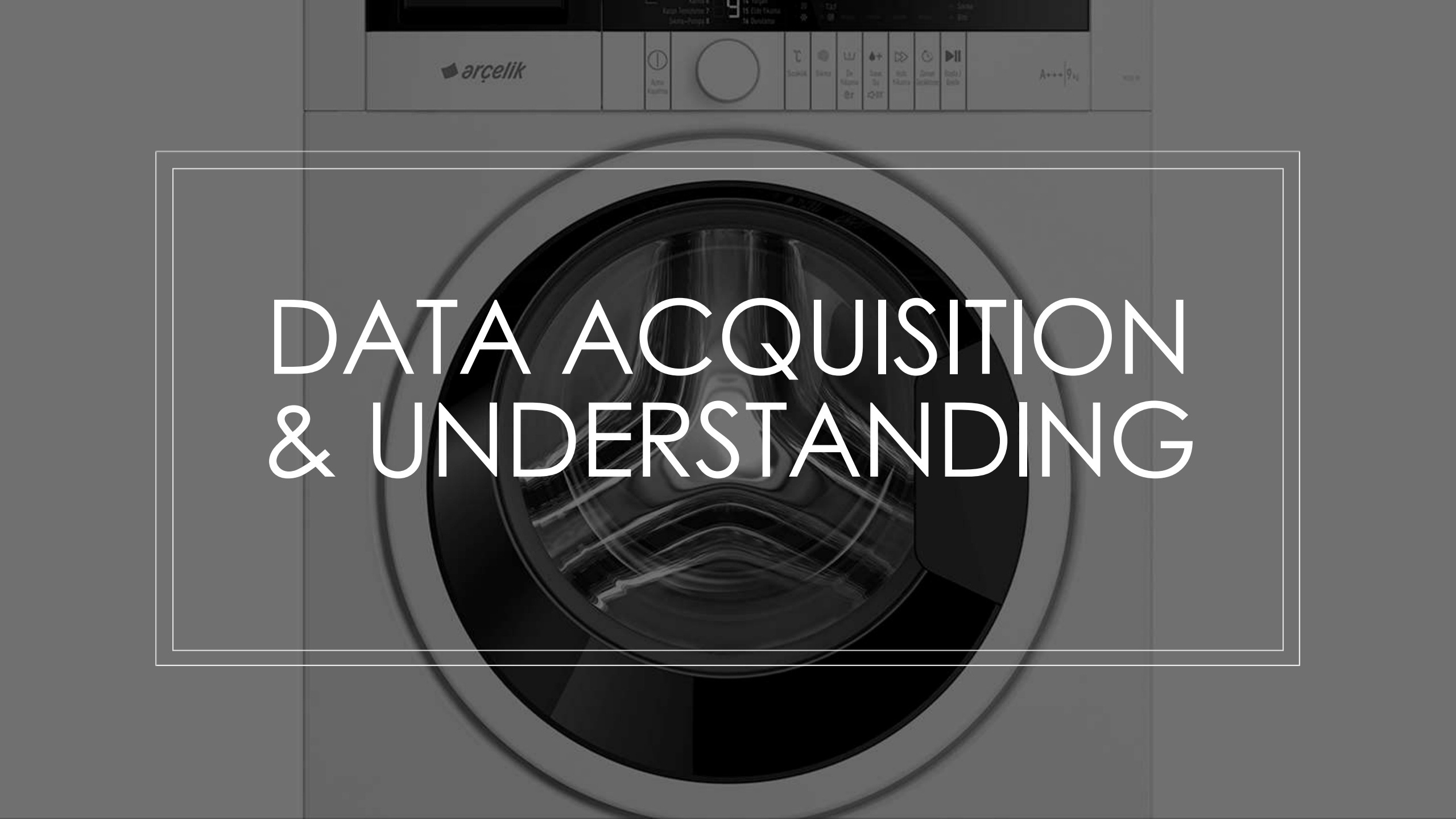
The GLM generalizes linear regression by allowing the linear model to be related to the response variable via a link function and by allowing the magnitude of the variance of each measurement to be a function of its predicted value.



EVOLUTION OF TREE-BASED ALGOS

METHODOLOGY





DATA ACQUISITION & UNDERSTANDING

DATA ACQUISITION & UNDERSTANDING
Washing Machine algorithm flow:

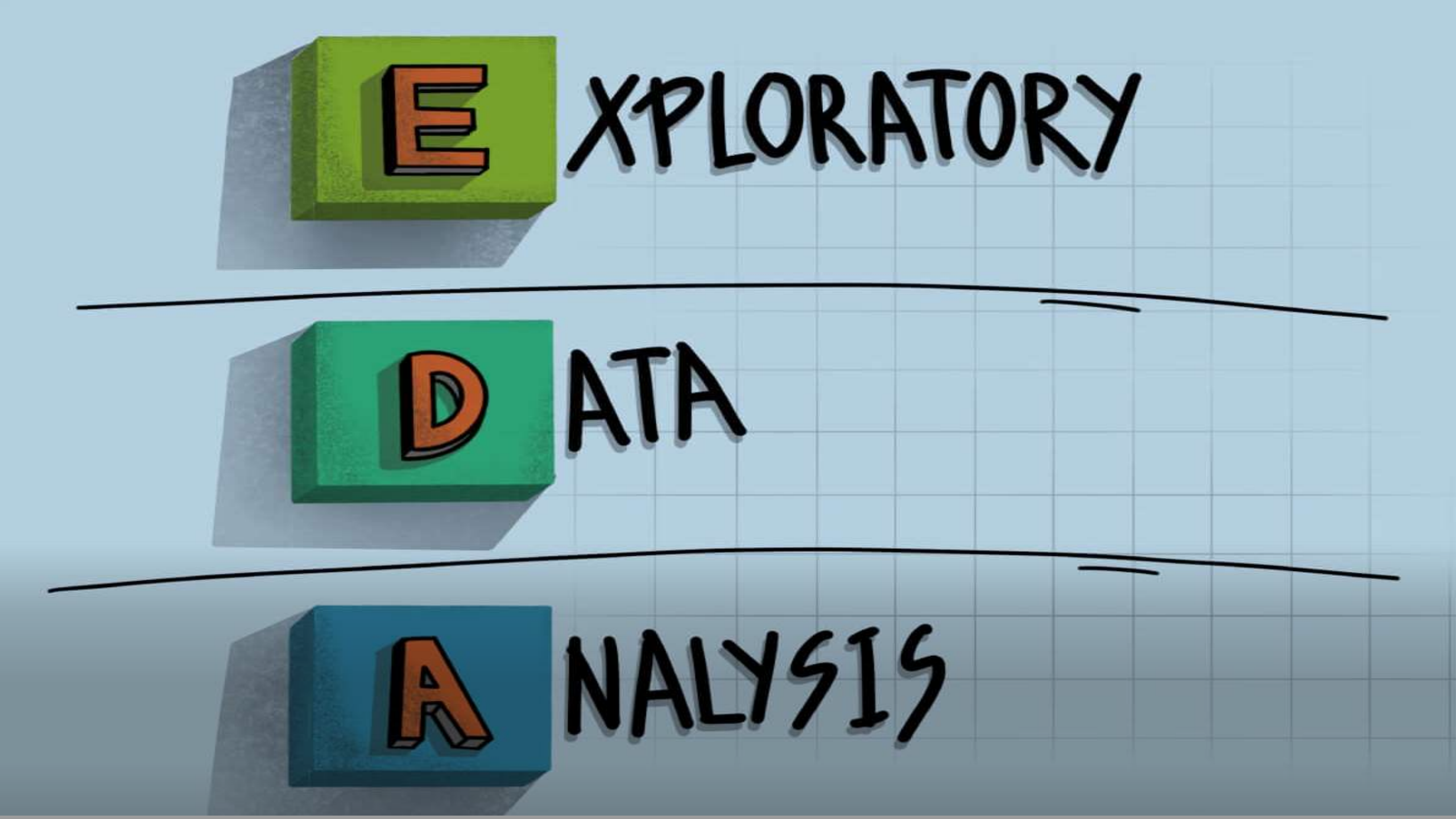
- First, water is taken,
- The laundry is washed with this water for a certain period of time without heating,
- Then the heater turns on,
- After turning on the heater, water is heated,
- Again the laundry is washed with heated water,
- Then, dirty waters are drainaged,
- After the drainage, the 1st rinse water is taken. The laundry is cleaned with detergent and 1st rinse water. And same process is applied again for 2nd and 3rd rinse water,
- Then, after the drainage, process is finished by wring laundry.

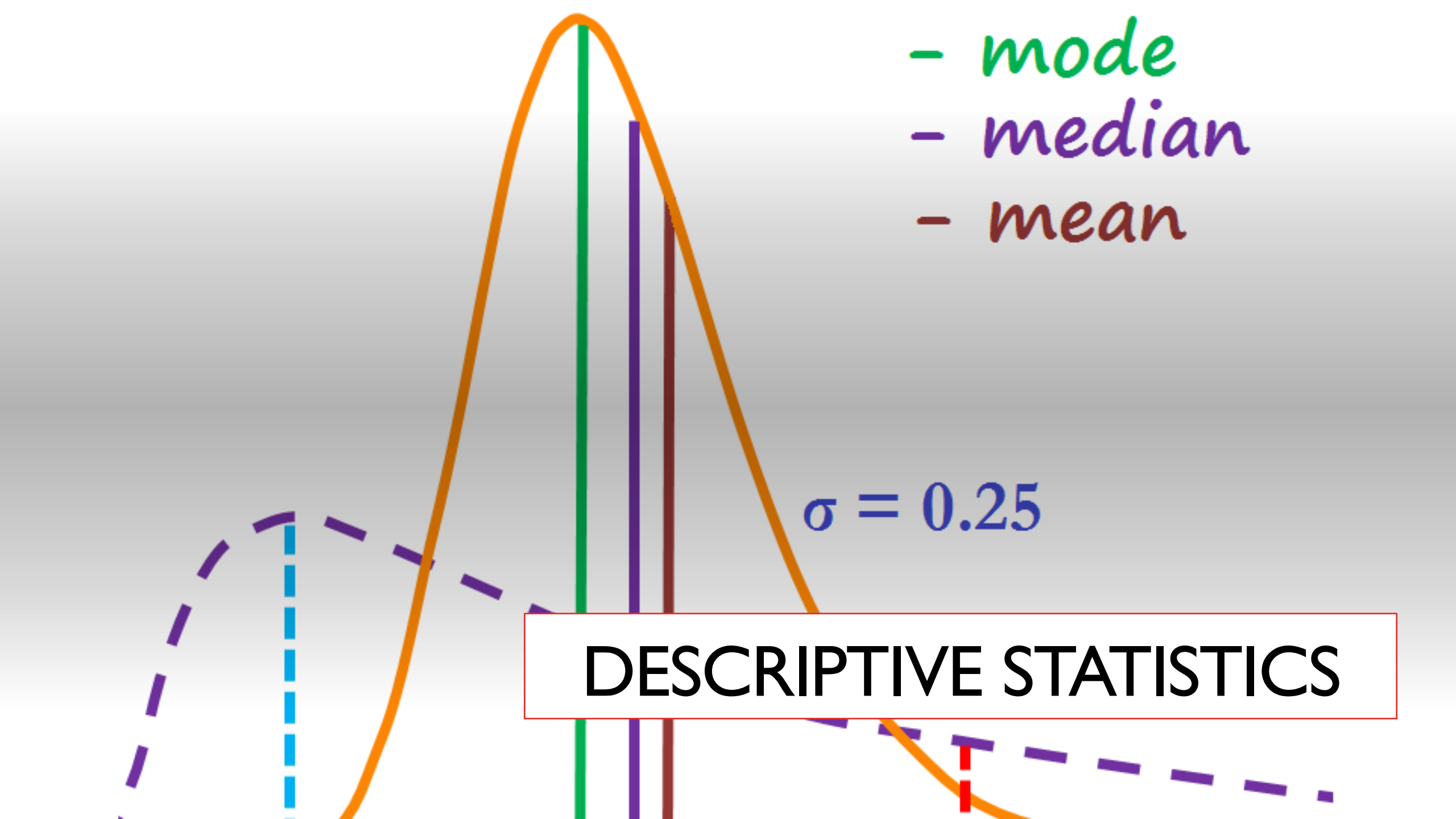
Test_No	# of test washings (1,2,3,4,5,6)
Numune_1_ Relakse_Sonrasi_ En_1	Measured width-1 value of the sample
Numune_1_ Relakse_Sonrasi_ En_2	Measured width-2 value of the sample
Numune_1_ Relakse_Sonrasi_ En_3	Measured width-3 value of the sample
Numune_1_Relakse_Sonrasi_Boy_1	Measured length-1 value of the sample
Numune_1_Relakse_Sonrasi_Boy_2	Measured length-2 value of the sample
Numune_1_Relakse_Sonrasi_Boy_3	Measured length-3 value of the sample
Giris_Alan_Ortalama	Average inlet surface area of the sample
Deterjan_Miktarı(gr)	The amount of detergent used in the algorithm
Test_Kapasitesi(kg)	Weight of sample and filling load (kg)

Su_Sertligi	Water hardness – German hardness (pH value)
Tambur_Hacmi_(lt)	The volume of the washing machine
SY1_Sure_(sn)	Cold Washing Time (in seconds)
SY1_Devir_(rpm)	Cold Washing speed
SY1_ED	Total rotation rate of the engine in cold wash (in %)
SG_Sic_(C)	Entry Temperature at which water enters first to washing machine in cold wash
I_Bas_Sic (C)	Heater start Temperature at the end of the Cold Wash
I_Sure_(sn)	Heating time of the heater (in mins)
I_Devir_(rpm)	Heater speed
I_MHY_ED	Total rotation rate of the engine in heater period (in %)
I_Tset_(C)	Amount of temperature at the end of heating

AY_Su_Mik_(lt)	The amount of water used in the main wash
AY_Sure_(sn)	Main wash time
AY_Devir_(rpm)	Main wash speed
AY_MHY_ED	Total rotation rate of the engine in the main wash (in %)
AY_Tah_(C)	The temperature of the discharged water during the 1st rinse water removal after the main wash is finished
Jet_Pompa	Jet pump system, 1 means machine has, 0 means does not have
Durulama_Sayisi	# of rinses
1D_Su_Mik_(lt)	The amount of water taken in the 1st rinsing
1D_Sure_(sn)	1st rinsing time
1D_Devir_(rpm)	1st rinsing speed
1D_MHY_ED	Total rotation rate of the engine in the 1st rinsing (in %)

2D_Su_Mik_(lt)	The amount of water taken in the 2nd rinsing
2D_Sure_(sn)	2nd rinsing time
2D_Devir_(rpm)	2nd rinsing speed
2D_MHY_ED	Total rotation rate of the engine in the 2nd rinsing (in %)
2D_Sikma_(rpm)	2nd rinsing spin amount
3D_Su_Mik_(lt)	The amount of water taken in the 3rd rinsing
3D_Sure_(sn)	3rd rinsing time
3D_Devir_(rpm)	3rd rinsing speed
3D_MHY_ED	Total rotation rate of the engine in the 3rd rinsing (in %)
Cikis_Alan_Ortalama	Average surface area of the sample at the end of the process


EXPLORATORY
DATA
ANALYSIS



DESCRIPTIVE STATISTICS

df.dtypes

Test_No	float64
Numune_1_ Relakse_Sonrasi_ En_1	float64
Numune_1_ Relakse_Sonrasi_ En_2	float64
Numune_1_ Relakse_Sonrasi_ En_3	float64
Numune_1_Relakse_Sonrasi_Boy_1	float64
Numune_1_Relakse_Sonrasi_Boy_2	float64
Numune_1_Relakse_Sonrasi_Boy_3	float64
Giris_Alan_Ortalama	float64
Deterjan_Miktarı(gr)	float64
Test_Kapasitesi(kg)	float64
Su_Sertligi	float64
Tambur_Hacmi_(lt)	float64
SY1_Sure_(sn)	float64
SY1_Devir_(rpm)	float64
SY1_ED	float64
SG_Sic_(C)	float64
1_Bas_Sic (C)	float64
1_Sure_(sn)	float64
1_Devir_(rpm)	float64
1_MHY_ED	float64
1_Tset_(C)	float64

AY_Su_Mik_(lt)	float64
AY_Sure_(sn)	float64
AY_Devir_(rpm)	float64
AY_MHY_ED	float64
AY_Tah_(C)	float64
Jet_Pompa	float64
Durulama_Sayisi	float64
1D_Su_Mik_(lt)	float64
1D_Sure_(sn)	float64
1D_Devir_(rpm)	float64
1D_MHY_ED	float64
2D_Su_Mik_(lt)	float64
2D_Sure_(sn)	float64
2D_Devir_(rpm)	float64
2D_MHY_ED	float64
2D_Sikma_(rpm)	float64
3D_Su_Mik_(lt)	float64
3D_Sure_(sn)	float64
3D_Devir_(rpm)	float64
3D_MHY_ED	float64
Cikis_Alan_Ortalama	float64

NO MISSING VALUE

```
In [52]: df.isnull()
Out[52]:
     Test_No   Numune_1_ Relakse_Sonrasi_ En_1  ...  3D_MHY_ED  Cikis_Alan_Ortalama
0       False                          False  ...      False                False
1       False                          False  ...      False                False
2       False                          False  ...      False                False
3       False                          False  ...      False                False
4       False                          False  ...      False                False
..        ...                            ...  ...        ...                  ...
175     False                          False  ...      False                False
176     False                          False  ...      False                False
177     False                          False  ...      False                False
178     False                          False  ...      False                False
179     False                          False  ...      False                False

[180 rows x 42 columns]
```

NO DUPLICATE ROW

```
In [53]: duplicate_rows_df = df[df.duplicated()]

In [54]: print('number of duplicate rows:', duplicate_rows_df.shape)
number of duplicate rows: (0, 42)
```

	Test_No	En_1	En_2	En_3	Boy_1	Boy_2	Boy_3	GAO	Deterjan_Miktarı(gr)	Test_Kapasitesi (kg)	Su_Sertligi	Tambur_Hacmi_(lt)	SY1_Sure_(sn)	SY1_Devir_(rpm)	SY1_ED	SG_Sic_(C)	I_Bas_Sic (C)	I_Sure_(sn)	I_Devir_(rpm)	I_MHY_ED	I_Tset_(C)	AY_Su_Mik_(lt)
count	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180
mean	3,5	273,05	272,4166667	272,9	267,5	267,3666667	267,4833333	72953,2907	56,27333333	1,866666667	6,2	65,36	382,4111111	28,33333333	0,040266667	15,18066667	16,19766667	456,0111111	24,33333333	0,824666667	34,98666667	12,63722222
std	1,712588945	1,694634978	2,018420756	1,814282258	5,000558628	4,722938201	4,688301142	1170,208449	3,647446493	0,34088285	3,496846145	0,681896796	327,8284991	7,695315269	0,018103859	0,500015195	6,437890811	232,9864644	12,26340334	1,607392017	8,339758581	0,55238038
min	1	271	270	270,5	248,5	250	250	68986,75	47	1	3	64	176,9	0	0	12,98	0	0	0	0	15,98	11,48
25%	2	272	271	271,5	268	268	268	72807,778	57,7	2	5	65,7	180	30	0,029	14,83	17,57	300	30	0,011	30,94	12,32
50%	3,5	272,75	271,75	272,5	268,75	268,25	268,75	73144	57,7	2	5	65,7	297	30	0,036	15,23	18,265	477	30	0,058	37,47	12,685
75%	5	275	273,5	273,5	269,5	269	269	73485,833	57,7	2	5	65,7	357	30	0,058	15,51	19,32	636	30	0,058	41,77	12,8
max	6	276,5	277	277	271	270	270,5	74292,167	57,7	2	15	65,7	1480	35	0,058	16,2	21	810	35	4,76	46,26	14,44
max=min	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil
max-min	5	5,5	7	6,5	22,5	20	20,5	5305,417	10,7	1	12	1,7	1303,1	35	0,058	3,22	21	810	35	4,76	30,28	2,96
median-mean	0	-0,3	-0,666666667	-0,4	1,25	0,883333333	1,266666667	190,7093	1,426666667	0,133333333	-1,2	0,34	-85,41111111	1,666666667	-0,004266667	0,049333333	2,067333333	20,98888889	5,666666667	-0,766666667	2,483333333	0,047777778
SKEWNESS	0	0,613060857	0,660676599	0,83572226	-3,260261705	-3,329913682	-3,327572638	-2,394629595	-2,175448351	-2,175448351	2,076044927	-1,512634581	2,603931062	-3,330688423	-0,713845603	-0,867887424	-2,082686198	-0,618799546	-1,470632541	1,602676423	-0,945102159	0,696366716
KURTOSIS	-1,270438828	-0,783263545	-0,712381127	-0,230475788	9,401776493	9,642285988	9,641744814	6,035844615	2,763155003	2,763155003	2,570052941	0,291174697	5,886342993	9,73013713	-0,439179542	2,635950529	2,54767411	-0,392994428	0,247312818	0,739203732	-0,045574969	2,56451849

	AY_Sure_(sn)	AY_Devir_(rpm)	AY_MHY_ED	AY_Tah_(C)	Jet_Pompa	Durulama_Sayisi	1D_Su_Mik_(lt)	1D_Sure_(sn)	1D_Devir_(rpm)	1D_MHY_ED	2D_Su_Mik_(lt)	2D_Sure_(sn)	2D_Devir_(rpm)	2D_MHY_ED	2D_Sikma_(rpm)	3D_Su_Mik_(lt)	3D_Sure_(sn)	3D_Devir_(rpm)	3D_MHY_ED	CAO
count	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180
mean	1187,833333	28,33333333	0,038666667	29,78288889	0,866666667	3	11,24555556	264,4333333	28,33333333	0,039466667	11,29544444	255,9555556	28,33333333	1,426133333	81,57777778	13,99811111	222,85	28,33333333	0,039533333	67735,04257
std	535,8431825	7,695315269	0,019429862	5,759223304	0,34088285	0	0,635216254	68,39681295	7,695315269	0,018863343	0,675129056	58,98696213	7,695315269	1,78588724	208,5725884	0,53650617	40,8537636	7,695315269	0,018759995	3541,243852
min	438	0	0	17,85	0	3	9,82	160	0	0	9,82	160	0	0	0	13,26	160	0	0	55492,5
25%	612	30	0,024	26	1	3	10,83	179	30	0,024	10,87	180	30	0,058	0	13,66	180	30	0,024	65556,12475
50%	955	30	0,036	31,585	1	3	11,595	295	30	0,036	11,62	295	30	0,058	0	13,78	235	30	0,036	68284
75%	1808	30	0,058	33,58	1	3	11,72	300	30	0,058	11,74	299	30	3,61	0	14,58	240	30	0,058	70600,49975
max	1918	35	0,058	41,28	1	3	11,84	366	35	0,058	12,71	304	35	4,76	620	15,91	352	35	0,058	73840,5
max=min	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil	eşit değil
max-min	1480	35	0,058	23,43	1	0	2,02	206	35	0,058	2,89	144	35	4,76	620	2,65	192	35	0,058	18348
median-mean	-232,8333333	1,666666667	-0,002666667	1,802111111	0,133333333	0	0,349444444	30,56666667	1,666666667	-0,003466667	0,324555556	39,04444444	1,666666667	-1,368133333	-81,57777778	-0,218111111	12,15	1,666666667	-0,003533333	548,9574333
SKEWNESS	0,102343503	-3,330688423	-0,53507349	-0,588976304	-2,175448351	0	-1,185669638	-0,334065516	-3,330688423	-0,626218518	-0,990964029	-0,726616709	-3,330688423	0,653548692	-2,175836646	-1,025273445	0,386394596	-3,330688423	-0,621556795	-0,771024791
KURTOSIS	-1,619877749	9,73013713	-0,994405967	-0,337445911	2,763155003	0	-0,142578778	-1,263010795	9,73013713	-0,771316786	-0,078394437	-1,453543148	9,73013713	-1,363625826	2,76564041	-0,354449722	-0,266542043	9,73013713	-0,766140788	-0,308871767

Interpretation for descriptive statistics

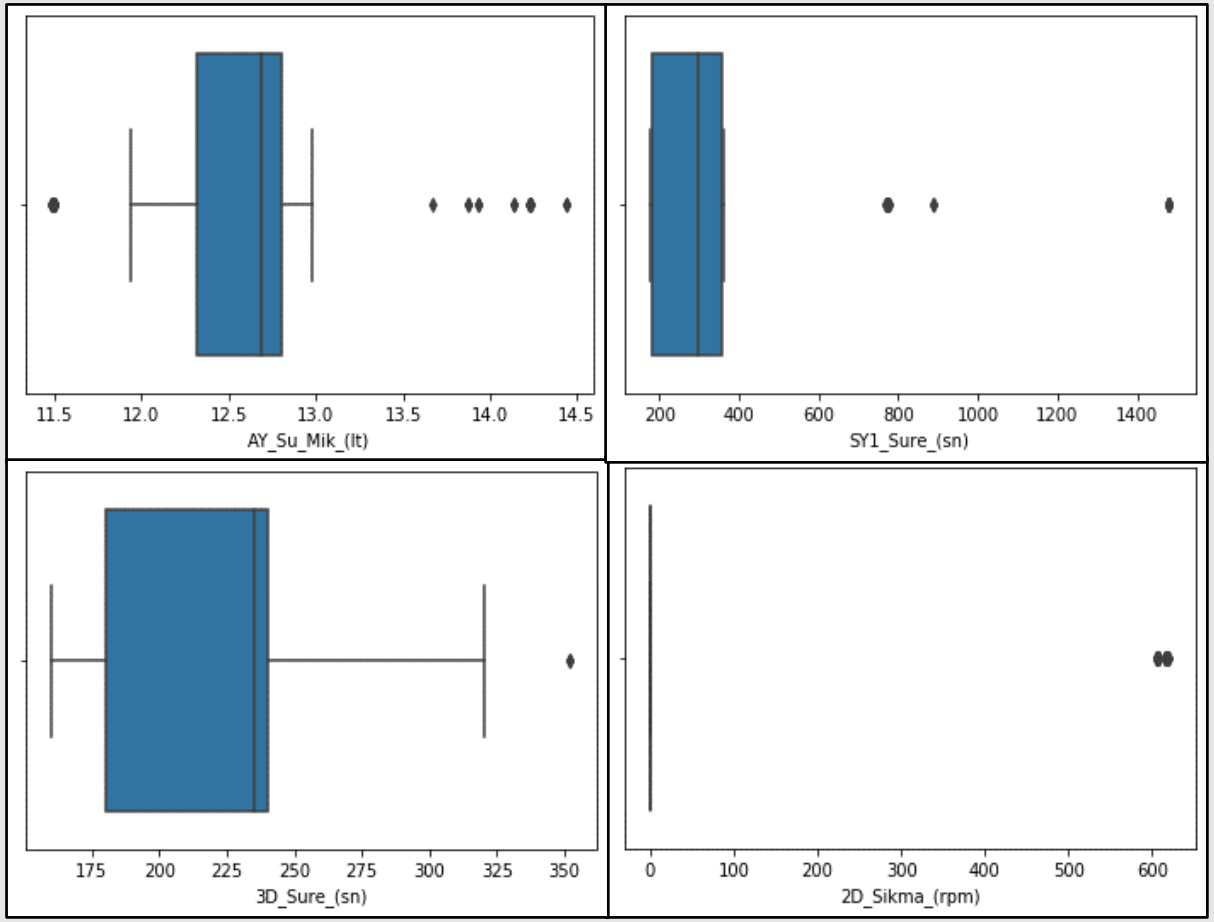
- **Count:** This is important for the first perception about the volume of «**missing data**».
  - Therefore, there are no missing values for all features.
- **Min and max values:** This can give an idea about the range of values and is helpful to detect outliers.
  - «Durulama Sayısı» has the same min and max values, which means this feature has to be dropped from the data frame. Bcz it causes error during the calculation of the standard deviation.
  - Furthermore, the other exception could eventually be the max values of «SY1_Sure_(sn)», «3D_Sure_(sn)» and «2D_Sikma_(rpm)».
- **Median and mean:** The proximity of the mean and median values indicate proximity to the normal distribution.

!!! If median equals mean (or is very close), the distribution can be considered as normally distributed!!!

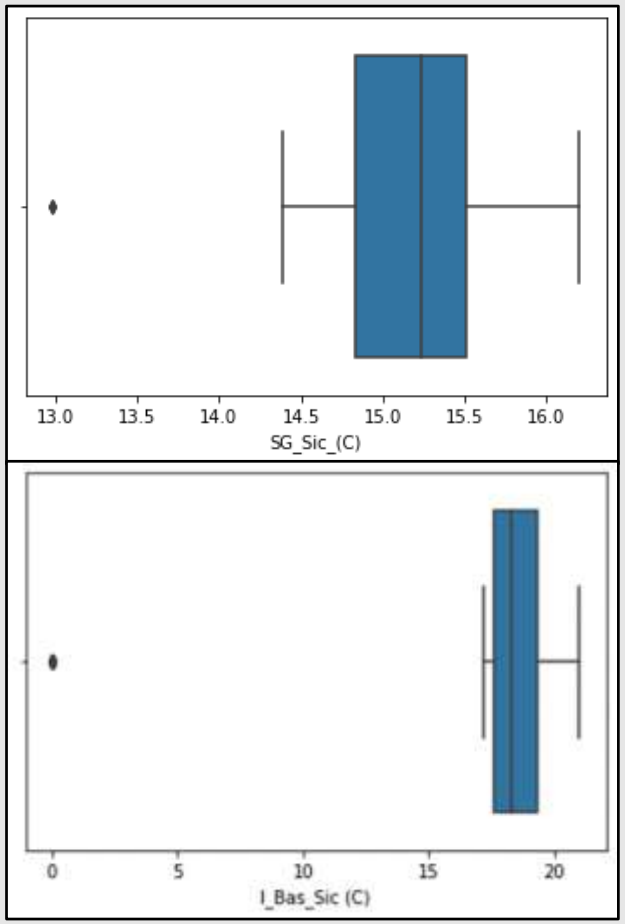
- **Mean and standard deviation:** «mean»; the central tendency of the distribution, «standard deviation», quantifies its amount of variation.

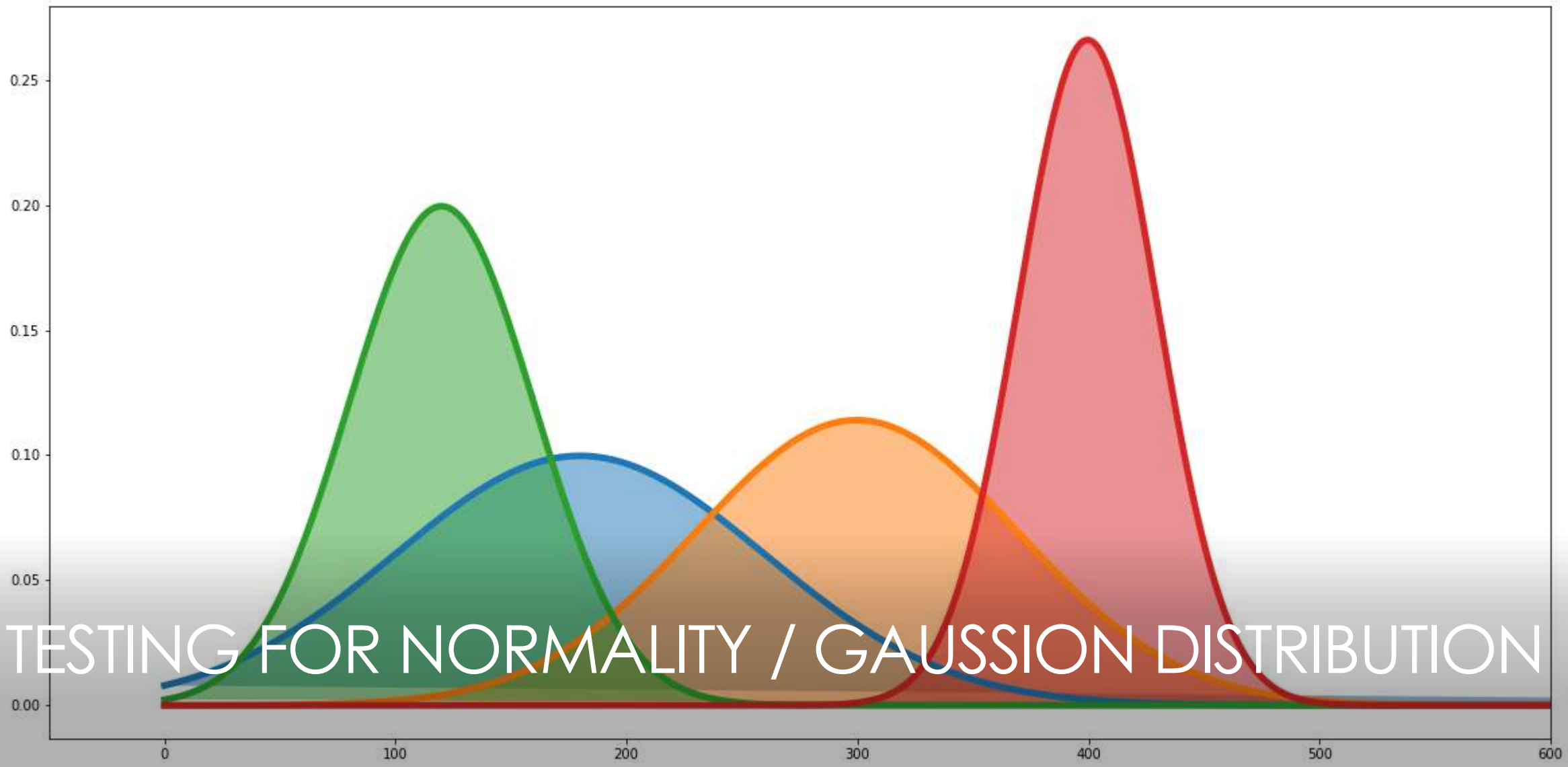
!!! A low standard deviation suggests that data points tend to be close to the mean!!!

Example of variables that have high max values

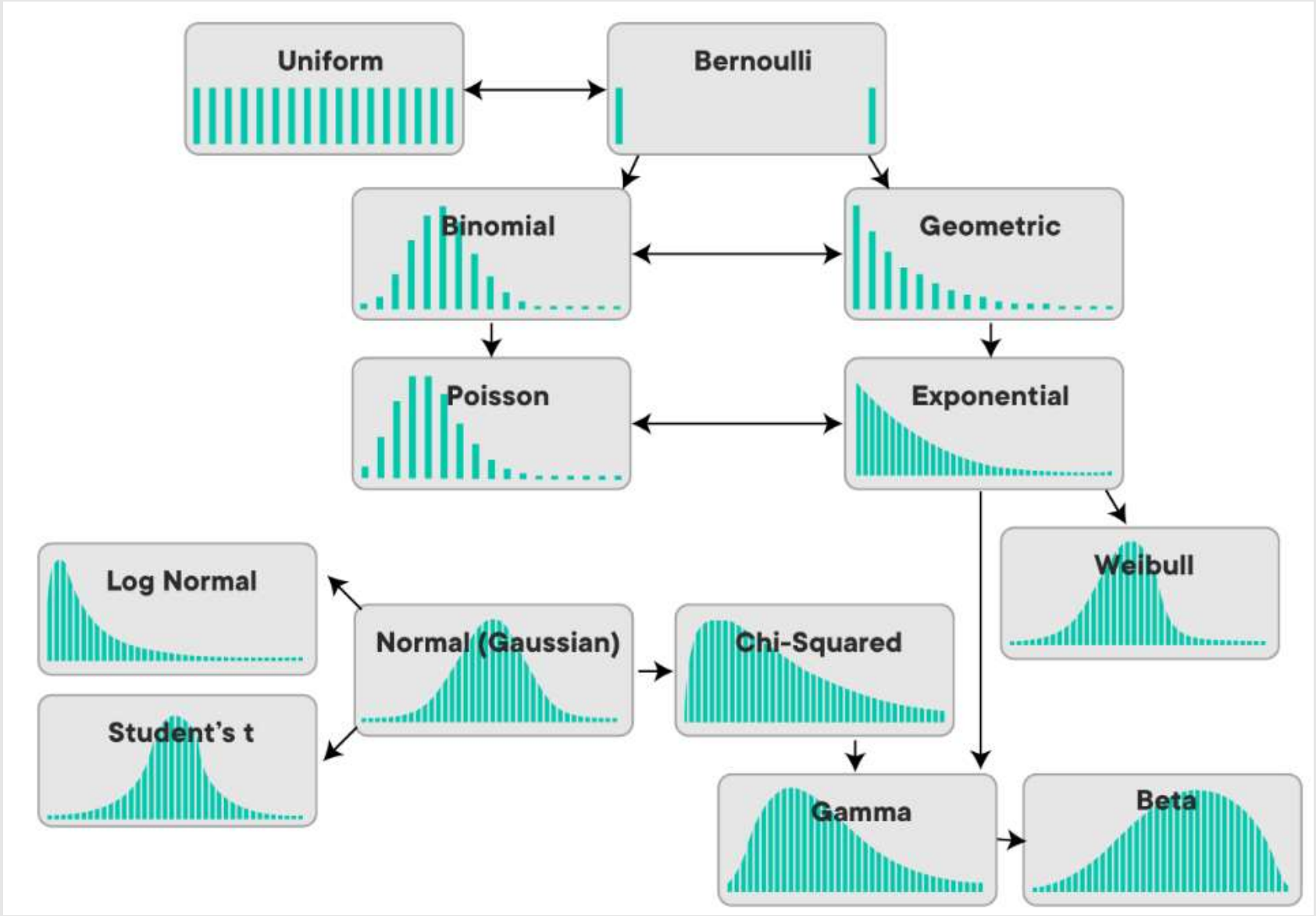


Example of min outlier value and zero outlier that caused by non-measurable value

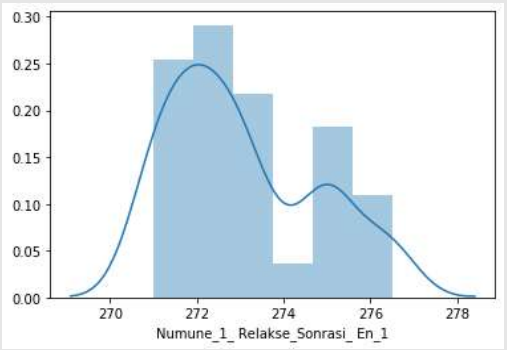
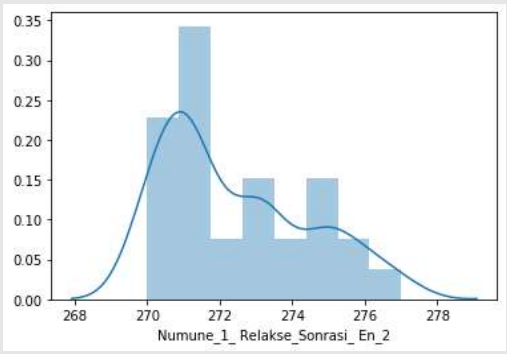
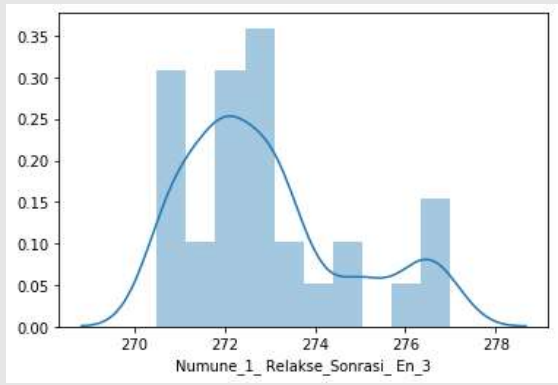
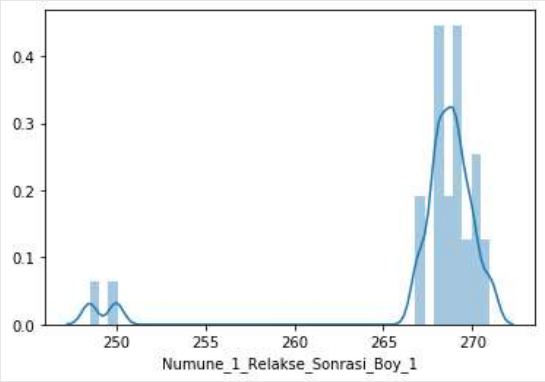
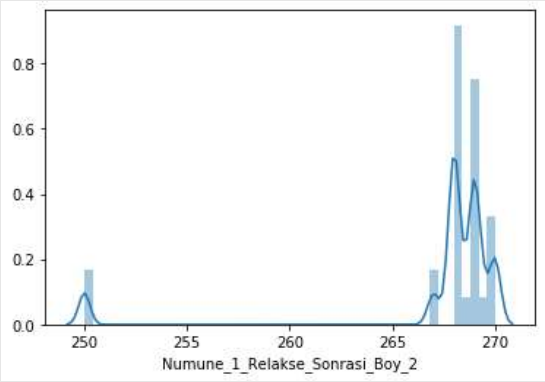
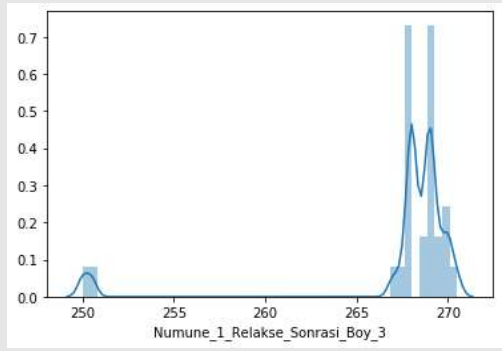




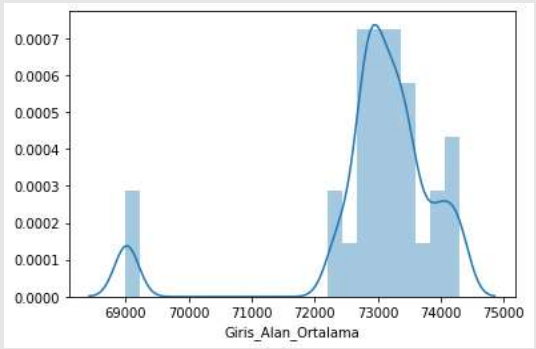
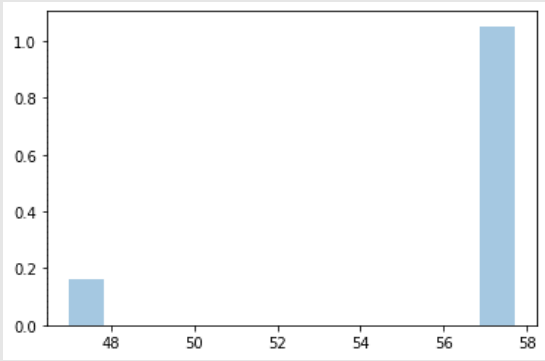
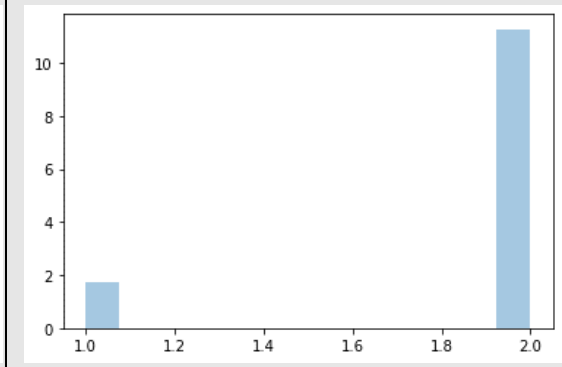
TESTING FOR NORMALITY / GAUSSION DISTRIBUTION

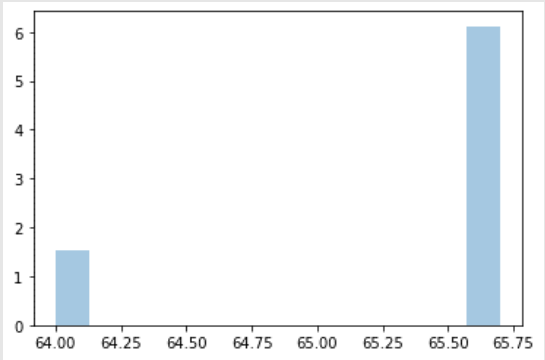
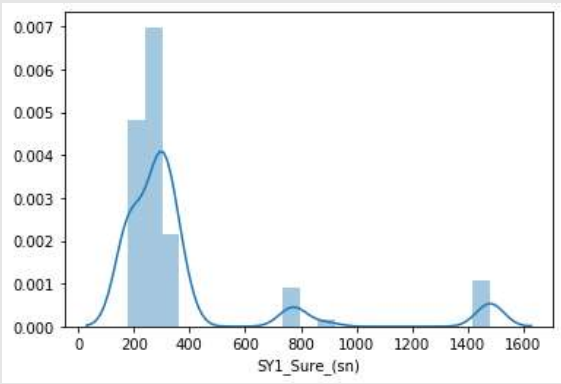
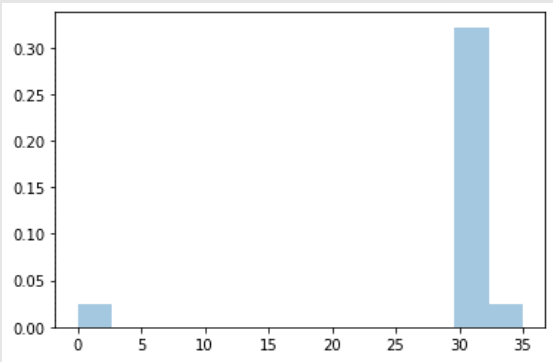

Uniform
Bernoulli
Binomial
Geometric
Poisson
Exponential
Weibull
Log Normal
Normal (Gaussian)
Chi-Squared
Student's t
Gamma
Beta

Visualization data with Histogram whether model variables are normal distributed or not

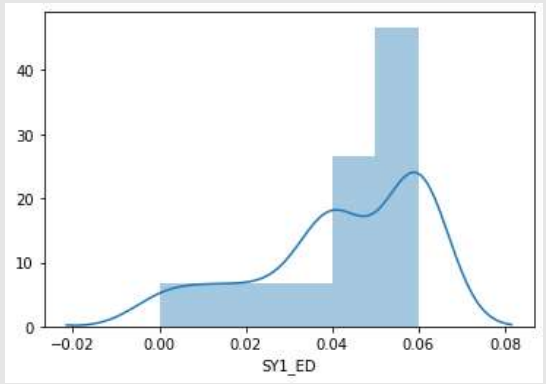
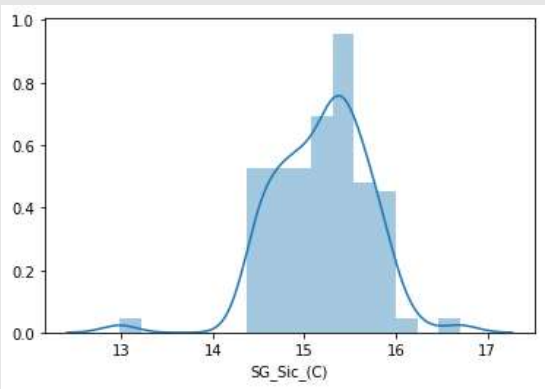
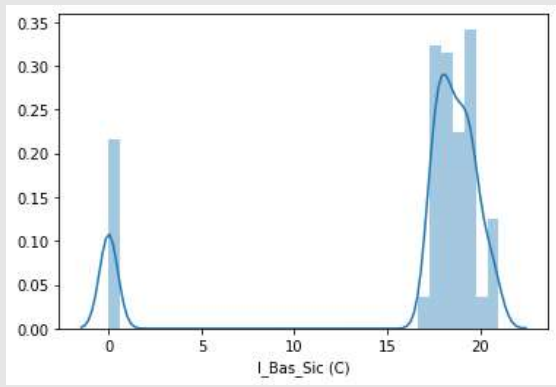
	Numune_1_ Relakse_Sonrasi_ En_1	Numune_1_ Relakse_Sonrasi_ En_2	Numune_1_ Relakse_Sonrasi_ En_3
			
mean	273,05	272,4166667	272,9
50%	272,75	271,75	272,5
	Numune_1_Relakse_Sonrasi_Boy_1	Numune_1_Relakse_Sonrasi_Boy_2	Numune_1_Relakse_Sonrasi_Boy_3
			
mean	267,5	267,3666667	267,4833333
50%	268,75	268,25	268,75

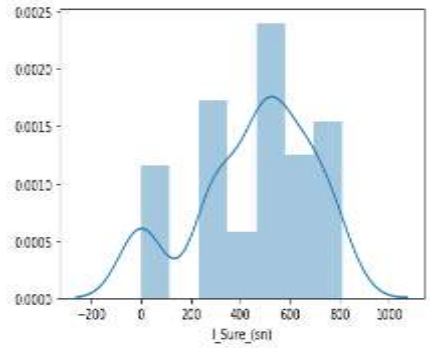
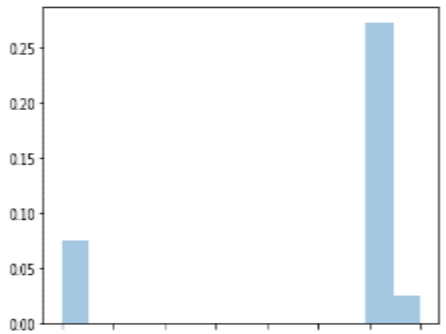
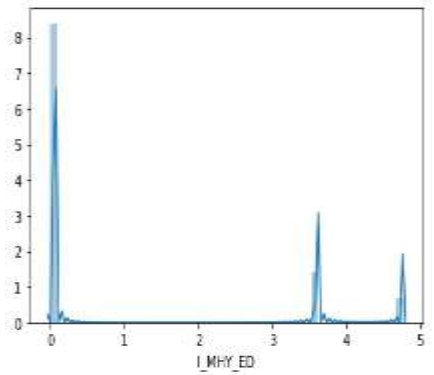
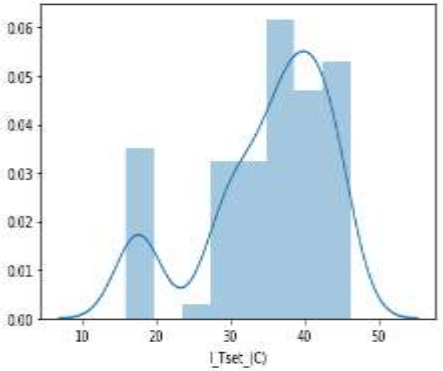
Visualization data with Histogram whether model variables are normal distributed or not

	Giris_Alan_Ortalama	Deterjan_Miktarı(gr)	Test_Kapasitesi(kg)
			
mean	72953,29033	56,27333333	1,866666667
50%	73144	57,7	2

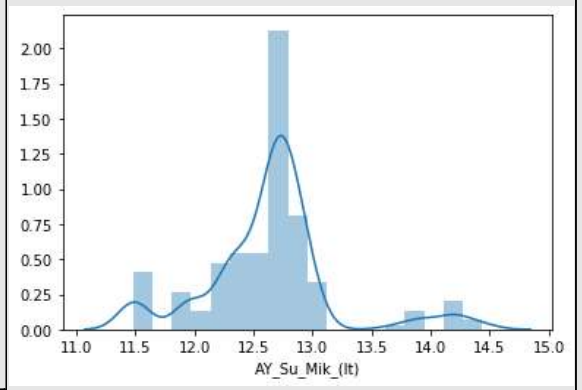
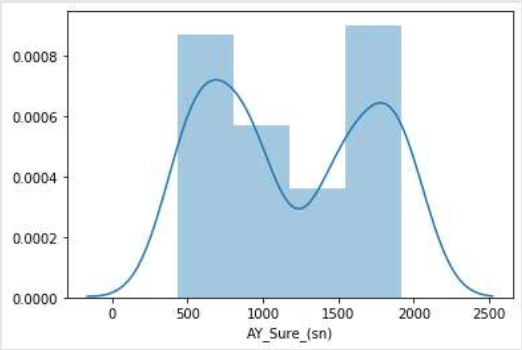
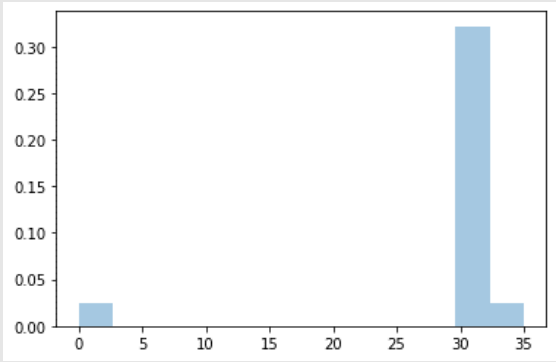
	Tambur_Hacmi_(lt)	SY1_Sure_(sn)	SY1_Devir_(rpm)
			
mean	65,36	382,4111111	28,33333333
50%	65,7	297	30

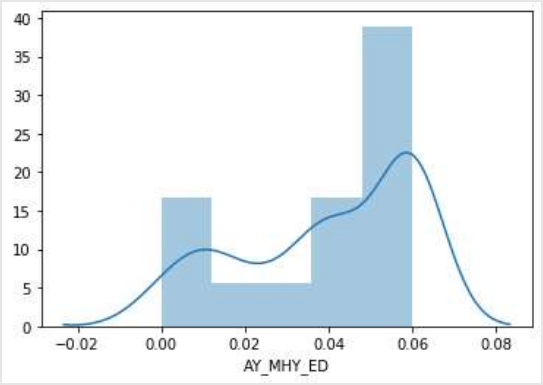
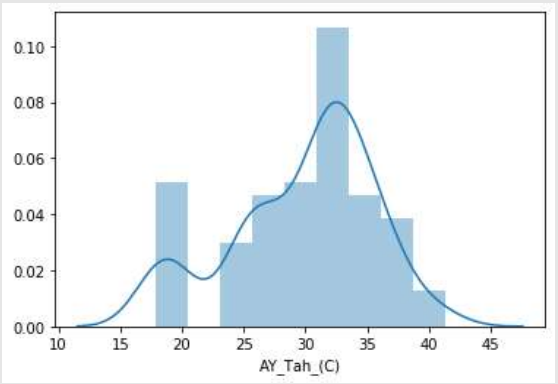
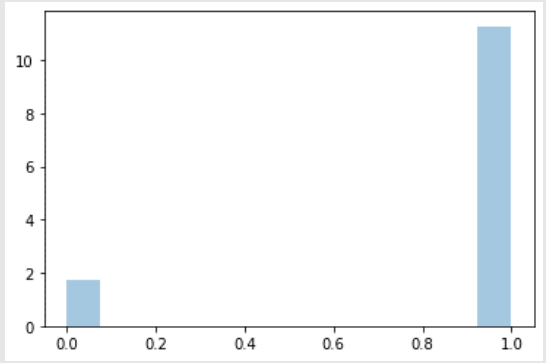
Visualization data with Histogram whether model variables are normal distributed or not

	SY1_ED	SG_Sic_(C)	I_Bas_Sic ©
			
mean	0,042	15,19066667	16,22316667
50%	0,04	15,24	18,3

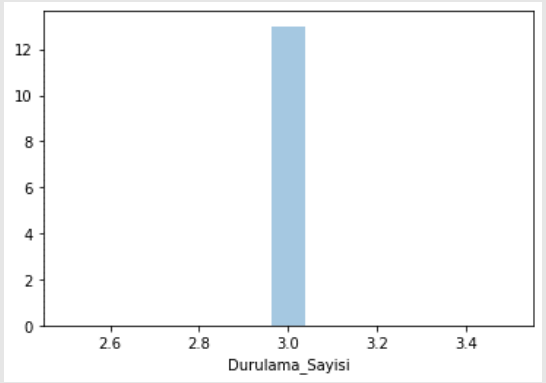
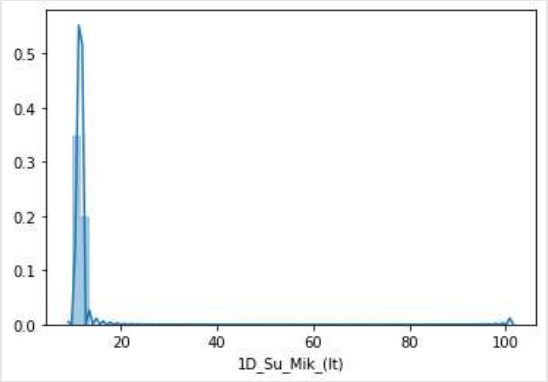
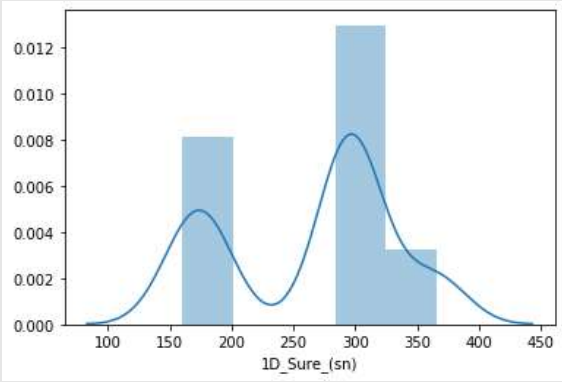
	I_Sure_(sn)	I_Devir_(rpm)	I_MHY_ED	I_Tset_(C)
				
mean	456,0111111	24,33333333	0,825333333	35,04066667
50%	477	30	0,06	37,71

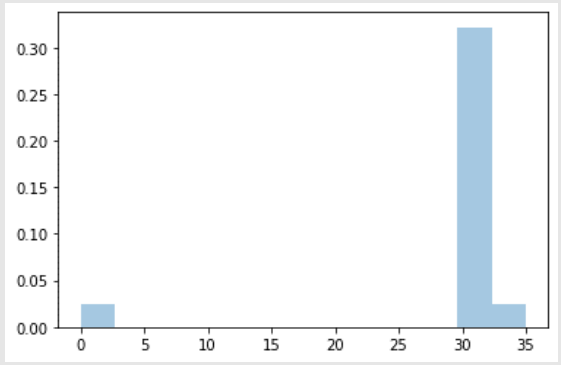
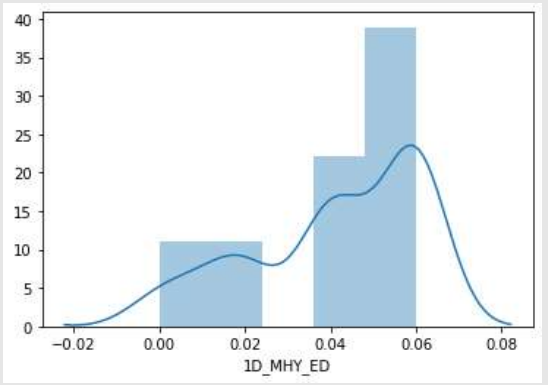
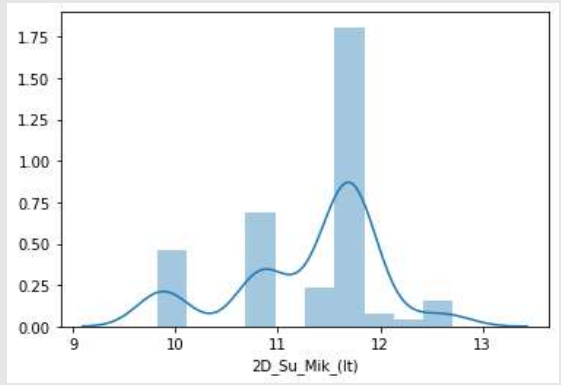
Visualization data with Histogram whether model variables are normal distributed or not

	AY_Su_Mik_(lt)	AY_Sure_(sn)	AY_Devir_(rpm)
			
mean	12,64277778	1187,833333	28,33333333
50%	12,69	955	30

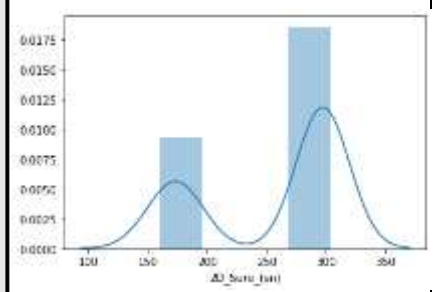
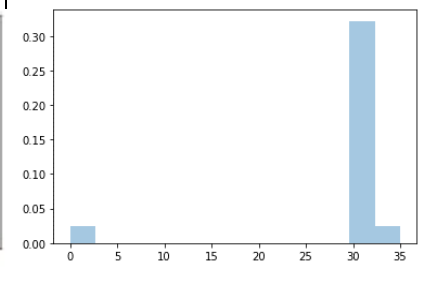
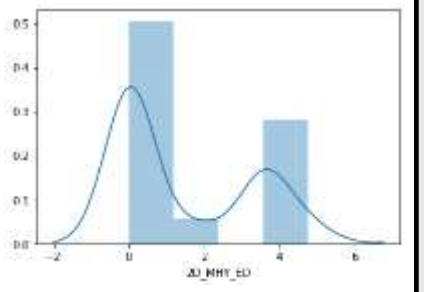
	AY_MHY_ED	AY_Tah_(C)	Jet_Pompa
			
mean	0,04	29,81788889	0,866666667
50%	0,04	31,585	1

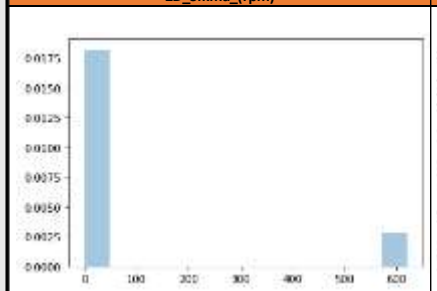
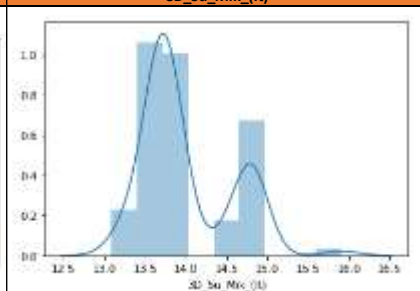
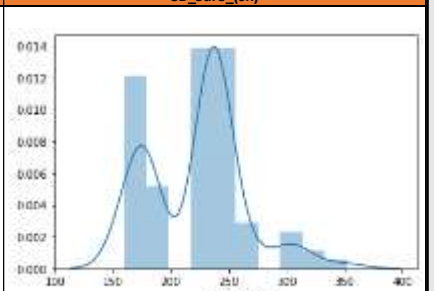
Visualization data with Histogram whether model variables are normal distributed or not

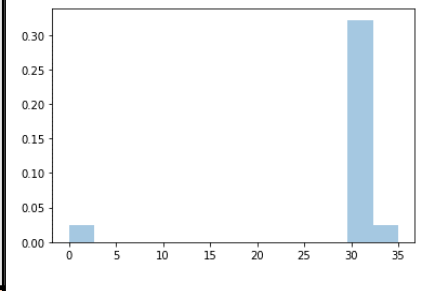
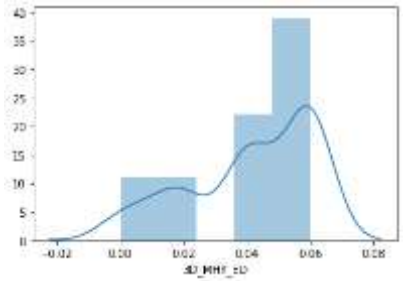
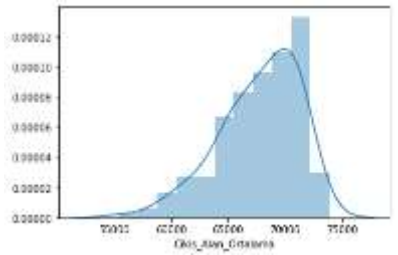
	Durulama_Sayisi	1D_Su_Mik_(lt)	1D_Sure_(sn)
			
mean	3	11,24888889	264,4333333
50%	3	11,6	295

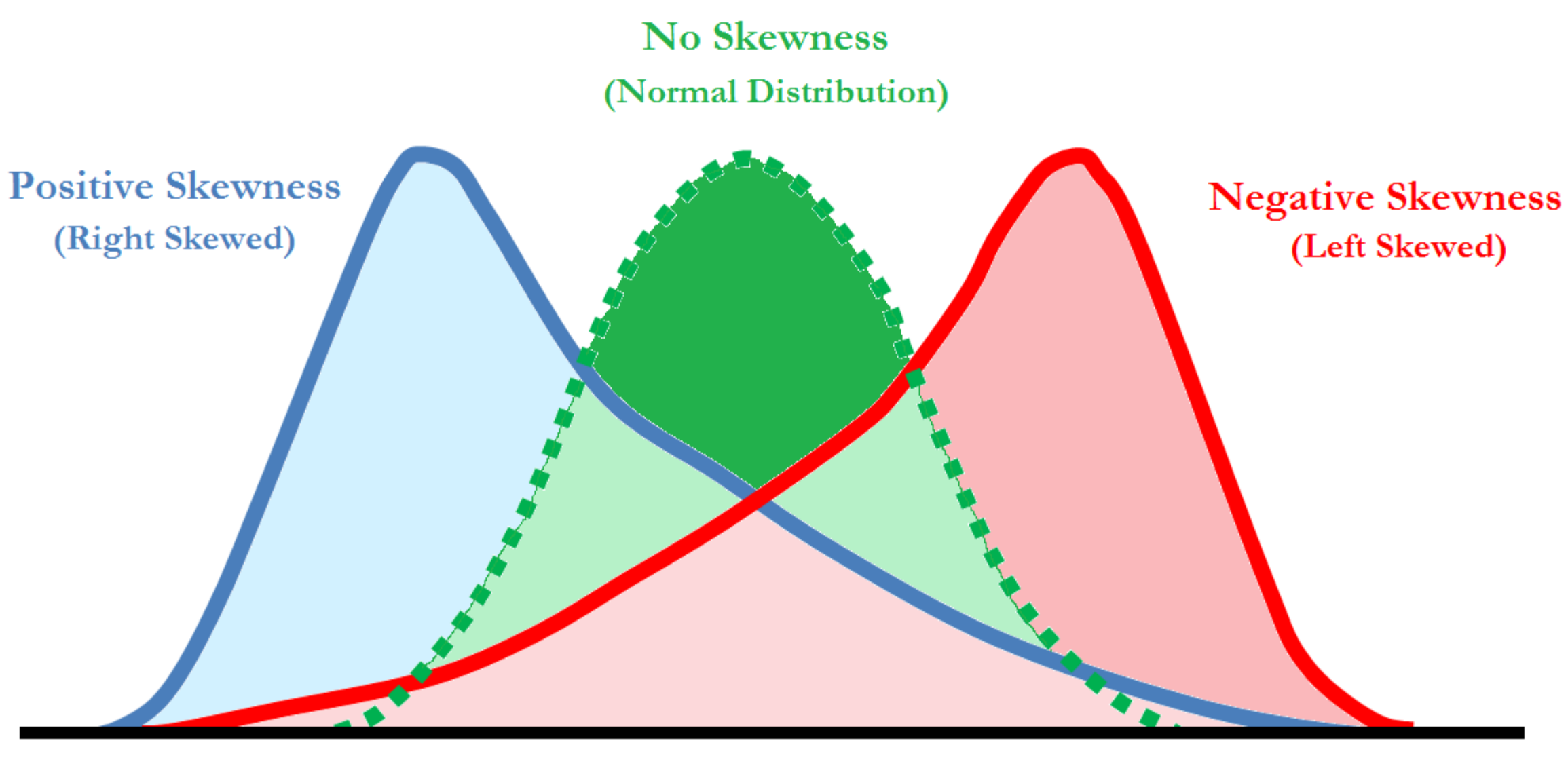
	1D_Devir_(rpm)	1D_MHY_ED	2D_Su_Mik_(lt)
			
mean	28,33333333	0,041333333	11,31644444
50%	30	0,04	11,62

Visualization data with Histogram whether model variables are normal distributed or not

	2D_Sure_(sn)	2D_Devir_(rpm)	2D_MHY_ED
			
mean	255,9555556	28,33333333	1,426666667
50%	295	30	0,06

	2D_Sikma_(rpm)	3D_Su_Mik_(lt)	3D_Sure_(sn)
			
mean	81,57777778	13,99411111	222,85
50%	0	13,78	235

	3D_Devir_(rpm)	3D_MHY_ED	Cikis_Alan_Ortalama
			
mean	28,33333333	0,041333333	67735,04272
50%	30	0,04	68284



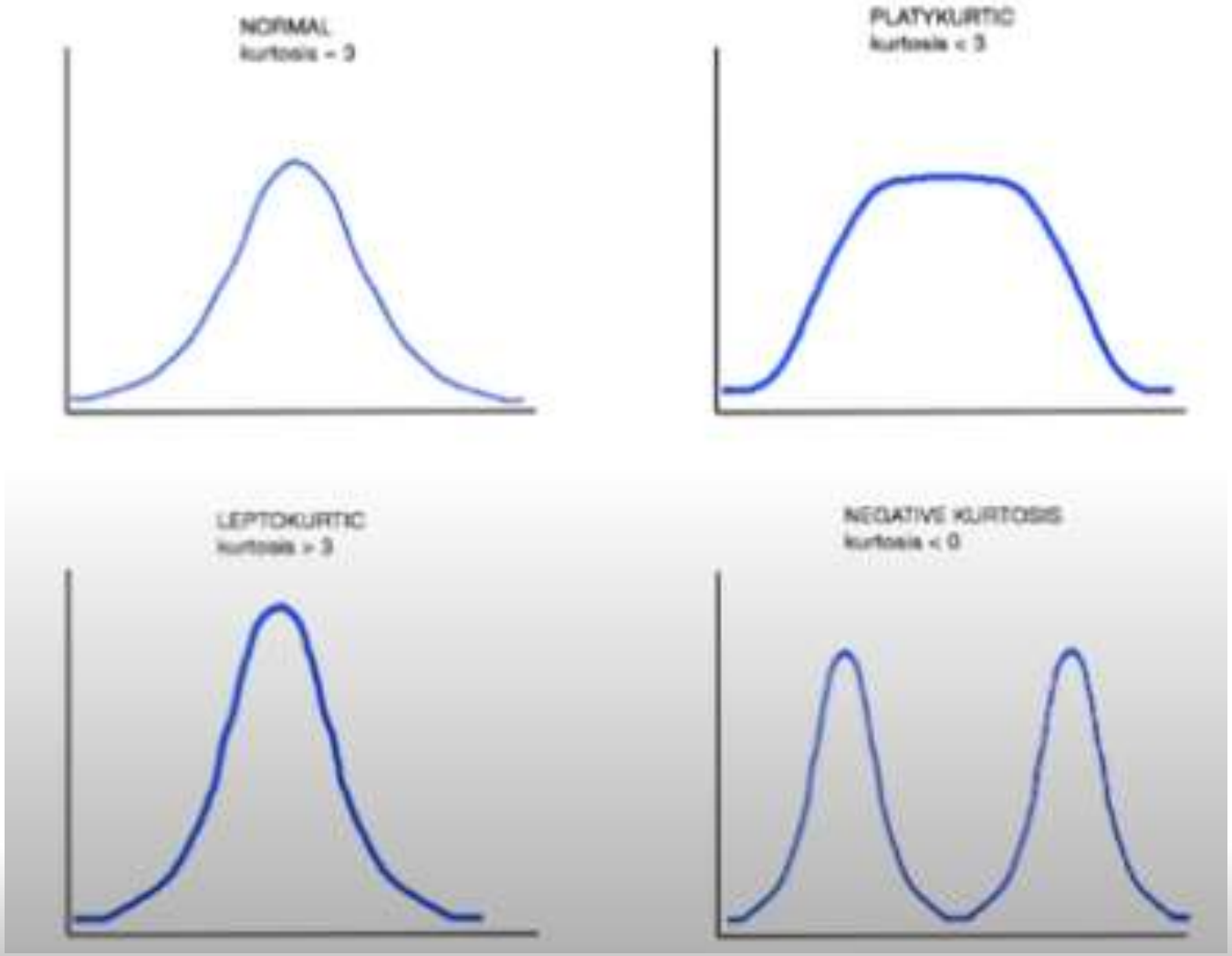
SKEWNESS ANALYSIS

```
Skewness_values = df.skew()
```

Test_No	0
Numune_1_ Relakse_Sonrasi_ En_1	0,613060857
Numune_1_ Relakse_Sonrasi_ En_2	0,660676599
Numune_1_ Relakse_Sonrasi_ En_3	0,83572226
Numune_1_Relakse_Sonrasi_Boy_1	-3,260261705
Numune_1_Relakse_Sonrasi_Boy_2	-3,329913682
Numune_1_Relakse_Sonrasi_Boy_3	-3,327572638
Giris_Alan_Ortalama	-2,394629595
Deterjan_Miktarı(gr)	-2,175448351
Test_Kapasitesi(kg)	-2,175448351
Su_Sertligi	2,076044927
Tambur_Hacmi_(lt)	-1,512634581
SY1_Sure_(sn)	2,603931062
SY1_Devir_(rpm)	-3,330688423
SY1_ED	-0,713845603
SG_Sic_(C)	-0,867887424
I_Bas_Sic (C)	-2,082686198
I_Sure_(sn)	-0,618799546
I_Devir_(rpm)	-1,470632541
I_MHY_ED	1,602676423
I_Tset_(C)	-0,945102159

AY_Su_Mik_(lt)	0,696366716
AY_Sure_(sn)	0,102343503
AY_Devir_(rpm)	-3,330688423
AY_MHY_ED	-0,53507349
AY_Tah_(C)	-0,588976304
Jet_Pompa	-2,175448351
Durulama_Sayisi	0
1D_Su_Mik_(lt)	-1,185669638
1D_Sure_(sn)	-0,334065516
1D_Devir_(rpm)	-3,330688423
1D_MHY_ED	-0,626218518
2D_Su_Mik_(lt)	-0,990964029
2D_Sure_(sn)	-0,726616709
2D_Devir_(rpm)	-3,330688423
2D_MHY_ED	0,653548692
2D_Sikma_(rpm)	2,175836646
3D_Su_Mik_(lt)	1,025273445
3D_Sure_(sn)	0,386394596
3D_Devir_(rpm)	-3,330688423
3D_MHY_ED	-0,621556795
Cikis_Alan_Ortalama	-0,771024791

Skewness is 0 in a normal distribution, so the farther away from 0, the more non-normal the distribution



KURTOSIS ANALYSIS

```
Kurtosisi_values = df.kurtosis()
```

Test_No	-1,270438828
Numune_1_ Relakse_Sonrasi_ En_1	-0,783263545
Numune_1_ Relakse_Sonrasi_ En_2	-0,712381127
Numune_1_ Relakse_Sonrasi_ En_3	-0,230475788
Numune_1_Relakse_Sonrasi_Boy_1	9,401776493
Numune_1_Relakse_Sonrasi_Boy_2	9,642285988
Numune_1_Relakse_Sonrasi_Boy_3	9,641744814
Giris_Alan_Ortalama	6,035844615
Deterjan_Miktarı(gr)	2,763155003
Test_Kapasitesi(kg)	2,763155003
Su_Sertligi	2,570052941
Tambur_Hacmi_(lt)	0,291174697
SY1_Sure_(sn)	5,886342993
SY1_Devir_(rpm)	9,73013713
SY1_ED	-0,439179542
SG_Sic_(C)	2,635950529
I_Bas_Sic (C)	2,54767411
I_Sure_(sn)	-0,392994428
I_Devir_(rpm)	0,247312818
I_MHY_ED	0,739203732
I_Tset_(C)	-0,045574969

AY_Su_Mik_(lt)	2,56451849
AY_Sure_(sn)	-1,619877749
AY_Devir_(rpm)	9,73013713
AY_MHY_ED	-0,994405967
AY_Tah_(C)	-0,337445911
Jet_Pompa	2,763155003
Durulama_Sayisi	0
1D_Su_Mik_(lt)	0,142578778
1D_Sure_(sn)	-1,263010795
1D_Devir_(rpm)	9,73013713
1D_MHY_ED	-0,771316786
2D_Su_Mik_(lt)	0,078394437
2D_Sure_(sn)	-1,453543148
2D_Devir_(rpm)	9,73013713
2D_MHY_ED	-1,363625826
2D_Sikma_(rpm)	2,76564041
3D_Su_Mik_(lt)	0,354449722
3D_Sure_(sn)	0,266542043
3D_Devir_(rpm)	9,73013713
3D_MHY_ED	-0,766140788
Cikis_Alan_Ortalama	0,308871767

The value is often compared to the kurtosis of the normal distribution, which is equal to 3. If the kurtosis is greater than 3, then the dataset has heavier tails than a normal distribution (more in the tails). If the kurtosis is less than 3, then the dataset has lighter tails than a normal distribution (less in the tails).

Detecting Outliers

"Sometimes outliers are bad data, and should be excluded, such as typos. Sometimes they are Wayne Gretzky or Michael Jordan, and should be kept."

- According to **IQR**, (not considering zero (0) values)
  - **Numune_1_ Relakse_Sonrasi_ En_3**
  - **Numune_1_Relakse_Sonrasi_Boy_1**
  - **Numune_1_Relakse_Sonrasi_Boy_2**
  - **Numune_1_Relakse_Sonrasi_Boy_3**
  - **Giris_Alan_Ortalama**
  - **Deterjan_Miktarı(gr)**
  - **Test_Kapasitesi(kg)**
  - **Su_Sertligi**
  - **Tambur_Hacmi_(lt)**
  - **SY1_Sure_(sn)**
  - **SY1_Devir_(rpm)**
  - **SG_Sic_(C)**
  - **I_Bas_Sic (C)** (bcz of zero values)
  - **I_Sure_(sn)**
  - **AY_Su_Mik_(lt)**
  - **AY_Devir_(rpm)**
  - **Jet_Pompa** (bcz of zero values)
  - **1D_Devir_(rpm)**
  - **2D_Devir_(rpm)**
  - **2D_Sikma_(rpm)** *
  - **3D_Sure_(sn) -> (2 outliers)**
  - **3D_Devir_(rpm)**
  - **3D_MHY_ED**
  - **Cikis_Alan_Ortalama -> (2 outliers)**

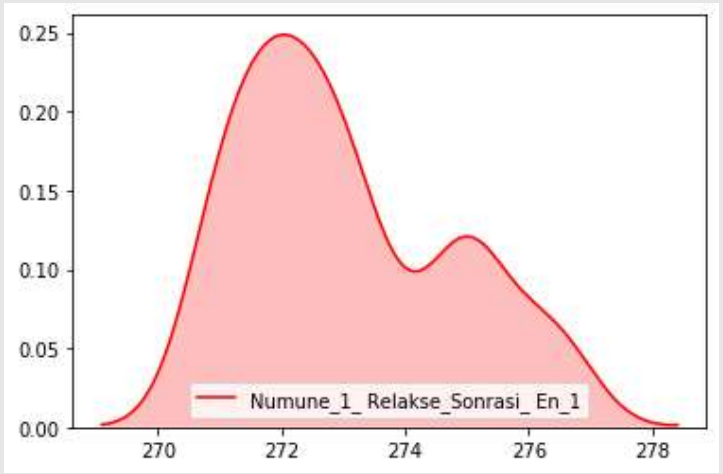
An outlier is a point which falls more than 1.5 times the interquartile range above the third (75%) quartile or below the first quartile (25%).

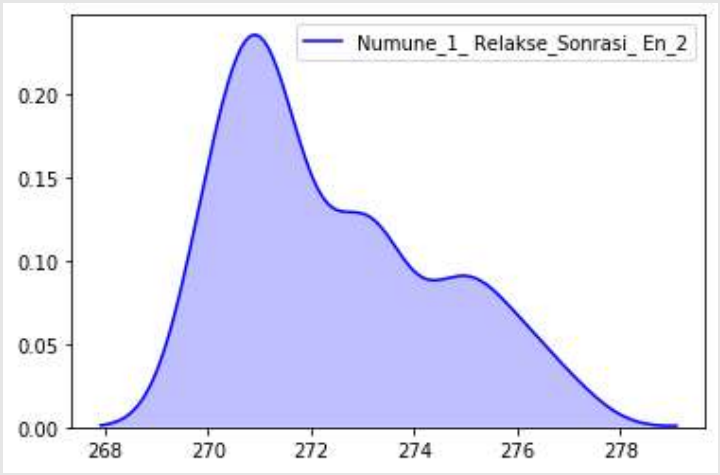
- According to **Z score**,
  - **Numune_1_Relakse_Sonrasi_Boy_1**
  - **Numune_1_Relakse_Sonrasi_Boy_2**
  - **Numune_1_Relakse_Sonrasi_Boy_3**
  - **Giris_Alan_Ortalama**
  - **SY1_Sure_(sn)**
  - **SY1_Devir_(rpm)**
  - **SG_Sic_(C)**
  - **AY_Su_Mik_(lt)**
  - **AY_Devir_(rpm)**
  - **1D_Devir_(rpm)**
  - **2D_Devir_(rpm)**
  - **3D_Su_Mik_(lt)**
  - **3D_Sure_(sn)**
  - **3D_Devir_(rpm)**
  - **Cikis_Alan_Ortalama**

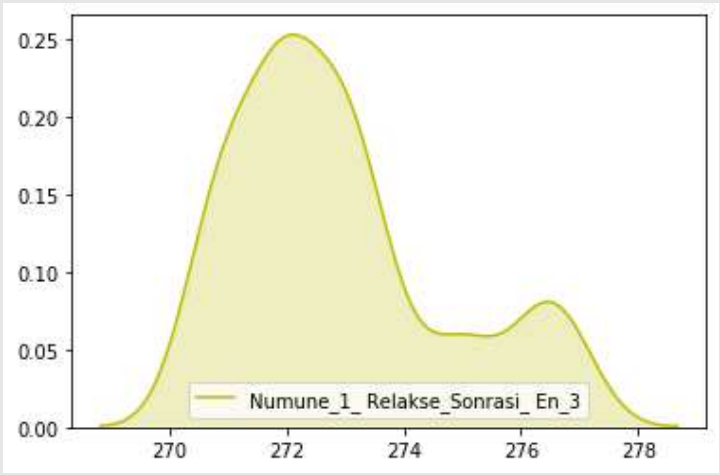
variables have values more than theshold which is 3.

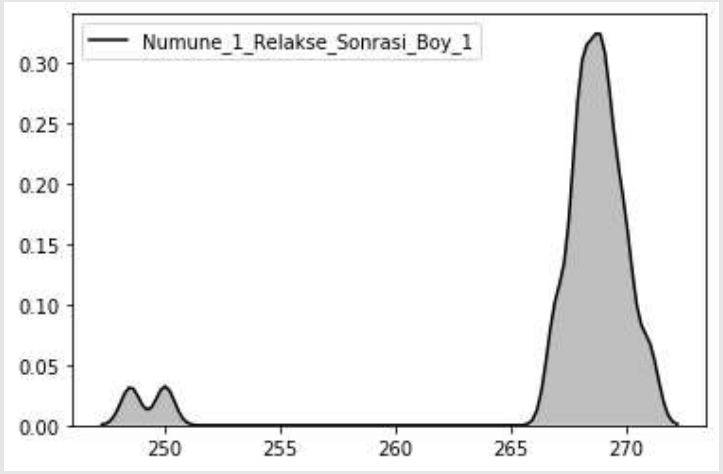
Formula for Z score = (Observation - Mean)/Standard Deviation

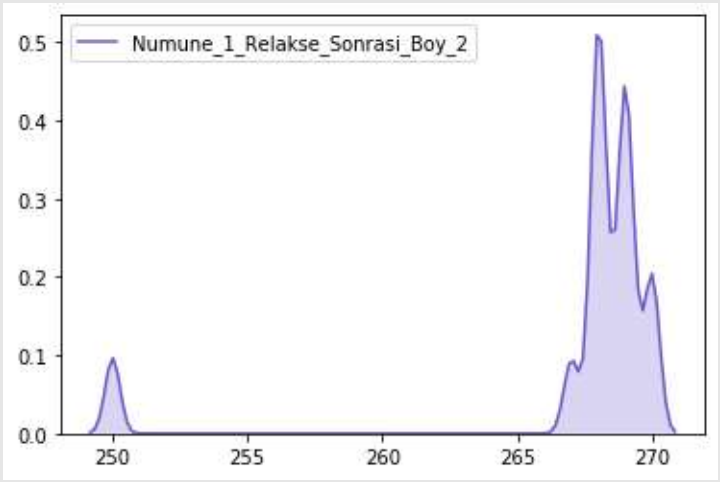
```
#########################################################################
#############   Detecting Outliers- Uç noktalarrrr ##############
#########################################################################
df.shape
(180, 42)

Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3-Q1
print(IQR)

df = df[~((df < (Q1-1.5*IQR)) |(df > (Q3+1.5*IQR))).any(axis=1)]
df.shape
(48, 42)  #132 rows are outliers.....

#z-score-outliers

from scipy import stats
import numpy as np
z = np.abs(stats.zscore(df))
print(z)
z_score = pd.DataFrame(data = z, index= range(180), columns = df.columns)

z_score.to_excel('C:\\Users\\DSİ\\Desktop\\Z_score_outliers.xlsx')
threshold = 3
print(np.where(z > 3))
# print(z[18][21])
```

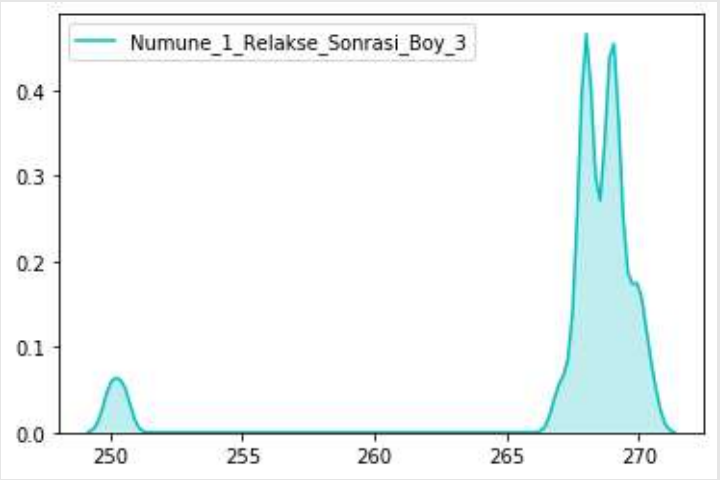

Numune_1_ Relakse_Sonrasi_ En_1
0.25
0.20
0.15
0.10
0.05
0.00
270
272
274
276
278

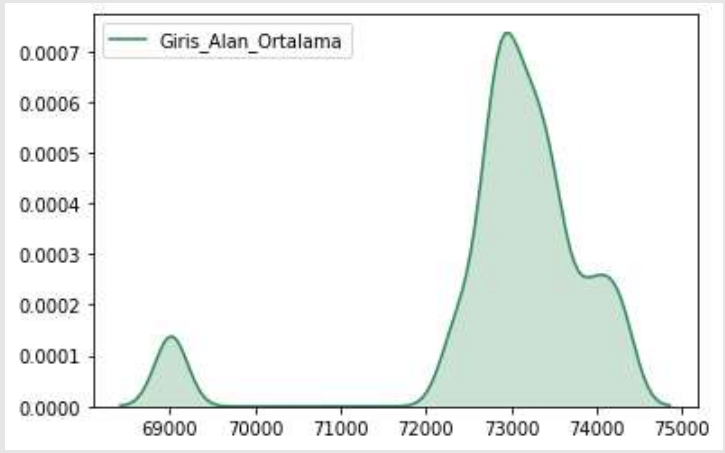

Numune_1_ Relakse_Sonrasi_ En_2
0.20
0.15
0.10
0.05
0.00
268
270
272
274
276
278

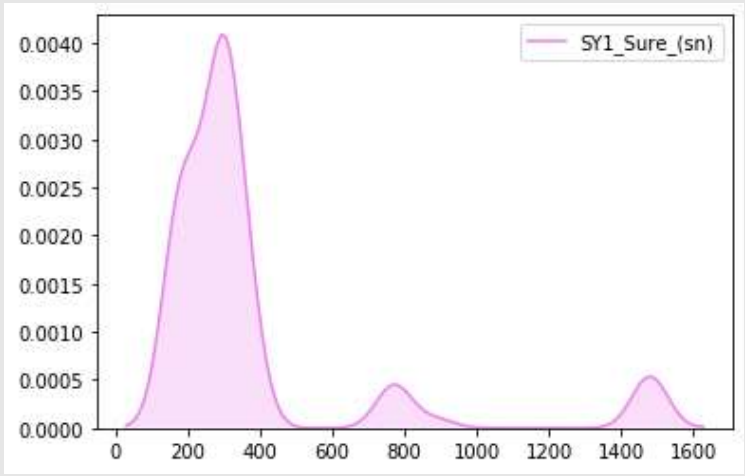

Numune_1_ Relakse_Sonrasi_ En_3
0.25
0.20
0.15
0.10
0.05
0.00
270
272
274
276
278

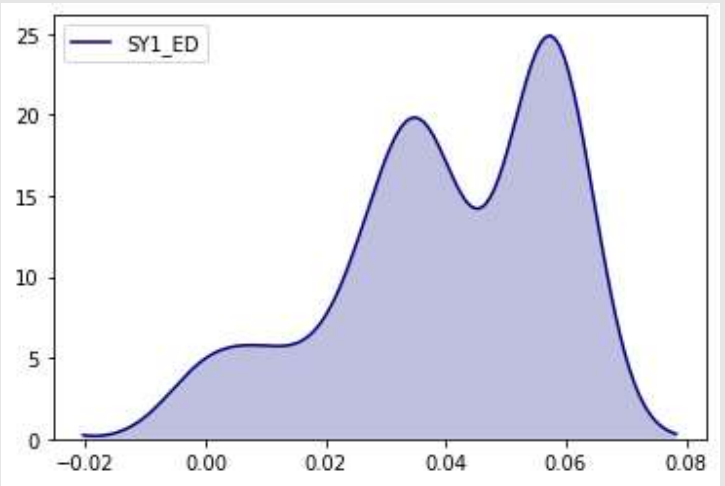

Numune_1_Relakse_Sonrasi_Boy_1
0.30
0.25
0.20
0.15
0.10
0.05
0.00
250
255
260
265
270

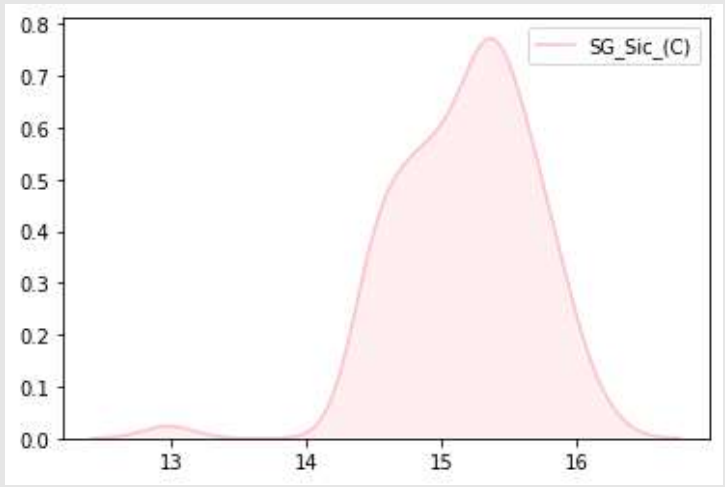

Numune_1_Relakse_Sonrasi_Boy_2
0.5
0.4
0.3
0.2
0.1
0.0
250
255
260
265
270

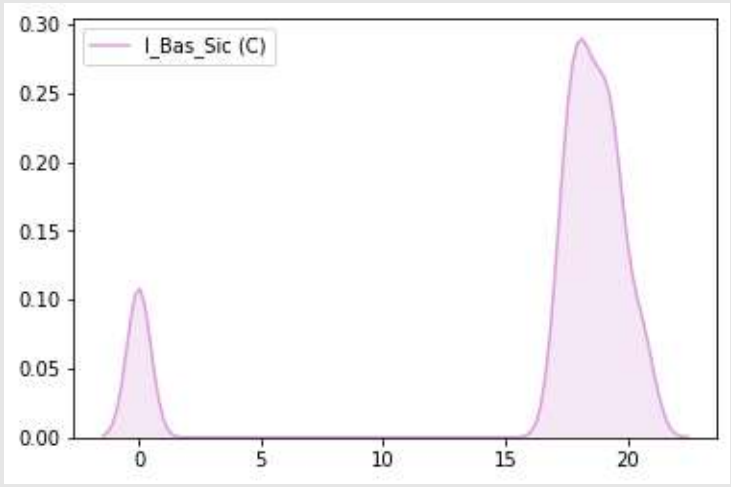

Numune_1_Relakse_Sonrasi_Boy_3
0.4
0.3
0.2
0.1
0.0
250
255
260
265
270

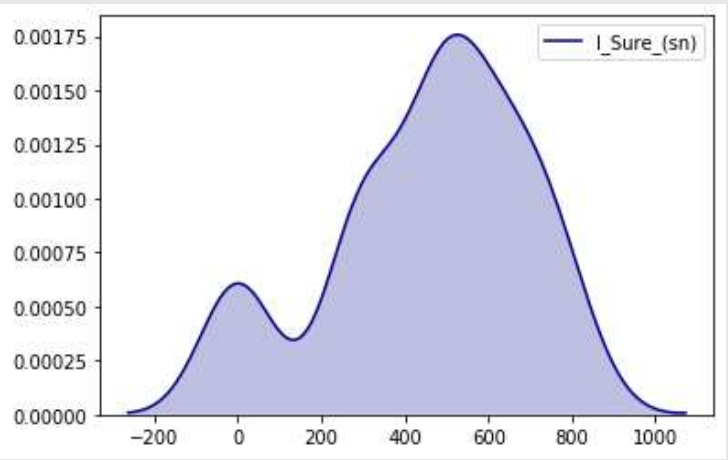

Giris_Alan_Ortalama
0.0007
0.0006
0.0005
0.0004
0.0003
0.0002
0.0001
0.0000
69000
70000
71000
72000
73000
74000
75000

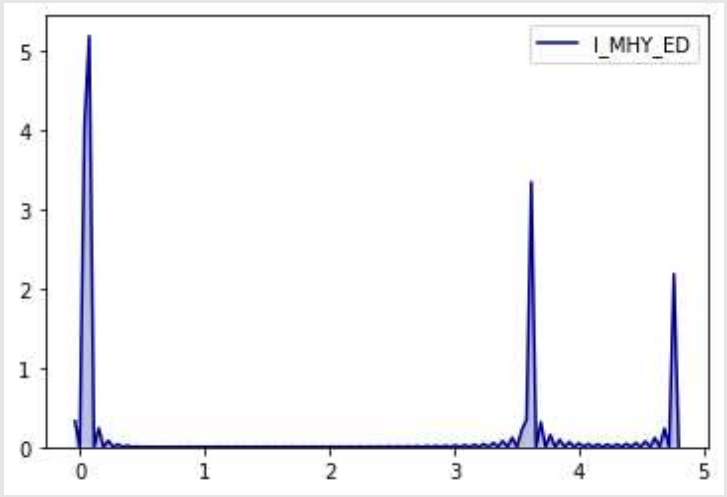

SY1_Sure_(sn)
0.0040
0.0035
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
0
200
400
600
800
1000
1200
1400
1600

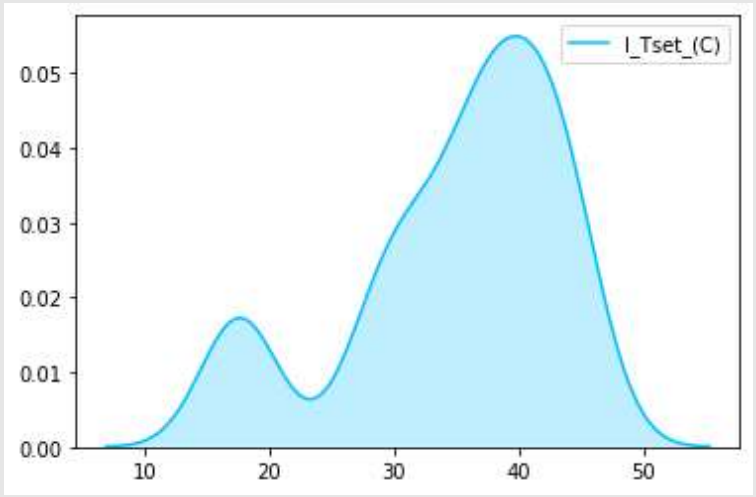

SY1_ED
25
20
15
10
5
0
−0.02
0.00
0.02
0.04
0.06
0.08

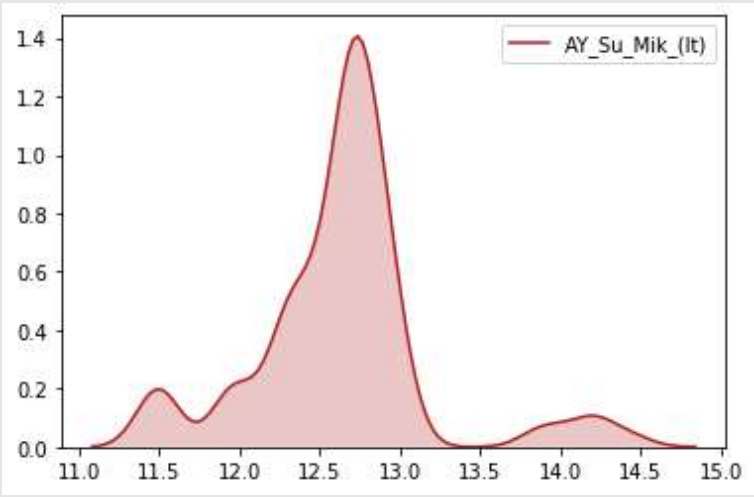

SG_Sic_(C)
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
13
14
15
16

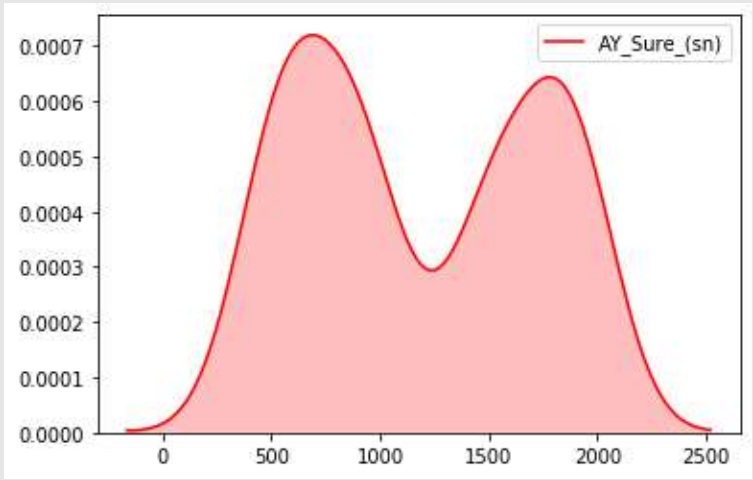

I_Bas_Sic (C)
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
5
10
15
20

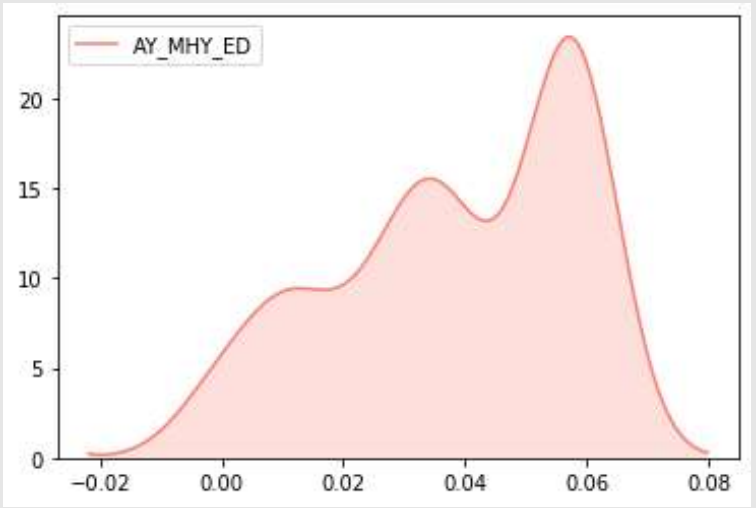

I_Sure_(sn)
0.00175
0.00150
0.00125
0.00100
0.00075
0.00050
0.00025
0.00000
−200
0
200
400
600
800
1000

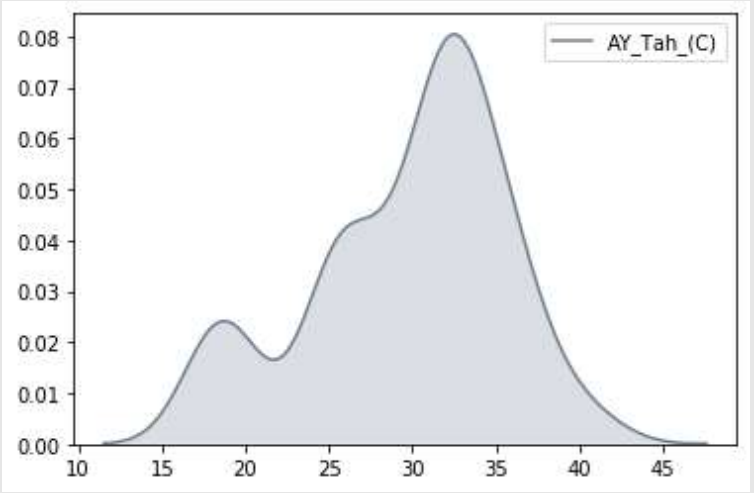

I_MHY_ED
0
1
2
3
4
5

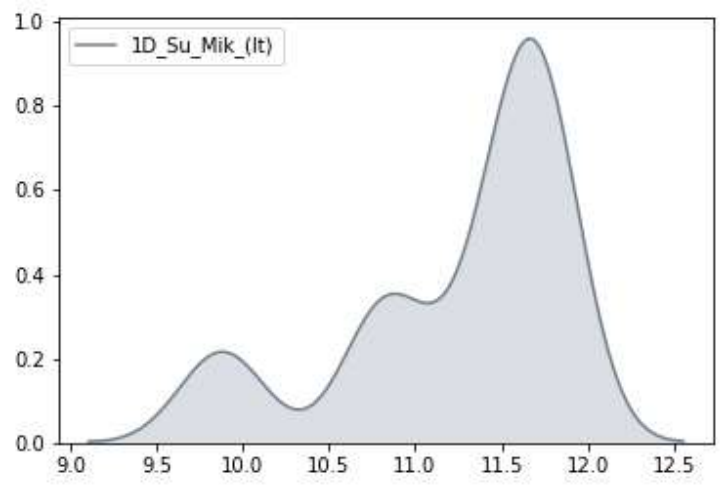

I_Tset_(C)
0.00
0.01
0.02
0.03
0.04
0.05
10
20
30
40
50

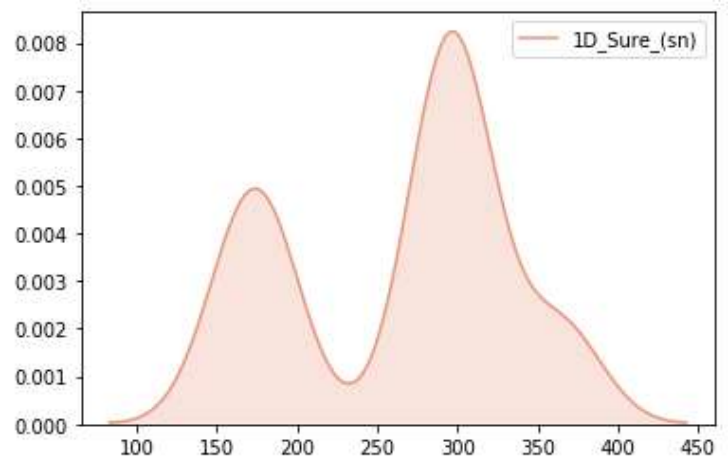

AY_Su_Mik_(lt)
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
11.0
11.5
12.0
12.5
13.0
13.5
14.0
14.5
15.0

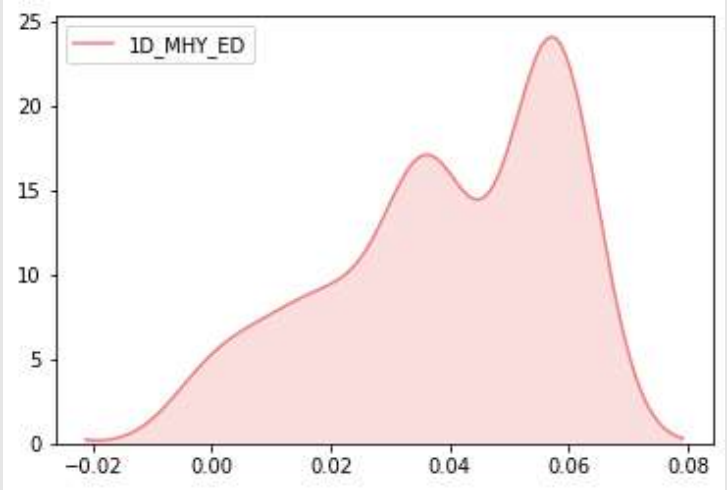

AY_Sure_(sn)
0.0000
0.0001
0.0002
0.0003
0.0004
0.0005
0.0006
0.0007
0
500
1000
1500
2000
2500

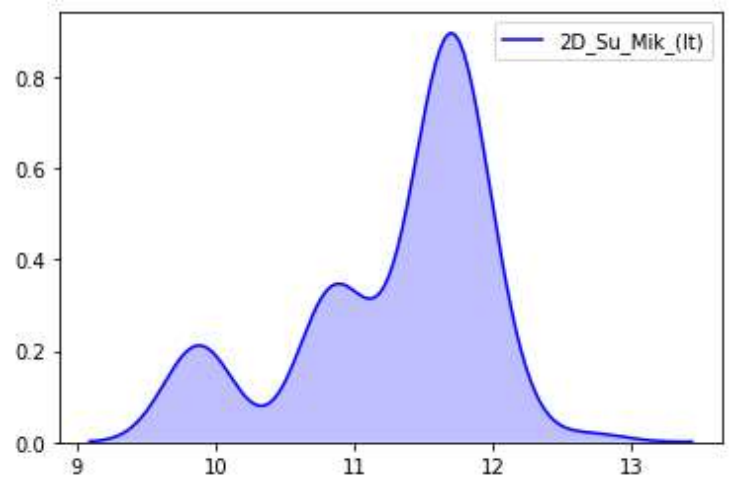

AY_MHY_ED
0
5
10
15
20
−0.02
0.00
0.02
0.04
0.06
0.08

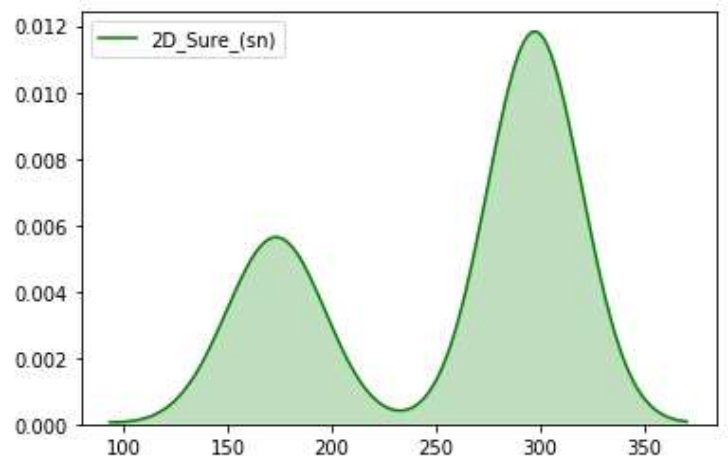

AY_Tah_(C)
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
10
15
20
25
30
35
40
45

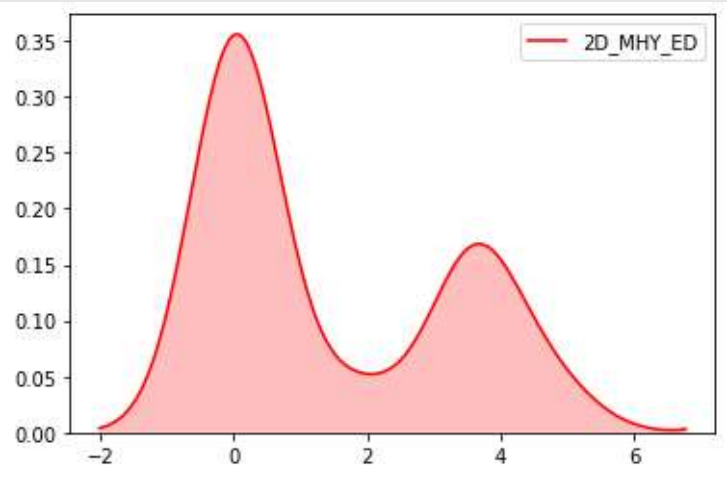

1D_Su_Mik_(lt)
1.0
0.8
0.6
0.4
0.2
0.0
9.0
9.5
10.0
10.5
11.0
11.5
12.0
12.5

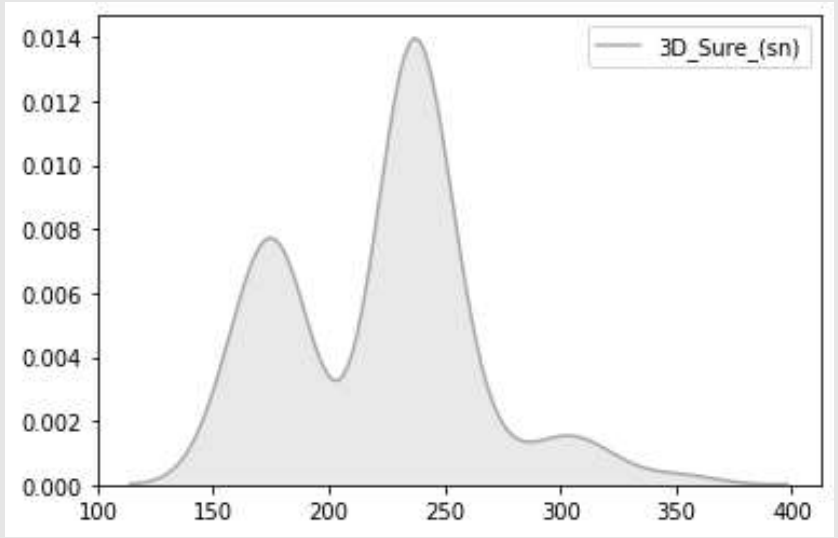

1D_Sure_(sn)
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.000
100
150
200
250
300
350
400
450

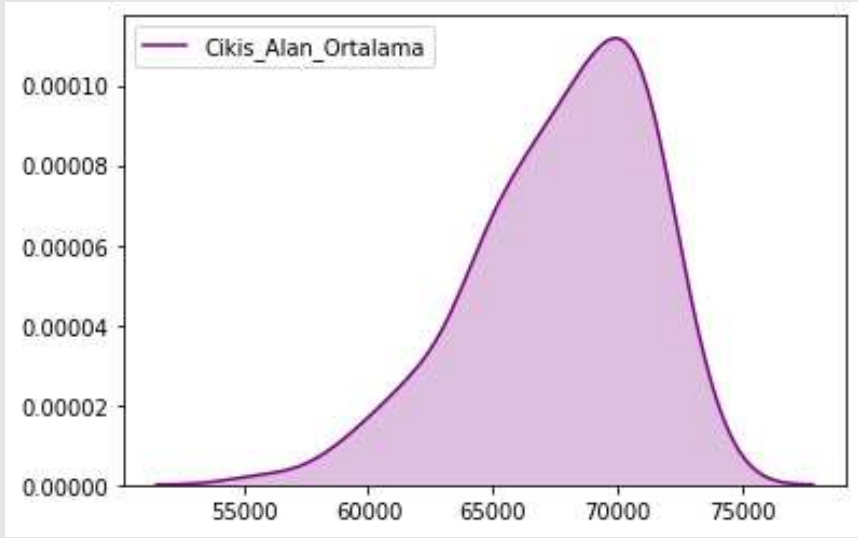

1D_MHY_ED
25
20
15
10
5
0
−0.02
0.00
0.02
0.04
0.06
0.08

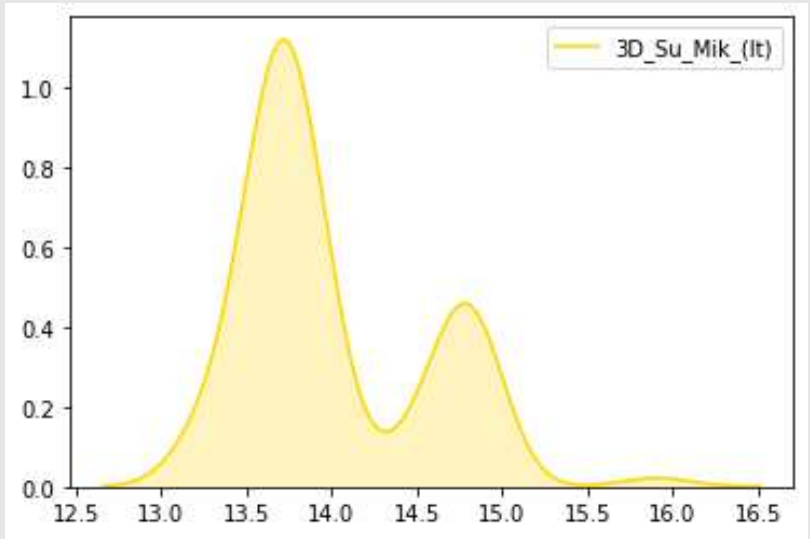

2D_Su_Mik_(lt)
0.8
0.6
0.4
0.2
0.0
9
10
11
12
13

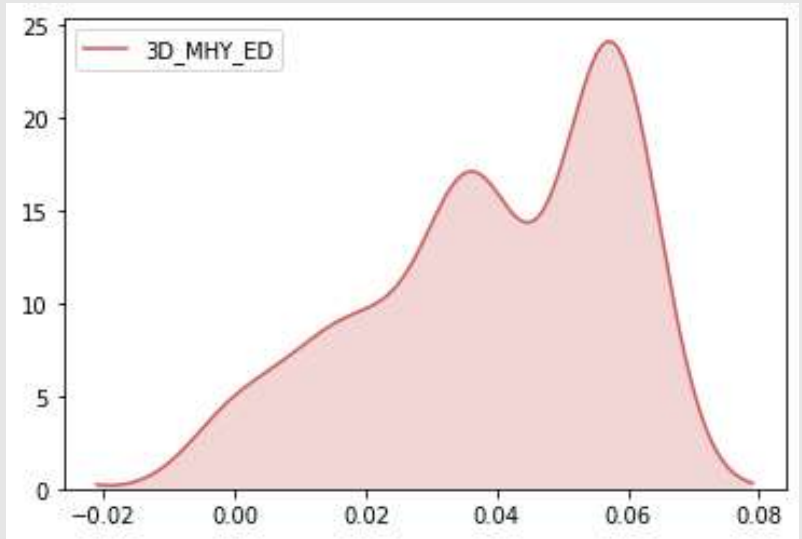

2D_Sure_(sn)
0.012
0.010
0.008
0.006
0.004
0.002
0.000
100
150
200
250
300
350


2D_MHY_ED
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−2
0
2
4
6


3D_Sure_(sn)
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
100
150
200
250
300
350
400


Cikis_Alan_Ortalama
0.00010
0.00008
0.00006
0.00004
0.00002
0.00000
55000
60000
65000
70000
75000

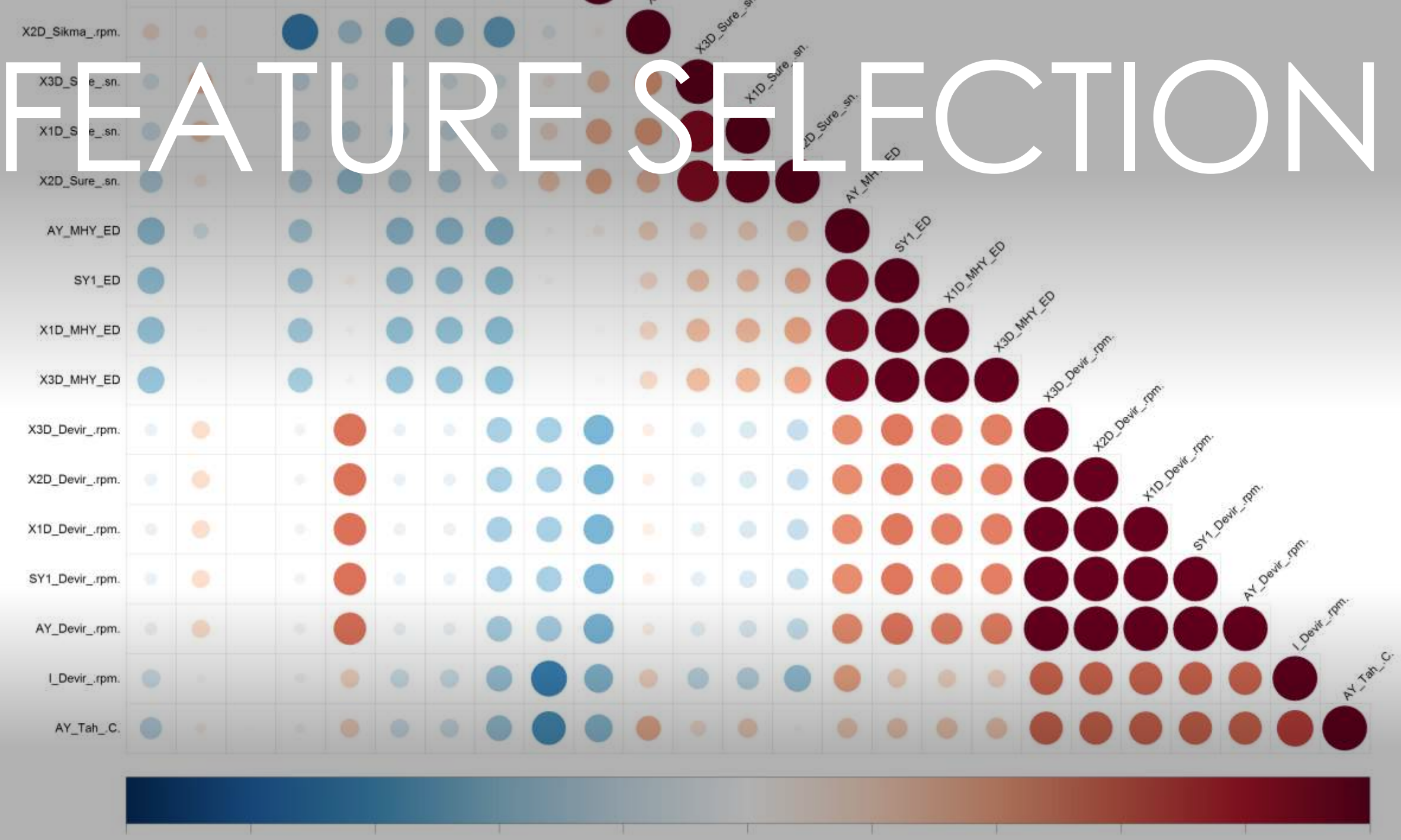

3D_Su_Mik_(lt)
1.0
0.8
0.6
0.4
0.2
0.0
12.5
13.0
13.5
14.0
14.5
15.0
15.5
16.0
16.5


3D_MHY_ED
25
20
15
10
5
0
−0.02
0.00
0.02
0.04
0.06
0.08

Variables that do not have a density

- Deterjan_Miktarı(gr)
- Su_Sertligi
- SY1_Devir_(rpm)
- AY_Devir_(rpm)
- 1D_Devir_(rpm)
- 2D_Sikma_(rpm)
- Test_Kapasitesi(kg)
- Tambur_Hacmi_(lt)
- 1_Devir_(rpm)
- Jet_Pompa
- 2D_Devir_(rpm)
- 3D_Devir_(rpm)

FEATURE SELECTION



PEARSON VS. SPEARMAN'S RANK CORRELATION ANALYSIS

Pearson *r* correlation is a parametric correlation statistic to measure the degree of the relationship between linearly related variables.

The Spearman's rank coefficient of correlation is a nonparametric measure of rank correlation.

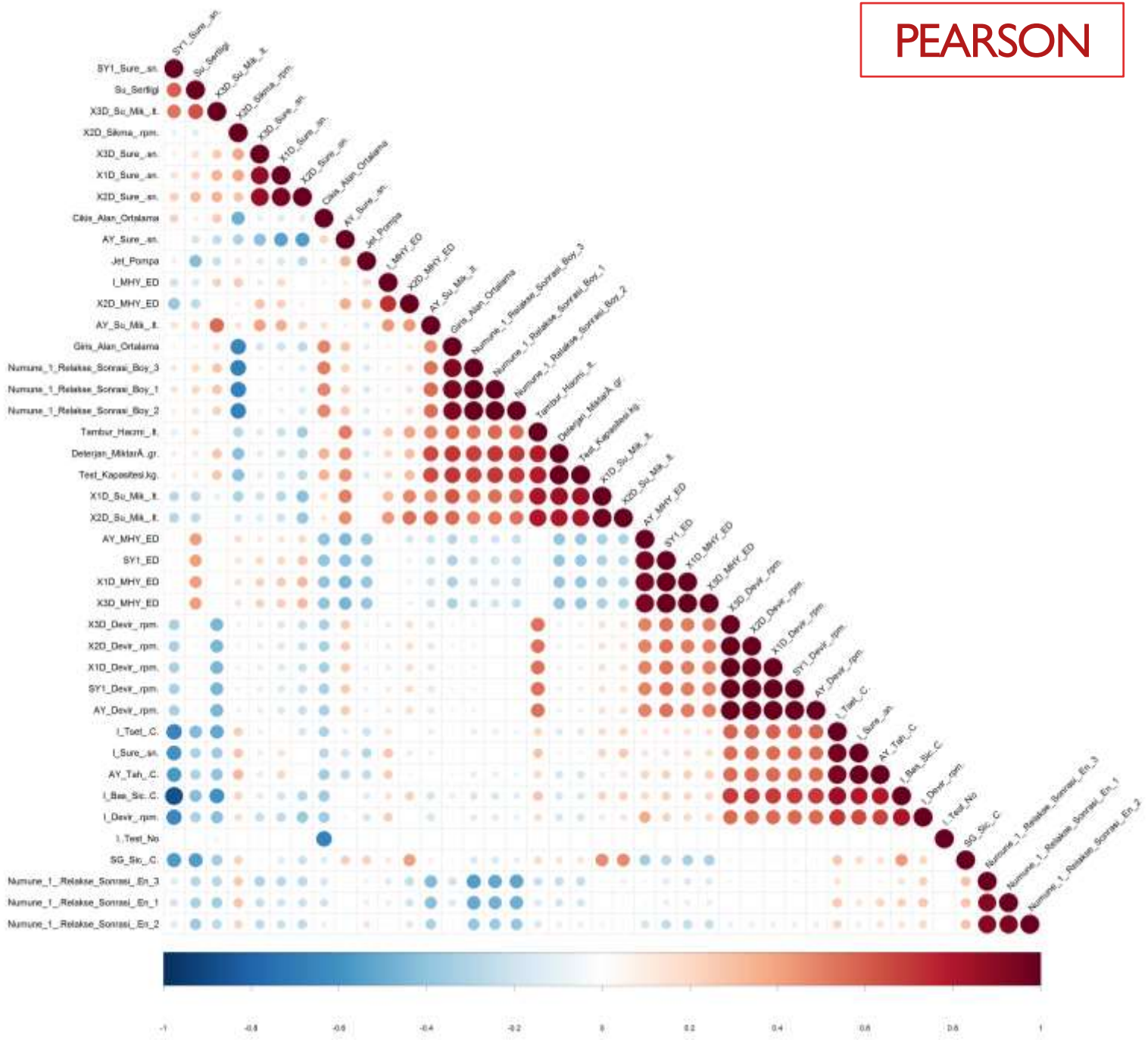
Size of Correlation	Interpretation
.90 to 1.00 (−.90 to −1.00)	Very high positive (negative) correlation
.70 to .90 (−.70 to −.90)	High positive (negative) correlation
.50 to .70 (−.50 to −.70)	Moderate positive (negative) correlation
.30 to .50 (−.30 to −.50)	Low positive (negative) correlation
.00 to .30 (.00 to −.30)	negligible correlation

Feature\Response	Continuous	Categorical
Continuous	Pearson's Correlation	LDA
Categorical	Anova	Chi-Square

To be able to calculate pearson's correlation, the condition of normality and linearity should be provided for each variable.

When variables do not have normal distribution, Spearman Rank correlation coefficient is preferred.

PEARSON



DEĞER	Sol_Korelasyon	to	Üst_Korelasyon	Çıkarılacak
1,00	Deterjan_Miktarı(gr)	=>	Test_Kapasitesi(kg)	Deterjan_Miktarı(gr)
1,00	SY1_Devir_(rpm)	=>	AY_Devir_(rpm)	AY_Devir_(rpm)
1,00	SY1_Devir_(rpm)	=>	1D_Devir_(rpm)	1D_Devir_(rpm)
1,00	SY1_Devir_(rpm)	=>	2D_Devir_(rpm)	2D_Devir_(rpm)
1,00	SY1_Devir_(rpm)	=>	3D_Devir_(rpm)	3D_Devir_(rpm)
1,00	AY_Devir_(rpm)	=>	1D_Devir_(rpm)	*
1,00	AY_Devir_(rpm)	=>	2D_Devir_(rpm)	*
1,00	AY_Devir_(rpm)	=>	3D_Devir_(rpm)	*
1,00	1D_Devir_(rpm)	=>	2D_Devir_(rpm)	*
1,00	1D_Devir_(rpm)	=>	3D_Devir_(rpm)	*
1,00	2D_Devir_(rpm)	=>	3D_Devir_(rpm)	*
1,00	1D_MHY_ED	=>	3D_MHY_ED	3D_MHY_ED
0,99	Numune_1_Relakse_Sonrasi_Boy_1	=>	Numune_1_Relakse_Sonrasi_Boy_3	Numune_1_Relakse_Sonrasi_Boy_1
0,99	Numune_1_Relakse_Sonrasi_Boy_1	=>	Numune_1_Relakse_Sonrasi_Boy_2	*
0,99	Numune_1_Relakse_Sonrasi_Boy_2	=>	Numune_1_Relakse_Sonrasi_Boy_3	Numune_1_Relakse_Sonrasi_Boy_2
0,99	SY1_ED	=>	1D_MHY_ED	SY1_ED
0,99	SY1_ED	=>	3D_MHY_ED	*
0,98	I_Sure_(sn)	=>	AY_Tah_(C)	I_Sure_(sn)
0,97	1D_Su_Mik_(lt)	=>	2D_Su_Mik_(lt)	2D_Su_Mik_(lt)
0,95	SY1_ED	=>	AY_MHY_ED	*
0,95	1D_Sure_(sn)	=>	2D_Sure_(sn)	2D_Sure_(sn)
0,95	I_Sure_(sn)	=>	I_Tset_(C)	I_Sure_(sn)
0,94	I_Tset_(C)	=>	AY_Tah_(C)	I_Tset_(C)
0,94	AY_MHY_ED	=>	3D_MHY_ED	*
0,94	AY_MHY_ED	=>	1D_MHY_ED	1D_MHY_ED
0,94	Numune_1_Relakse_Sonrasi_Boy_1	=>	Giris_Alan_Ortalama	*
0,93	Numune_1_Relakse_Sonrasi_En_1	=>	Numune_1_Relakse_Sonrasi_En_2	Numune_1_Relakse_Sonrasi_En_2
0,93	Numune_1_Relakse_Sonrasi_Boy_2	=>	Giris_Alan_Ortalama	*
0,92	Numune_1_Relakse_Sonrasi_Boy_3	=>	Giris_Alan_Ortalama	Numune_1_Relakse_Sonrasi_Boy_3
0,91	Numune_1_Relakse_Sonrasi_En_2	=>	Numune_1_Relakse_Sonrasi_En_3	*
0,91	Numune_1_Relakse_Sonrasi_En_1	=>	Numune_1_Relakse_Sonrasi_En_3	Numune_1_Relakse_Sonrasi_En_3
0,88	1D_Sure_(sn)	=>	3D_Sure_(sn)	3D_Sure_(sn)
0,87	2D_Sure_(sn)	=>	3D_Sure_(sn)	*
0,85	I_Bas_Sic(C)	=>	I_Tset_(C)	*
0,85	Test_Kapasitesi(kg)	=>	1D_Su_Mik_(lt)	1D_Su_Mik_(lt)
0,85	Deterjan_Miktarı(gr)	=>	1D_Su_Mik_(lt)	*
0,82	Test_Kapasitesi(kg)	=>	2D_Su_Mik_(lt)	*
0,82	Deterjan_Miktarı(gr)	=>	2D_Su_Mik_(lt)	*
0,82	Tambur_Hacmi_(lt)	=>	1D_Su_Mik_(lt)	*
0,81	I_Bas_Sic(C)	=>	I_Devir_(rpm)	I_Devir_(rpm) ???
0,80	Tambur_Hacmi_(lt)	=>	2D_Su_Mik_(lt)	*
0,79	I_Bas_Sic(C)	=>	I_Sure_(sn)	*
0,79	I_Bas_Sic(C)	=>	AY_Tah_(C)	AY_Tah_(C)
0,78	Test_Kapasitesi(kg)	=>	Tambur_Hacmi_(lt)	Tambur_Hacmi_(lt)
0,78	Deterjan_Miktarı(gr)	=>	Tambur_Hacmi_(lt)	*
0,73	I_MHY_ED	=>	2D_MHY_ED	2D_MHY_ED
0,71	Giris_Alan_Ortalama	=>	Deterjan_Miktarı(gr)	
0,71	Giris_Alan_Ortalama	=>	Test_Kapasitesi(kg)	
-0,88	SY1_Sure_(sn)	=>	I_Bas_Sic(C)	SY1_Sure_(sn)

Sol Başlıkları Brleştir

Renklendir

İstatistik Tablosu Yap

SPEARMAN CORR

	Test_No	Numune_1_ Relakse_Sonrasi_ En_1	Numune_1_ Relakse_Sonrasi_ En_2	Numune_1_ Relakse_Sonrasi_ En_3	Numune_1_ Relakse_Sonrasi_Boy_1	Numune_1_Relakse_Sonrasi_Boy_2	Numune_1_Relakse_Sonrasi_Boy_3	Giris_Alan_Ortalama	Deterjan_Miktarı(gr)	Test_Kapasitesi(kg)	Su_Sertligi	Tambur_Hacmi_(lt)	SY1_Sure_(sn)	SY1_Devir_(rpm)	SY1_ED	SG_Sic_(C)	I_Bas_Sic (C)	I_Sure_(sn)	I_Devir_(rpm)	I_MHY_ED	I_Tset_(C)	AY_Su_Mik_(lt)	AY_Sure_(sn)	AY_Devir_(rpm)	AY_MHY_ED	AY_Tah_(C)	Jet_Pompa	1D_Su_Mik_(lt)	1D_Sure_(sn)	1D_Devir_(rpm)	1D_MHY_ED	2D_Su_Mik_(lt)	2D_Sure_(sn)	2D_Devir_(rpm)	2D_MHY_ED	2D_Sikma_(rpm)	3D_Su_Mik_(lt)	3D_Sure_(sn)	3D_Devir_(rpm)	3D_MHY_ED	Cikis_Alan_Ortalama
Test_No	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03	0,05	-0,02	0,00	0,00	0,02	0,02	0,00	0,00	0,00	-0,02	0,00	-0,01	0,01	0,00	0,00	-0,01	-0,03	0,00	0,00	0,00	-0,03	-0,04	0,00	0,00	-0,72
Numune_1_ Relakse_Sonrasi_ En_1	1,00	0,94	0,86	0,08	0,08	-0,01	0,51	-0,07	-0,07	-0,36	-0,03	0,05	0,13	-0,14	0,28	0,36	0,12	0,31	0,07	0,28	-0,40	0,09	0,13	-0,10	0,20	0,11	0,31	-0,19	0,13	-0,13	0,21	-0,21	0,13	-0,18	0,18	-0,34	-0,24	0,13	-0,13	-0,13
Numune_1_ Relakse_Sonrasi_ En_2	1,00	0,85	0,02	-0,05	-0,09	0,51	-0,06	-0,06	-0,40	-0,08	0,12	0,05	-0,22	0,28	0,32	0,14	0,23	0,06	0,30	-0,35	0,10	0,05	-0,19	0,21	0,18	0,31	-0,17	0,05	-0,21	0,25	-0,21	0,05	-0,18	0,20	-0,27	-0,23	0,05	-0,21	-0,06
Numune_1_ Relakse_Sonrasi_ En_3	1,00	0,00	0,01	-0,10	0,45	-0,11	-0,11	-0,30	-0,07	-0,01	0,06	-0,13	0,31	0,31	-0,03	0,29	0,04	0,17	-0,45	0,19	0,06	-0,07	0,07	0,10	0,28	-0,35	0,06	-0,14	0,14	-0,37	0,06	-0,20	0,11	-0,32	-0,38	0,06	-0,14	-0,11
Numune_1_Relakse_Sonrasi_Boy_1	1,00	0,83	0,86	0,74	0,41	0,41	0,41	0,39	0,29	0,01	0,10	-0,30	-0,17	0,18	-0,13	-0,11	0,01	0,36	-0,18	0,01	0,12	0,07	-0,25	0,17	0,02	0,01	0,11	0,14	0,05	0,01	-0,13	-0,34	0,41	0,05	0,01	0,11	0,18			
Numune_1_Relakse_Sonrasi_Boy_2	1,00	0,74	0,70	0,46	0,46	0,24	0,46	0,08	0,07	-0,06	-0,07	-0,02	0,24	0,03	-0,10	0,08	0,30	-0,05	0,07	-0,03	0,11	-0,28	0,27	-0,03	0,07	-0,06	0,24	-0,01	0,07	-0,08	-0,31	0,23	0,00	0,07	-0,06	0,15			
Numune_1_Relakse_Sonrasi_Boy_3	1,00	0,65	0,35	0,35	0,42	0,29	0,12	-0,03	0,16	-0,35	-0,18	0,07	-0,15	-0,12	-0,11	0,32	-0,28	-0,03	0,15	-0,07	-0,29	0,06	-0,04	-0,03	0,17	0,01	0,01	-0,03	-0,12	-0,40	0,37	0,01	-0,03	0,17	0,21				
Giris_Alan_Ortalama	1,00	0,48	0,48	0,00	0,39	0,15	0,04	-0,25	0,00	0,03	0,18	0,12	-0,12	0,14	0,10	0,09	0,04	-0,20	0,07	-0,10	0,44	-0,28	0,04	-0,24	0,34	-0,26	0,04	-0,22	-0,39	0,16	-0,27	0,04	-0,24	0,24					
Deterjan_Miktarı(gr)	1,00	1,00	0,07	0,78	-0,02	0,00	-0,43	0,16	-0,27	0,05	-0,12	-0,19	-0,11	0,59	0,44	0,00	-0,42	-0,17	-0,15	0,59	-0,17	0,00	-0,43	0,59	-0,12	0,00	0,14	-0,38	0,34	-0,15	0,00	-0,43	0,28						
Test_Kapasitesi(kg)	1,00	0,07	0,78	-0,02	0,00	-0,43	0,16	-0,27	0,05	-0,12	-0,19	-0,11	0,59	0,44	0,00	-0,42	-0,17	-0,15	0,59	-0,17	0,00	-0,43	0,59	-0,12	0,00	0,14	-0,38	0,34	-0,15	0,00	-0,43	0,28							
Su_Sertligi	1,00	0,08	0,03	-0,39	0,51	-0,30	-0,40	-0,29	-0,53	-0,04	-0,48	0,50	-0,37	-0,39	0,47	-0,34	-0,39	-0,26	0,12	-0,39	0,47	-0,24	0,17	-0,39	0,06	-0,06	0,62	0,18	-0,39	0,47	0,06								
Tambur_Hacmi_(lt)	1,00	-0,25	0,46	-0,08	0,21	0,02	0,29	0,20	0,08	0,14	0,32	0,49	0,46	-0,08	0,13	-0,20	0,62	-0,09	0,46	-0,08	0,63	-0,06	0,46	0,40	-0,26	0,04	-0,06	0,46	-0,08	-0,01									
SY1_Sure_(sn)	1,00	-0,05	-0,20	-0,47	0,03	0,15	-0,13	-0,31	0,20	0,15	-0,15	-0,05	-0,14	0,19	0,07	-0,25	-0,12	-0,05	-0,16	-0,19	-0,20	-0,05	-0,66	0,10	0,35	-0,13	-0,05	-0,16	0,17										
SY1_Devir_(rpm)	1,00	0,09	-0,10	0,54	0,45	0,71	0,13	0,49	-0,37	0,33	1,00	0,13	0,47	0,00	0,04	-0,16	1,00	0,13	0,06	-0,19	1,00	0,09	0,00	-0,57	-0,15	1,00	0,13	-0,26											
SY1_ED	1,00	-0,33	0,03	0,02	-0,01	0,41	-0,02	-0,08	-0,54	0,09	0,97	0,09	-0,43	-0,22	0,19	0,09	1,00	-0,23	0,20	0,09	0,16	0,12	0,08	0,28	0,09	1,00	-0,36												
SG_Sic_(C)	1,00	0,22	0,01	0,05	0,05	0,11	-0,10	0,33	-0,10	-0,36	0,01	0,26	0,48	0,12	-0,10	-0,37	0,51	0,11	-0,10	0,35	0,26	-0,33	0,07	-0,10	-0,37	-0,14													
I_Bas_Sic (C)	1,00	0,55	0,73	0,28	0,67	-0,45	-0,15	0,54	0,11	0,60	-0,02	0,06	-0,19	0,54	0,06	0,17	-0,28	0,54	-0,23	0,45	-0,43	-0,19	0,54	0,06	-0,38														
I_Sure_(sn)	1,00	0,56	0,46	0,89	0,00	-0,20	0,45	0,05	0,95	-0,33	0,28	0,27	0,45	0,05	0,44	0,19	0,45	0,03	0,25	-0,05	0,26	0,45	0,05	-0,21															
I_Devir_(rpm)	1,00	0,51	0,61	-0,44	0,10	0,71	0,14	0,59	-0,12	0,19	-0,23	0,71	0,02	0,21	-0,29	0,71	-0,01	0,12	-0,42	-0,23	0,71	0,02	-0,31																
I_MHY_ED	1,00	0,35	-0,05	-0,25	0,13	0,49	0,48	-0,19	0,22	0,25	0,13	0,41	0,38	0,20	0,13	0,53	0,36	0,15	0,23	0,13	0,41	-0,37																	
I_Tset_(C)	1,00	-0,27	-0,09	0,49	0,02	0,90	-0,20	0,30	0,06	0,49	0,02	0,38	-0,02	0,49	-0,15	0,23	-0,25	0,07	0,49	0,02	-0,25																		
AY_Su_Mik_(lt)	1,00	-0,04	-0,37	-0,09	-0,14	-0,38	0,00	0,36	-0,37	-0,09	0,19	0,35	-0,37	0,18	0,04	0,71	0,39	-0,37	-0,09	0,16																			
AY_Sure_(sn)	1,00	0,33	-0,53	-0,21	0,32	0,34	-0,30	0,33	-0,54	0,26	-0,26	0,33	0,26	-0,26	-0,28	-0,31	0,33	-0,54	0,12																				
AY_Devir_(rpm)	1,00	0,13	0,47	0,00	0,04	-0,16	1,00	0,13	0,06	-0,19	1,00	0,09	0,00	-0,57	-0,15	1,00	0,13	-0,26																					
AY_MHY_ED	1,00	0,13	-0,42	-0,22	0,13	0,13	0,97	-0,21	0,13	0,13	0,11	0,15	0,11	0,22	0,13	0,97	-0,38																						
AY_Tah_(C)	1,00	-0,25	0,17	0,33	0,47	0,11	0,31	0,23	0,47	0,02	0,36	-0,17	0,31	0,47	0,11	-0,29																							
Jet_Pompa	1,00	-0,01	-0,06	0,00	-0,43	0,05	-0,08	0,00	0,14	0,15	-0,31	-0,20	0,00	-0,43	0,12																										
1D_Su_Mik_(lt)	1,00	-0,17	0,04	-0,23	0,87	-0,13	0,04	0,23	-0,24	-0,01	-0,18	0,04	-0,23	0,03																											
1D_Sure_(sn)	1,00	-0,16	0,18	0,03	0,95	-0,16	0,52	0,42	0,05	0,91	-0,16	0,18	-0,21																											
1D_Devir_(rpm)	1,00	0,13	0,06	-0,19	1,00	0,09	0,00	-0,57	-0,15	1,00	0,13	-0,26																											
1D_MHY_ED	1,00	-0,24	0,19	0,13	0,13	0,12	0,07	0,27	0,13	1,00	-0,36																												
2D_Su_Mik_(lt)	1,00	0,00	0,06	0,37	0,08	0,11	-0,02	0,06	-0,24	-0,05																													
2D_Sure_(sn)	1,00	-0,19	0,54	0,28	0,05	0,92	-0,19	0,19	-0,14																														
2D_Devir_(rpm)	1,00	0,09	0,00	-0,57	-0,15	1,00	0,13	-0,26																															
2D_MHY_ED	1,00	0,16	-0,03	0,46	0,09	0,13	-0,19																																
2D_Sikma_(rpm)	1,00	-0,06	0,35	0,00	0,12	-0,39																																	
3D_Su_Mik_(lt)	1,00	0,08	-0,57	0,07	0,26																																		
3D_Sure_(sn)	1,00	-0,15	0,27	-0,18																																			
3D_Devir_(rpm)	1,00	0,13	-0,26																																				
3D_MHY_ED	1,00	-0,36																																					
Cikis_Alan_Ortalama	1,00																																						

SPEARMAN'S RANK

DEĞER	Sol_Korelasyon	to	Üst_Korelasyon
1,00	Deterjan_Miktarı(gr)	=>	Test_Kapasitesi(kg)
1,00	SY1_Devir_(rpm)	=>	AY_Devir_(rpm)
1,00	SY1_Devir_(rpm)	=>	1D_Devir_(rpm)
1,00	SY1_Devir_(rpm)	=>	2D_Devir_(rpm)
1,00	SY1_Devir_(rpm)	=>	3D_Devir_(rpm)
1,00	AY_Devir_(rpm)	=>	1D_Devir_(rpm)
1,00	AY_Devir_(rpm)	=>	2D_Devir_(rpm)
1,00	AY_Devir_(rpm)	=>	3D_Devir_(rpm)
1,00	1D_Devir_(rpm)	=>	2D_Devir_(rpm)
1,00	1D_Devir_(rpm)	=>	3D_Devir_(rpm)
1,00	1D_MHY_ED	=>	3D_MHY_ED
1,00	2D_Devir_(rpm)	=>	3D_Devir_(rpm)
1,00	SY1_ED	=>	1D_MHY_ED
1,00	SY1_ED	=>	3D_MHY_ED
0,97	SY1_ED	=>	AY_MHY_ED
0,97	AY_MHY_ED	=>	1D_MHY_ED
0,97	AY_MHY_ED	=>	3D_MHY_ED
0,95	1D_Sure_(sn)	=>	2D_Sure_(sn)
0,95	I_Sure_(sn)	=>	AY_Tah_(C)
0,94	Numune_1_ Relakse_Sonrasi_ En_1	=>	Numune_1_ Relakse_Sonrasi_ En_2
0,92	2D_Sure_(sn)	=>	3D_Sure_(sn)
0,91	1D_Sure_(sn)	=>	3D_Sure_(sn)
0,90	I_Tset_(C)	=>	AY_Tah_(C)
0,89	I_Sure_(sn)	=>	I_Tset_(C)
0,87	1D_Su_Mik_(lt)	=>	2D_Su_Mik_(lt)
0,86	Numune_1_Relakse_Sonrasi_Boy_1	=>	Numune_1_Relakse_Sonrasi_Boy_3
0,86	Numune_1_ Relakse_Sonrasi_ En_1	=>	Numune_1_ Relakse_Sonrasi_ En_3
0,85	Numune_1_ Relakse_Sonrasi_ En_2	=>	Numune_1_ Relakse_Sonrasi_ En_3
0,83	Numune_1_Relakse_Sonrasi_Boy_1	=>	Numune_1_Relakse_Sonrasi_Boy_2
0,78	Deterjan_Miktarı(gr)	=>	Tambur_Hacmi_(lt)
0,78	Test_Kapasitesi(kg)	=>	Tambur_Hacmi_(lt)
0,74	Numune_1_Relakse_Sonrasi_Boy_2	=>	Numune_1_Relakse_Sonrasi_Boy_3
0,74	Numune_1_Relakse_Sonrasi_Boy_1	=>	Giris_Alan_Ortalama
0,73	I_Bas_Sic (C)	=>	I_Devir_(rpm)
0,71	SY1_Devir_(rpm)	=>	I_Devir_(rpm)
0,71	I_Devir_(rpm)	=>	AY_Devir_(rpm)
0,71	I_Devir_(rpm)	=>	1D_Devir_(rpm)
0,71	I_Devir_(rpm)	=>	2D_Devir_(rpm)
0,71	I_Devir_(rpm)	=>	3D_Devir_(rpm)
0,71	AY_Su_Mik_(lt)	=>	3D_Su_Mik_(lt)
-0,72	Test_No	=>	Cikis_Alan_Ortalama

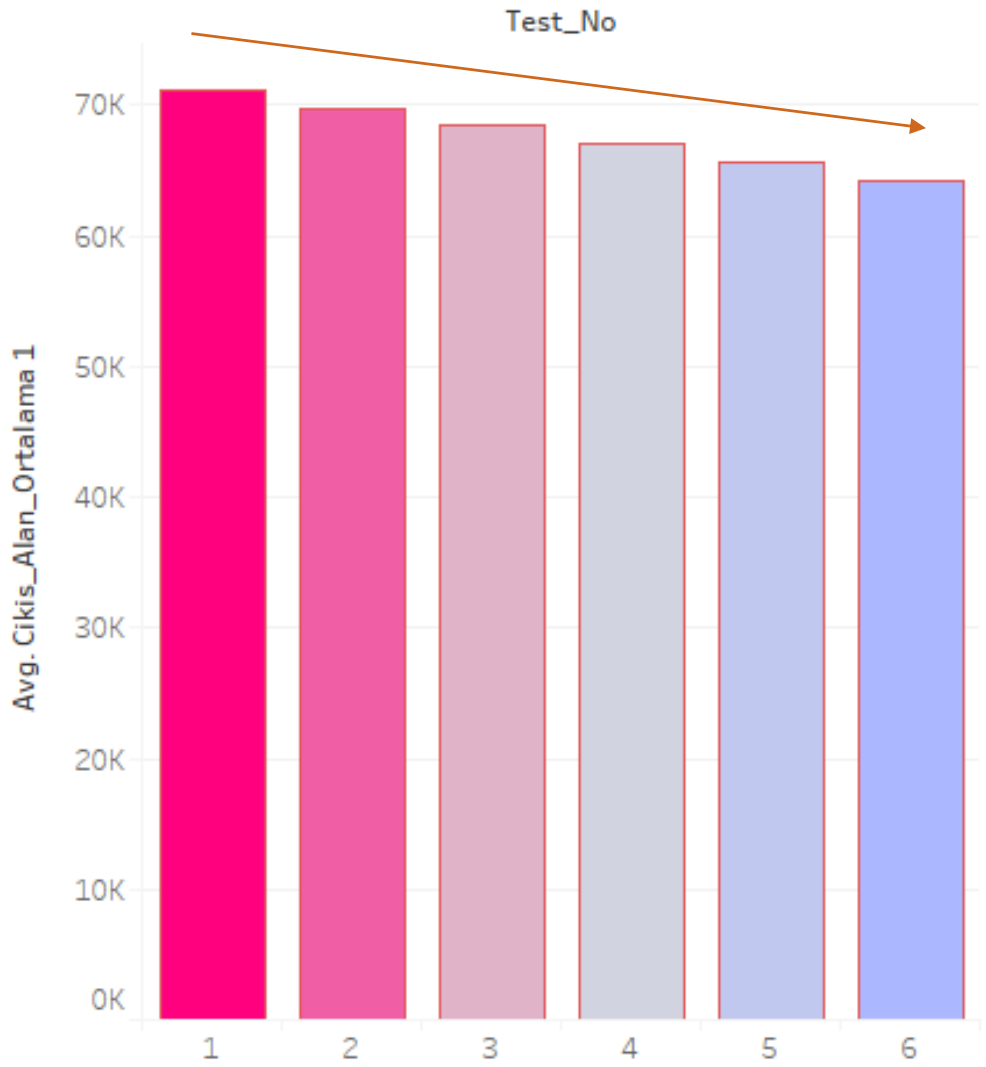
PEARSON CORRELATION W/ OUTPUT

Cikis_Alan_Ortalama	1,00
Numune_1_Relakse_Sonrasi_Boy_3	0,50
Giris_Alan_Ortalama	0,48
Numune_1_Relakse_Sonrasi_Boy_1	0,48
Numune_1_Relakse_Sonrasi_Boy_2	0,48
Test_Kapasitesi(kg)	0,34
Deterjan_Miktarı(gr)	0,34
3D_Su_Mik_(lt)	0,25
SY1_Sure_(sn)	0,22
AY_Sure_(sn)	0,21
1D_Su_Mik_(lt)	0,17
AY_Su_Mik_(lt)	0,16
Jet_Pompa	0,12
2D_Su_Mik_(lt)	0,11
Su_Sertligi	0,09
Tambur_Hacmi_(lt)	0,07
2D_MHY_ED	-0,01
I_MHY_ED	-0,07
Numune_1_ Relakse_Sonrasi_ En_2	-0,09
3D_Sure_(sn)	-0,11
2D_Sure_(sn)	-0,13
SG_Sic_(C)	-0,15
1D_Sure_(sn)	-0,15
Numune_1_ Relakse_Sonrasi_ En_3	-0,18
Numune_1_ Relakse_Sonrasi_ En_1	-0,20
I_Sure_(sn)	-0,28
I_Tset_(C)	-0,31
3D_Devir_(rpm)	-0,33
SY1_Devir_(rpm)	-0,33
AY_Devir_(rpm)	-0,33
1D_Devir_(rpm)	-0,33
2D_Devir_(rpm)	-0,33
AY_Tah_(C)	-0,34
I_Devir_(rpm)	-0,35
I_Bas_Sic (C)	-0,36
1D_MHY_ED	-0,40
SY1_ED	-0,40
3D_MHY_ED	-0,40
AY_MHY_ED	-0,41
2D_Sikma_(rpm)	-0,49
Test_No	-0,67

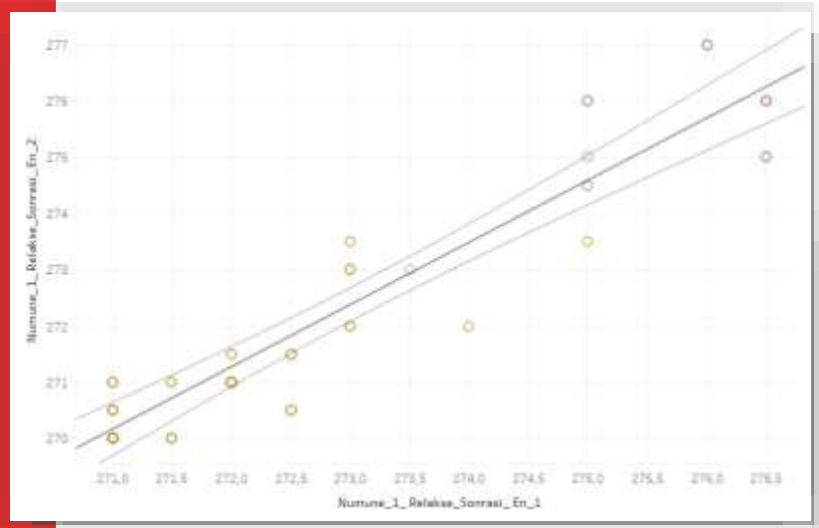
SPEARMAN CORRELATION W/ OUTPUT

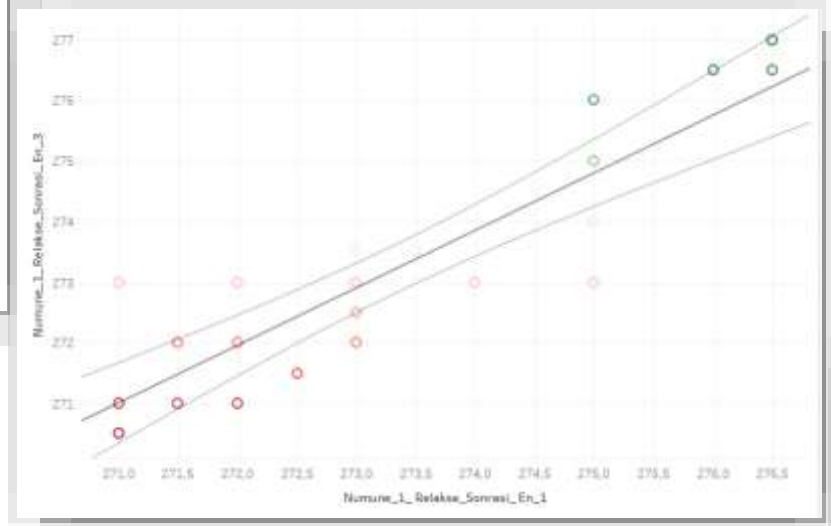
1,00	Cikis_Alan_Ortalama
0,28	Deterjan_Miktarı(gr)
0,28	Test_Kapasitesi(kg)
0,26	3D_Su_Mik_(lt)
0,24	Giris_Alan_Ortalama
0,21	Numune_1_Relakse_Sonrasi_Boy_3
0,18	Numune_1_Relakse_Sonrasi_Boy_1
0,17	SY1_Sure_(sn)
0,16	AY_Su_Mik_(lt)
0,15	Numune_1_Relakse_Sonrasi_Boy_2
0,12	AY_Sure_(sn)
0,12	Jet_Pompa
0,06	Su_Sertligi
0,03	1D_Su_Mik_(lt)
-0,01	Tambur_Hacmi_(lt)
-0,05	2D_Su_Mik_(lt)
-0,06	Numune_1_ Relakse_Sonrasi_ En_2
-0,11	Numune_1_ Relakse_Sonrasi_ En_3
-0,13	Numune_1_ Relakse_Sonrasi_ En_1
-0,14	2D_Sure_(sn)
-0,14	SG_Sic_(C)
-0,18	3D_Sure_(sn)
-0,19	2D_MHY_ED
-0,21	1D_Sure_(sn)
-0,21	I_Sure_(sn)
-0,25	I_Tset_(C)
-0,26	SY1_Devir_(rpm)
-0,26	AY_Devir_(rpm)
-0,26	1D_Devir_(rpm)
-0,26	2D_Devir_(rpm)
-0,26	3D_Devir_(rpm)
-0,29	AY_Tah_(C)
-0,31	I_Devir_(rpm)
-0,36	SY1_ED
-0,36	1D_MHY_ED
-0,36	3D_MHY_ED
-0,37	I_MHY_ED
-0,38	I_Bas_Sic (C)
-0,38	AY_MHY_ED
-0,39	2D_Sikma_(rpm)
-0,72	Test_No

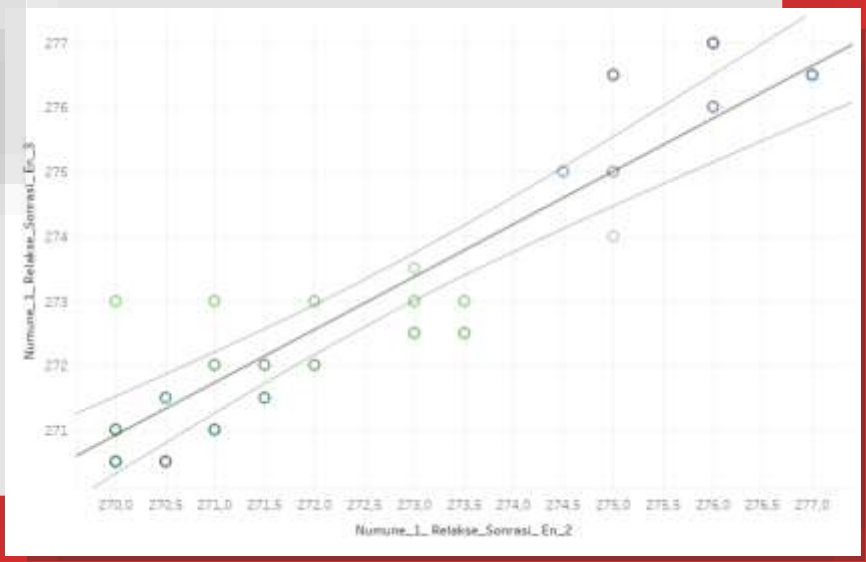
DEĞER	Sol_Korelasyon		Üst_Korelasyon
-0,72	Test_No	=>	Cikis_Alan_Ortalama

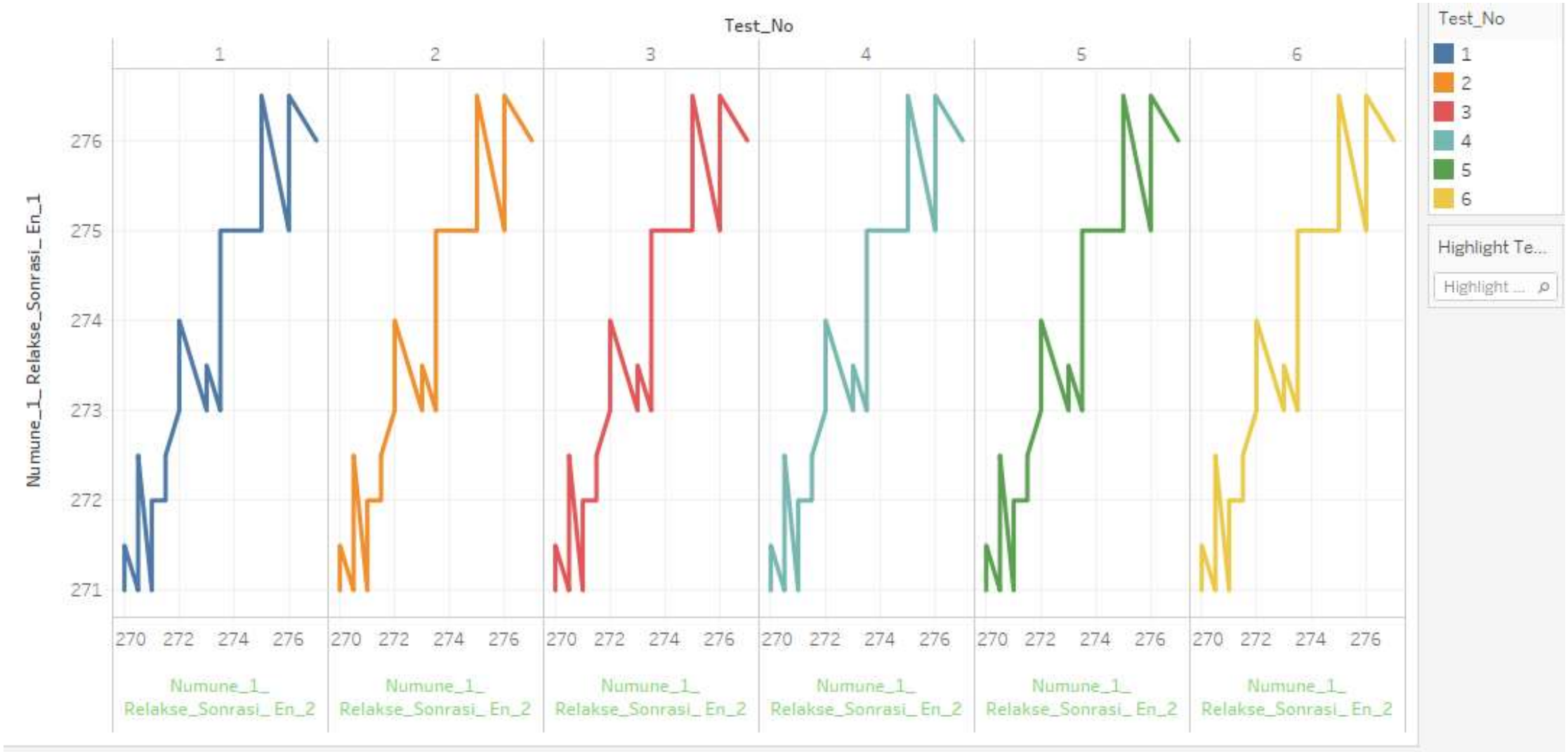


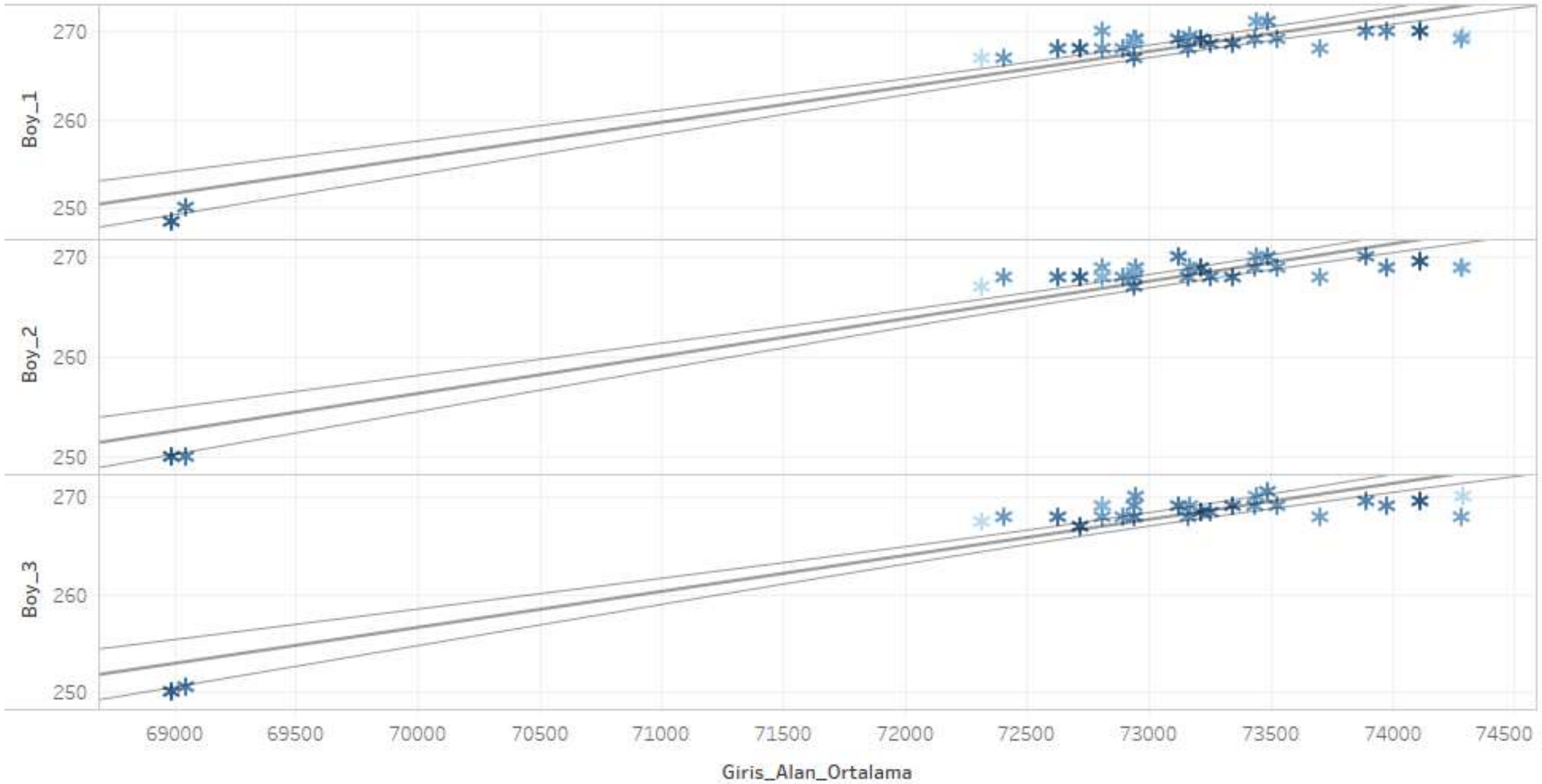
DEĞER	Sol_Korelasyon	to	Üst_Korelasyon
0,94	Numune_1_ Relakse_Sonrasi_ En_1	=>	Numune_1_ Relakse_Sonrasi_ En_2
0,86	Numune_1_ Relakse_Sonrasi_ En_1	=>	Numune_1_ Relakse_Sonrasi_ En_3
0,85	Numune_1_ Relakse_Sonrasi_ En_2	=>	Numune_1_ Relakse_Sonrasi_ En_3



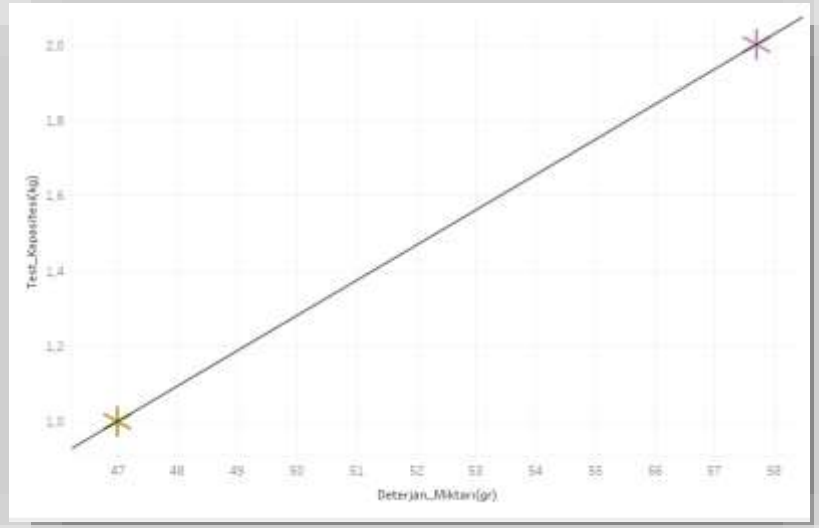


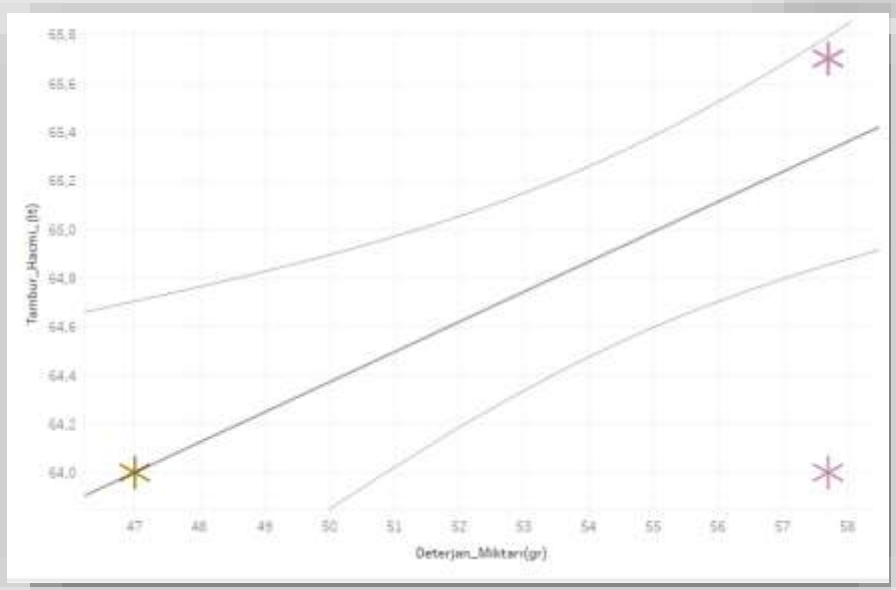



Test_No
1
2
3
4
5
6
Numune_1_ Relakse_Sonrasi_ En_1
276
275
274
273
272
271
270
272
274
276
Numune_1_ Relakse_Sonrasi_ En_2
Test_No
1
2
3
4
5
6
Highlight Te...
Highlight ...

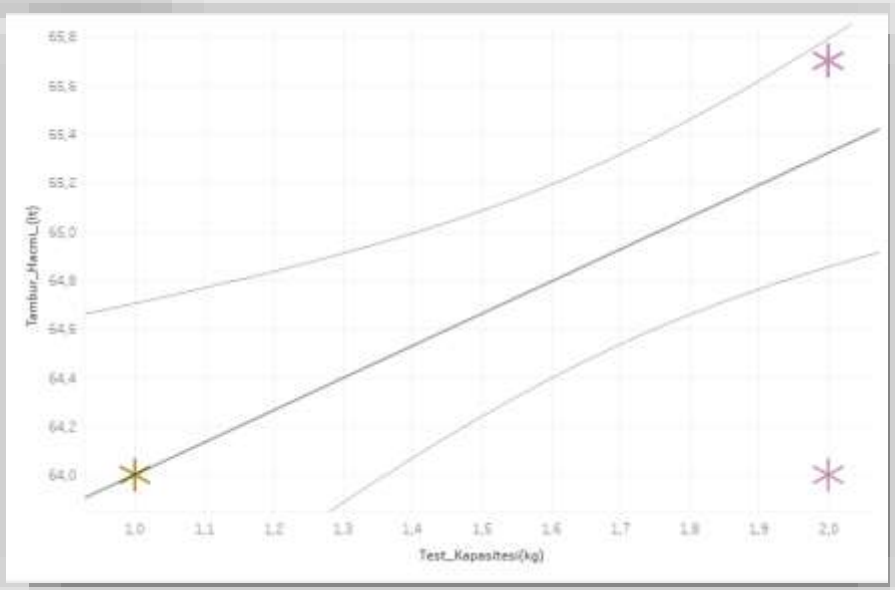

Boy_1
Boy_2
Boy_3
270
260
250
69000
69500
70000
70500
71000
71500
72000
72500
73000
73500
74000
74500
Giris_Alan_Ortalama

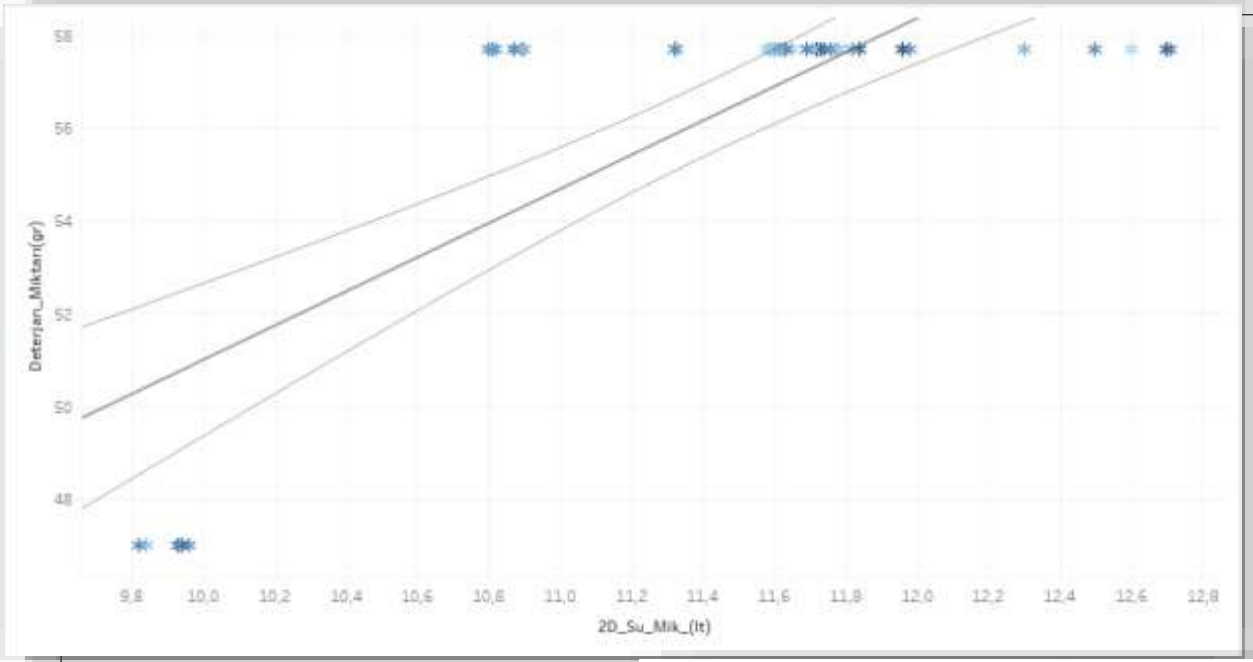
1,00	Deterjan_Miktarı(gr)	=>	Test_Kapasitesi(kg)
0,78	Deterjan_Miktarı(gr)	=>	Tambur_Hacmi_(lt)
0,78	Test_Kapasitesi(kg)	=>	Tambur_Hacmi_(lt)

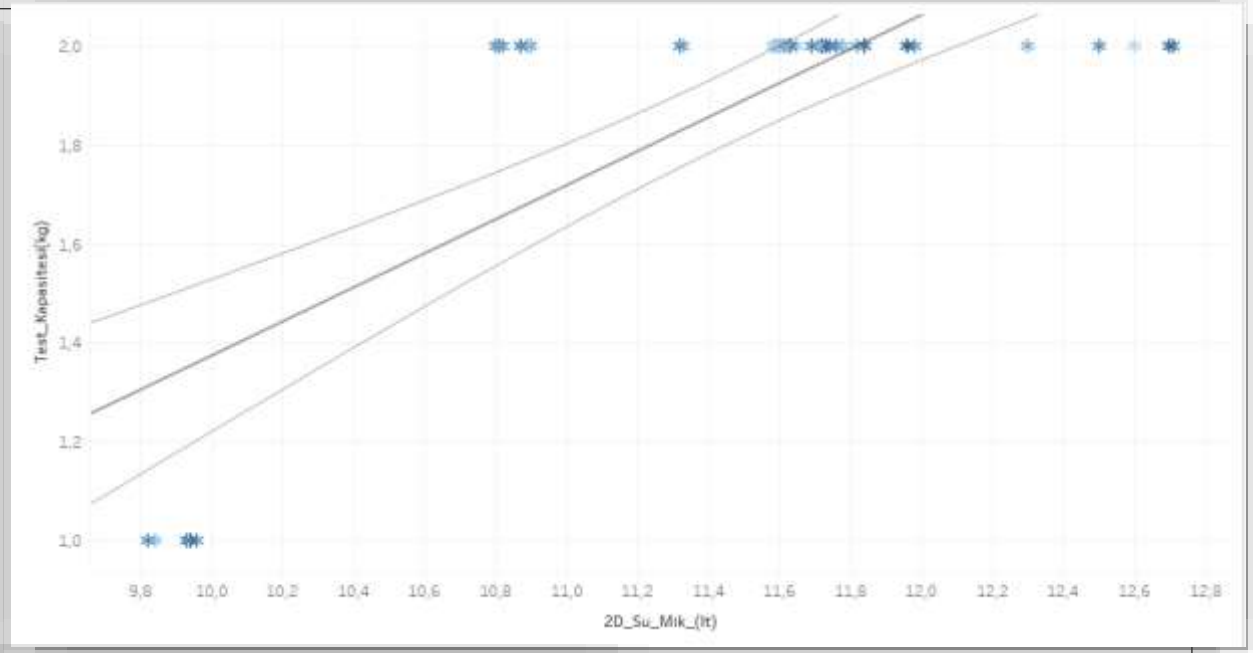


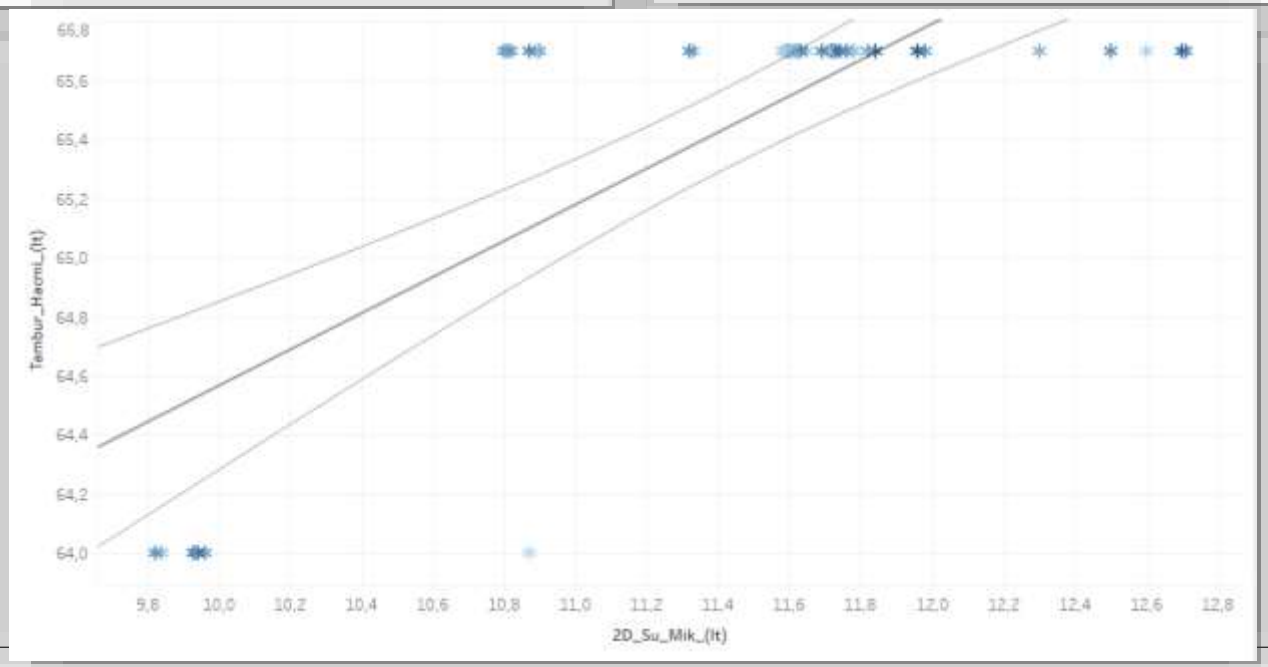


=

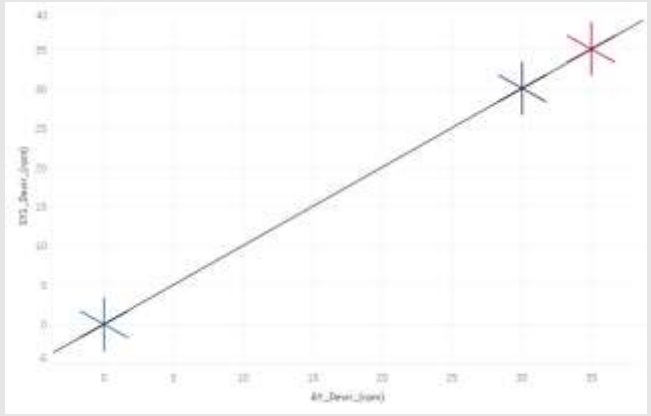


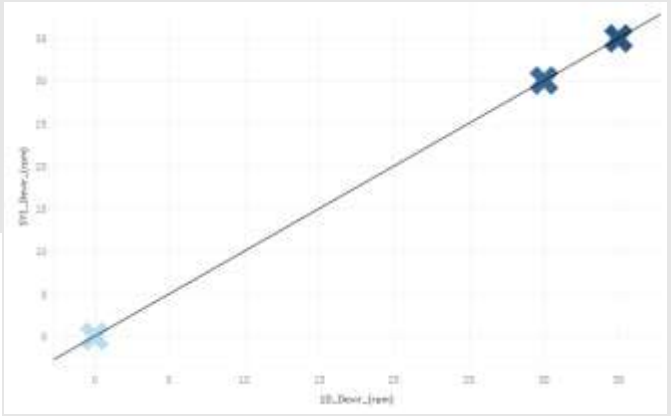

Deterjan_Miktarı(gr)
58
56
54
52
50
48
9,8
10,0
10,2
10,4
10,6
10,8
11,0
11,2
11,4
11,6
11,8
12,0
12,2
12,4
12,6
12,8
2D_Su_Mik_(lt)

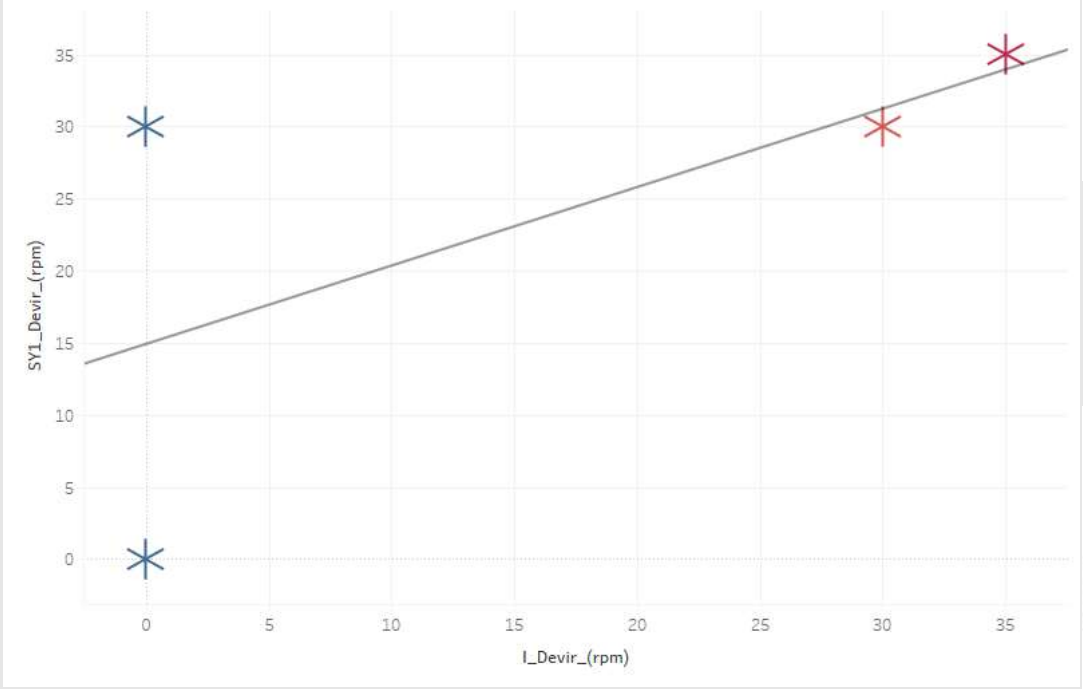

Test_Kapasitesi(kg)
2,0
1,8
1,6
1,4
1,2
1,0
9,8
10,0
10,2
10,4
10,6
10,8
11,0
11,2
11,4
11,6
11,8
12,0
12,2
12,4
12,6
12,8
2D_Su_Mik_(lt)

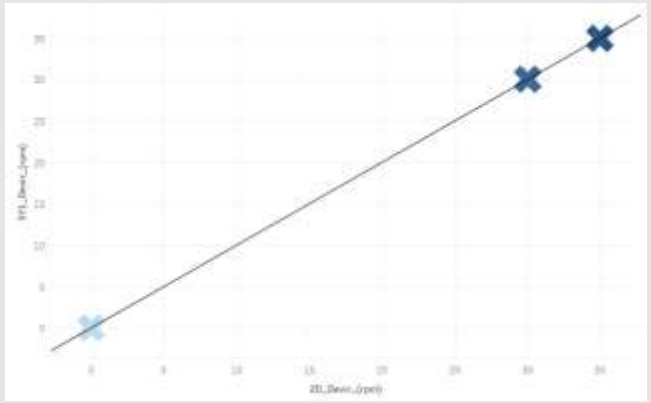

Tambur_Hacmi_(lt)
65,8
65,6
65,4
65,2
65,0
64,8
64,6
64,4
64,2
64,0
9,8
10,0
10,2
10,4
10,6
10,8
11,0
11,2
11,4
11,6
11,8
12,0
12,2
12,4
12,6
12,8
2D_Su_Mik_(lt)

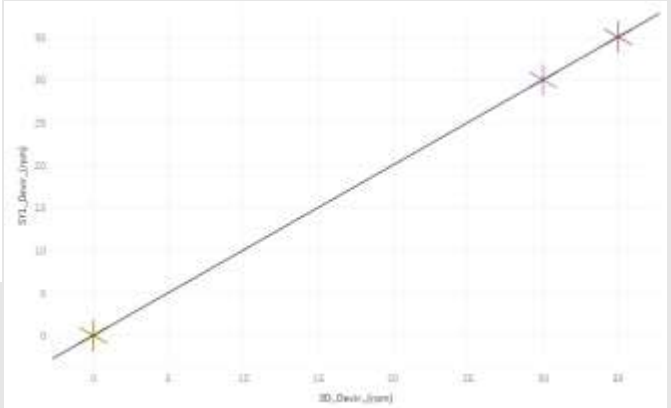
0,71	SY1_Devir_(rpm)	=>	I_Devir_(rpm)
1,00	SY1_Devir_(rpm)	=>	AY_Devir_(rpm)
1,00	SY1_Devir_(rpm)	=>	1D_Devir_(rpm)
1,00	SY1_Devir_(rpm)	=>	2D_Devir_(rpm)
1,00	SY1_Devir_(rpm)	=>	3D_Devir_(rpm)





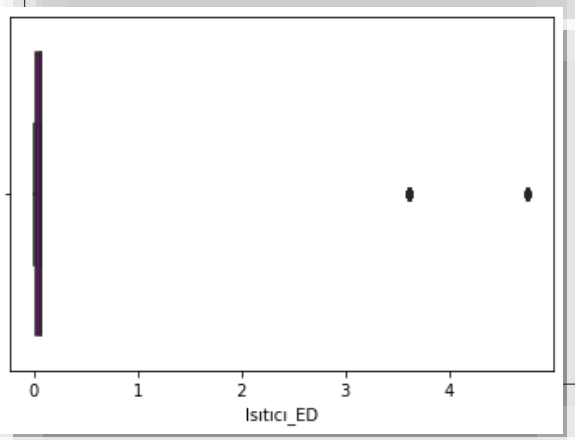


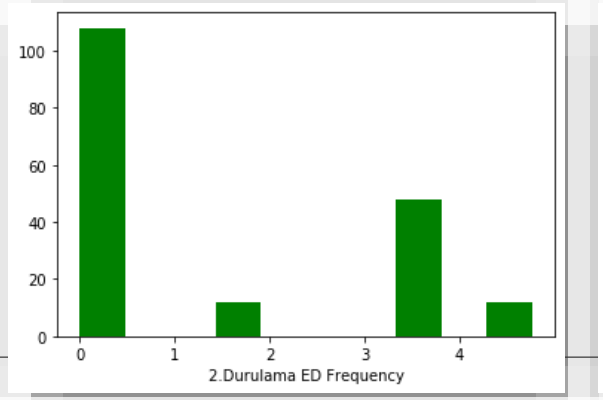


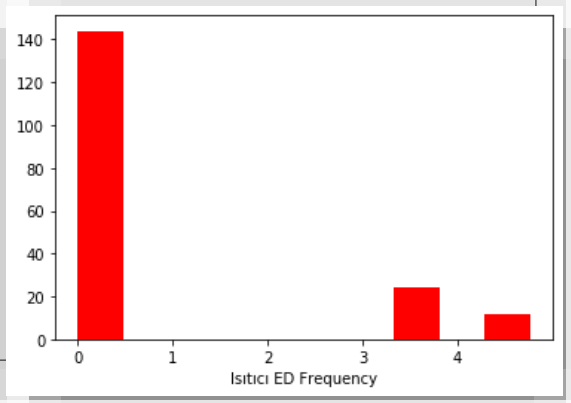


Revolution and Engine Motion Analysis (rpm)

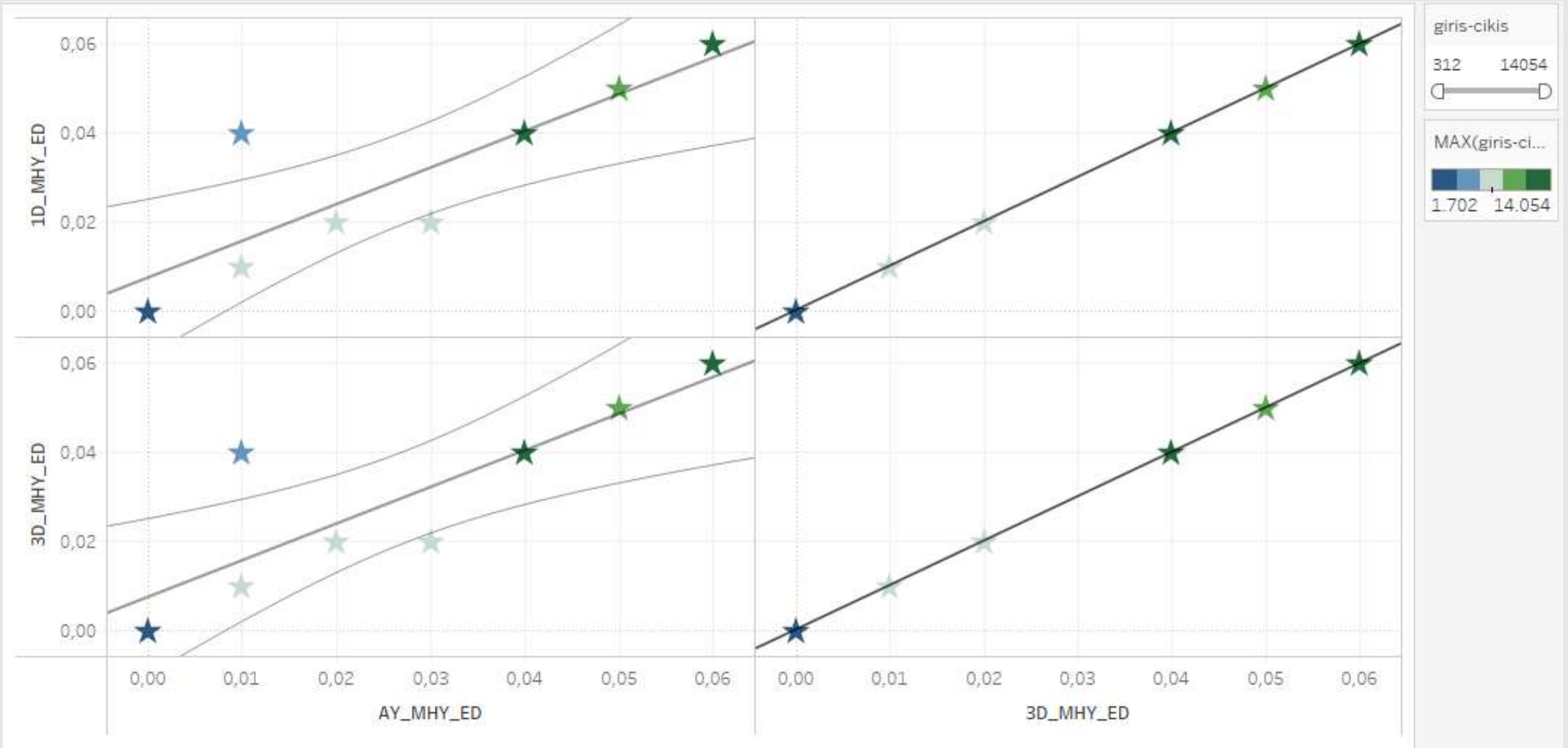
- All cycle data are batch data. According to the Spearman correlation, if Isıtıcı_devir (heater speed) increases, the shrinkage amount increases by 31% !!
- In addition, among the ED variables, Isıtıcı_ED variable contains outliers, the distribution of other ED variables is seen regularly.
- Of the ED variables, AY_ED (main wash) has the most relation with CAO. If AY_ED increases, it can be interpreted that the shrinkage rate in woolen textile increases 38%. AY is also 97% same with 1D, 3D and SY values.
- SY_ED = 1D_ED = 3D_ED THESE MUST BE REMOVED!!!!
- Also Isıtıcı_ED has 53% corr with 2D_ED, with Isıtıcı_devir has 51%.



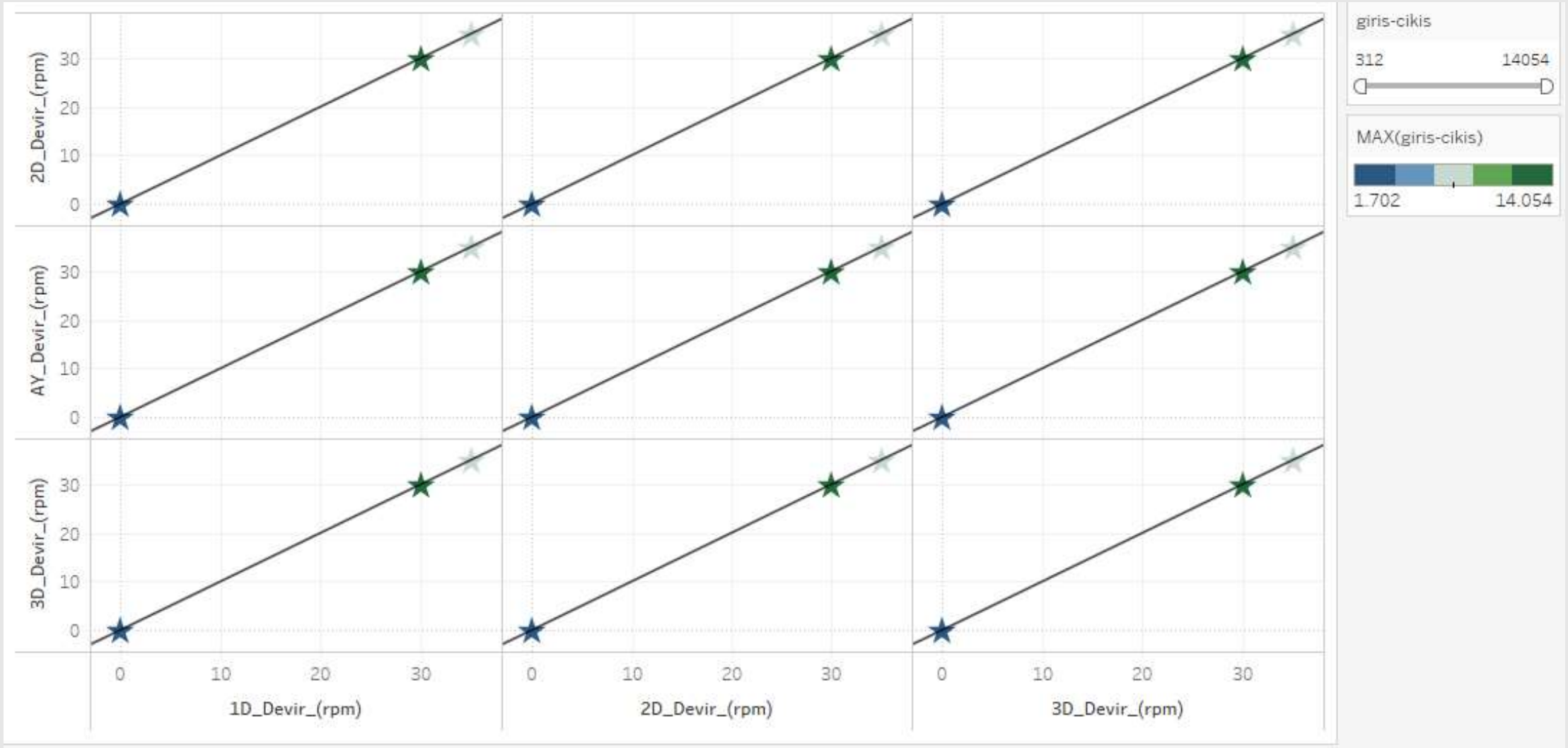




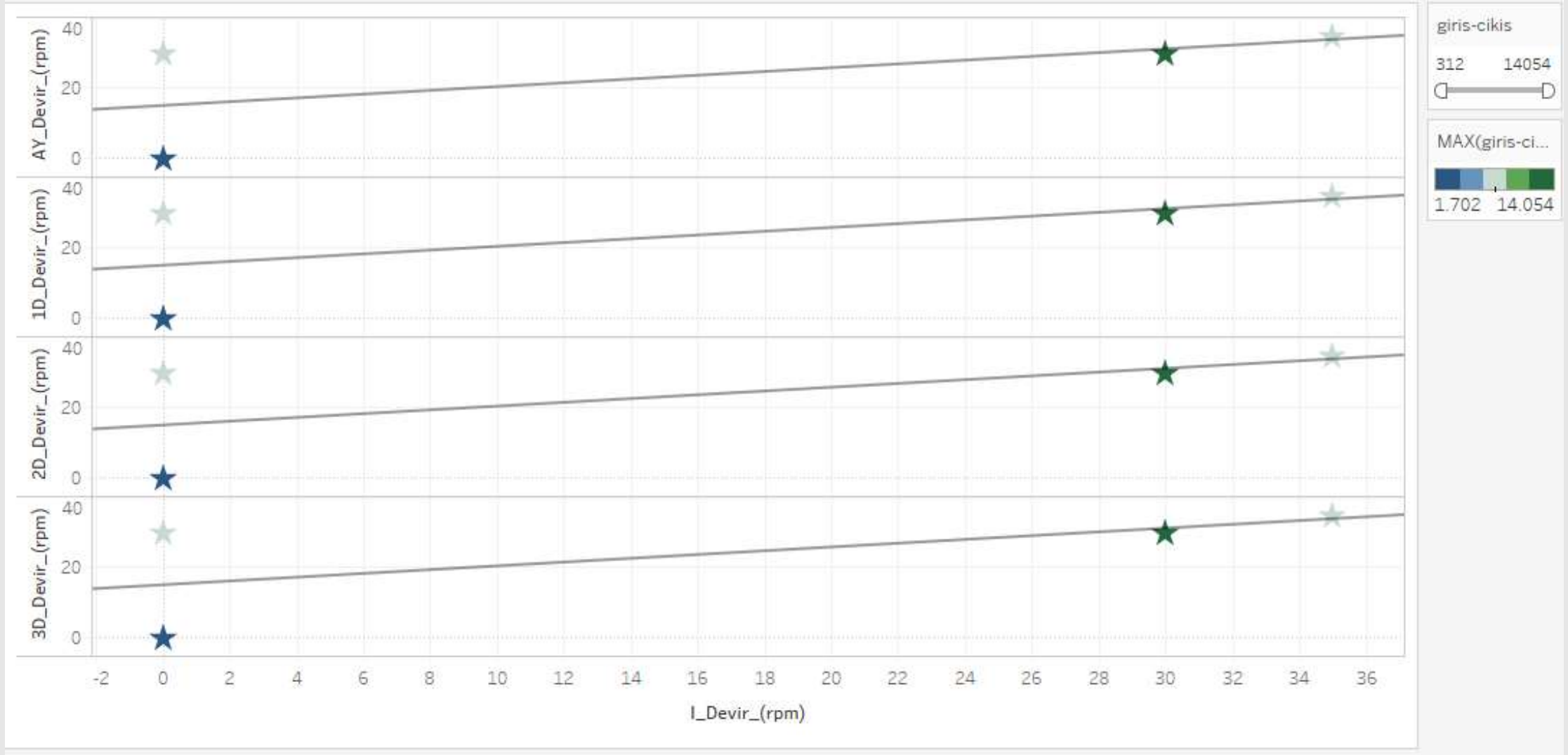
0,96 AY_MHY_ED	=>	1D_MHY_ED
0,96 AY_MHY_ED	=>	3D_MHY_ED
1,00 1D_MHY_ED	=>	3D_MHY_ED

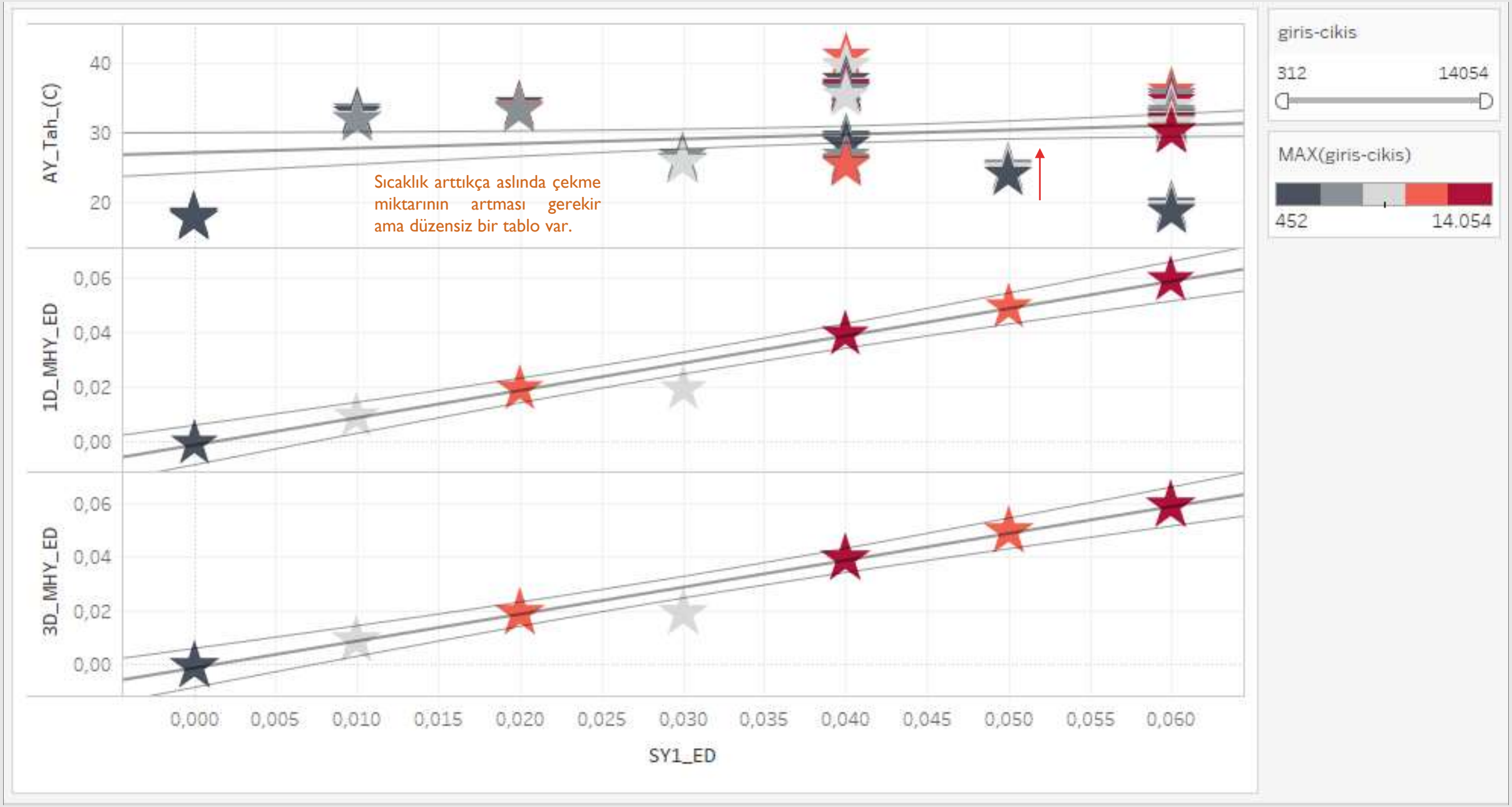


1,00	AY_Devir_(rpm)	=>	1D_Devir_(rpm)
1,00	AY_Devir_(rpm)	=>	2D_Devir_(rpm)
1,00	AY_Devir_(rpm)	=>	3D_Devir_(rpm)
1,00	1D_Devir_(rpm)	=>	2D_Devir_(rpm)
1,00	1D_Devir_(rpm)	=>	3D_Devir_(rpm)
1,00	2D_Devir_(rpm)	=>	3D_Devir_(rpm)

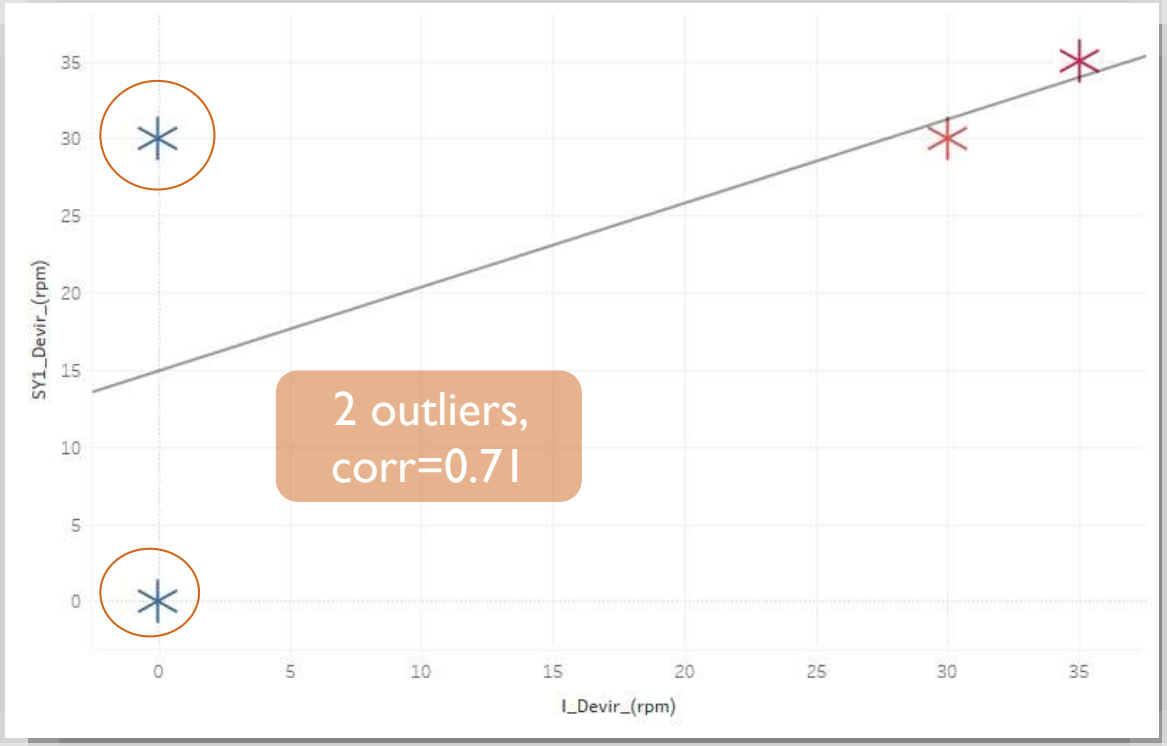


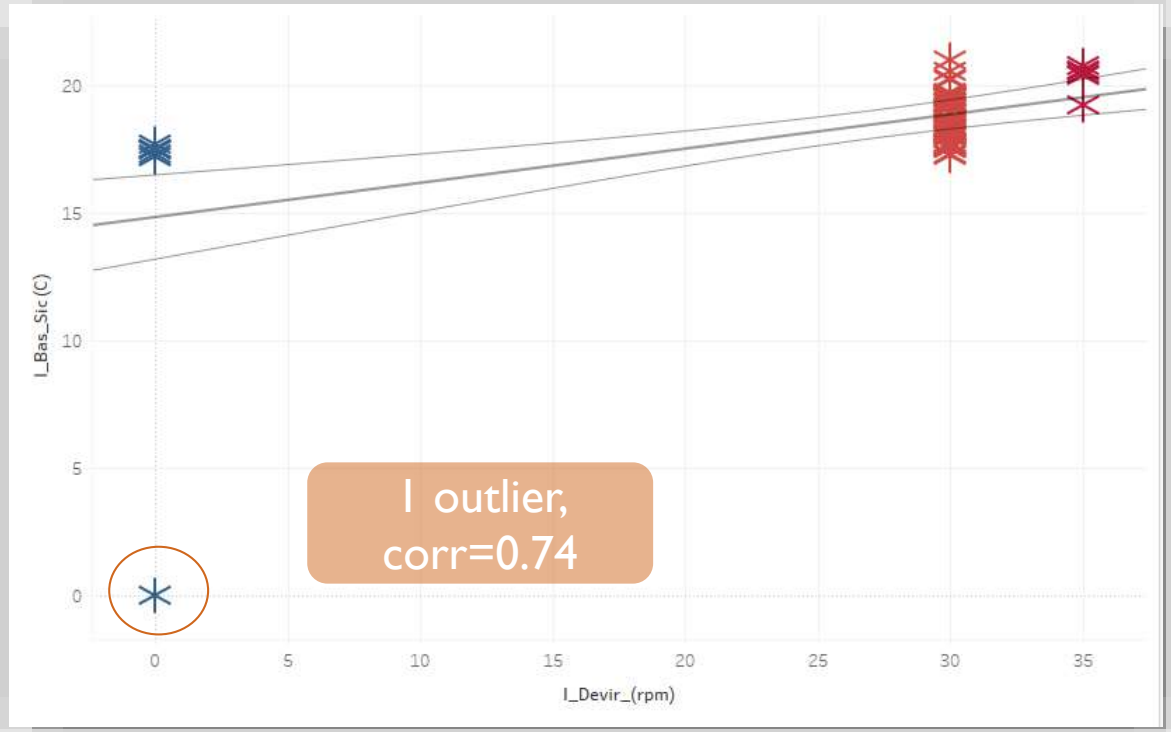
0,71 I_Devir_(rpm)	=>	AY_Devir_(rpm)
0,71 I_Devir_(rpm)	=>	1D_Devir_(rpm)
0,71 I_Devir_(rpm)	=>	2D_Devir_(rpm)
0,71 I_Devir_(rpm)	=>	3D_Devir_(rpm)



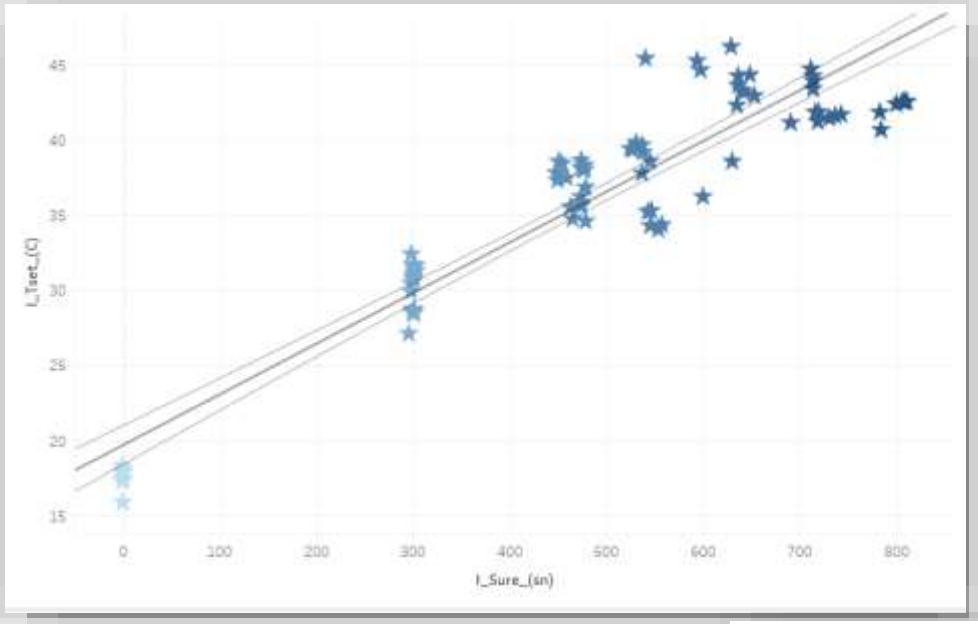

AY_Tah_(C)
40
30
20
Sıcaklık arttıkça aslında çekme miktarının artması gerekir ama düzensiz bir tablo var.
1D_MHY_ED
0,06
0,04
0,02
0,00
3D_MHY_ED
0,06
0,04
0,02
0,00
0,000
0,005
0,010
0,015
0,020
0,025
0,030
0,035
0,040
0,045
0,050
0,055
0,060
SY1_ED
giris-cikis
312
14054
MAX(giris-cikis)
452
14.054

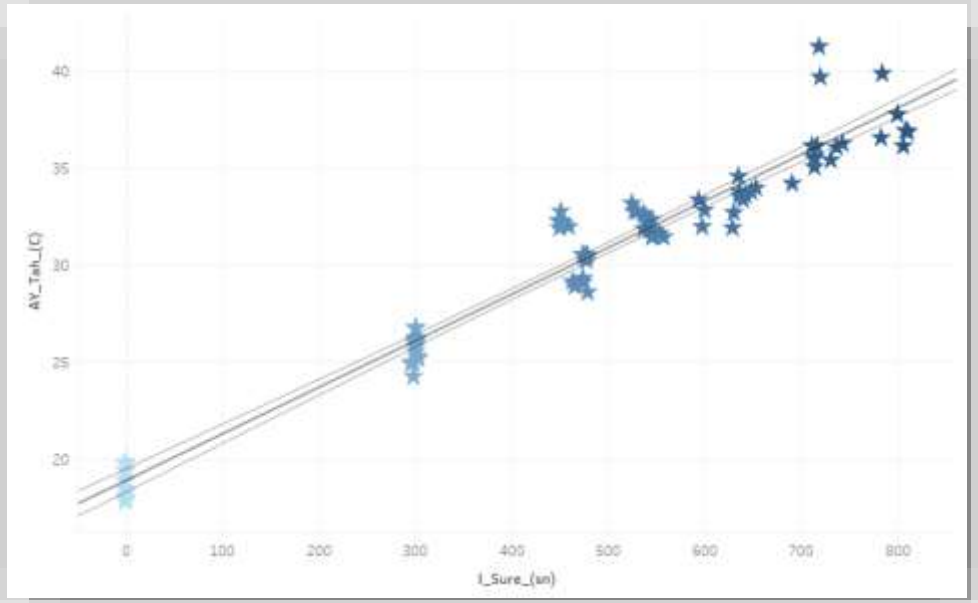
0,71	SY1_Devir_(rpm)	=>	I_Devir_(rpm)
0,74	I_Bas_Sic (C)	=>	I_Devir_(rpm)

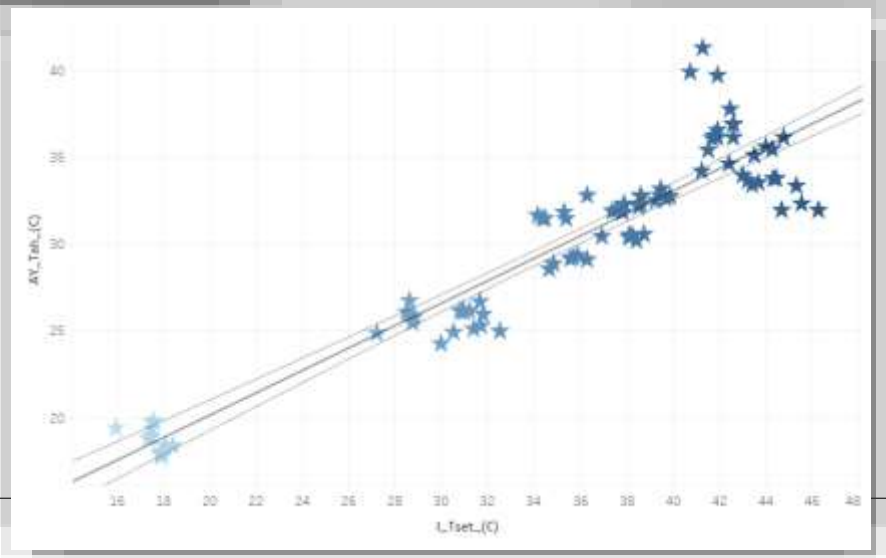




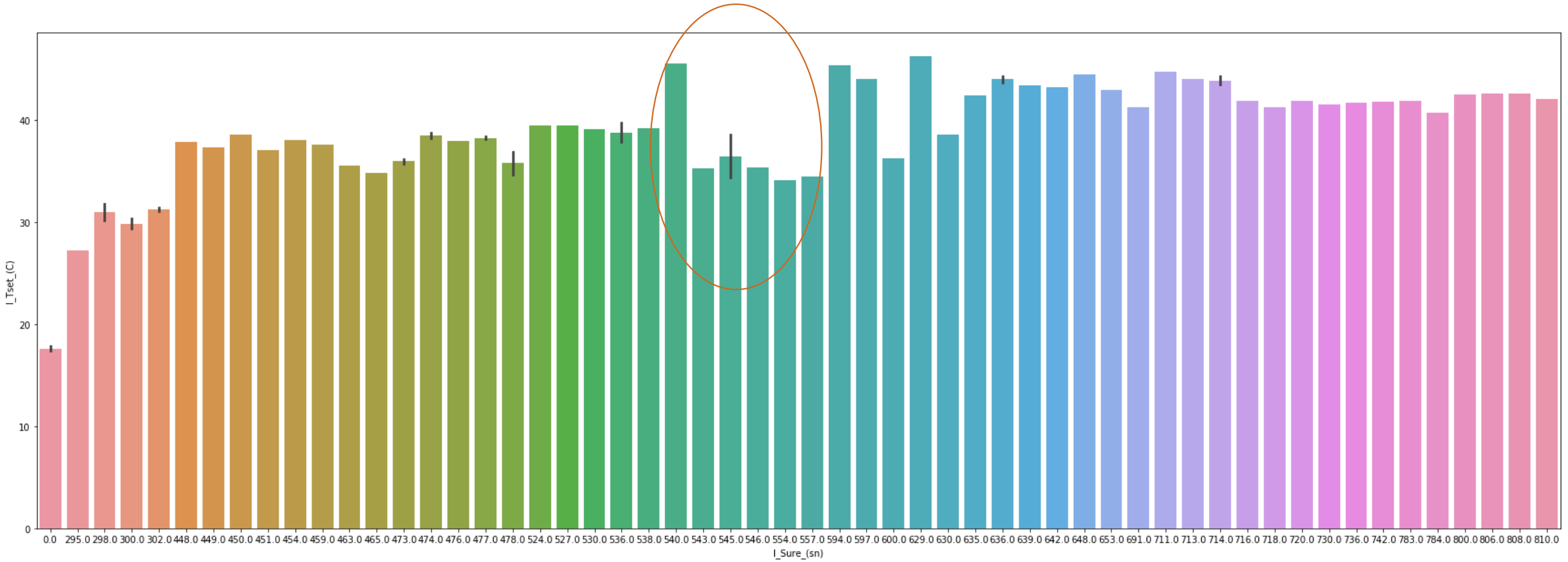
0,89	I_Sure_(sn)	=>	I_Tset_(C)
0,95	I_Sure_(sn)	=>	AY_Tah_(C)
0,90	I_Tset_(C)	=>	AY_Tah_(C)

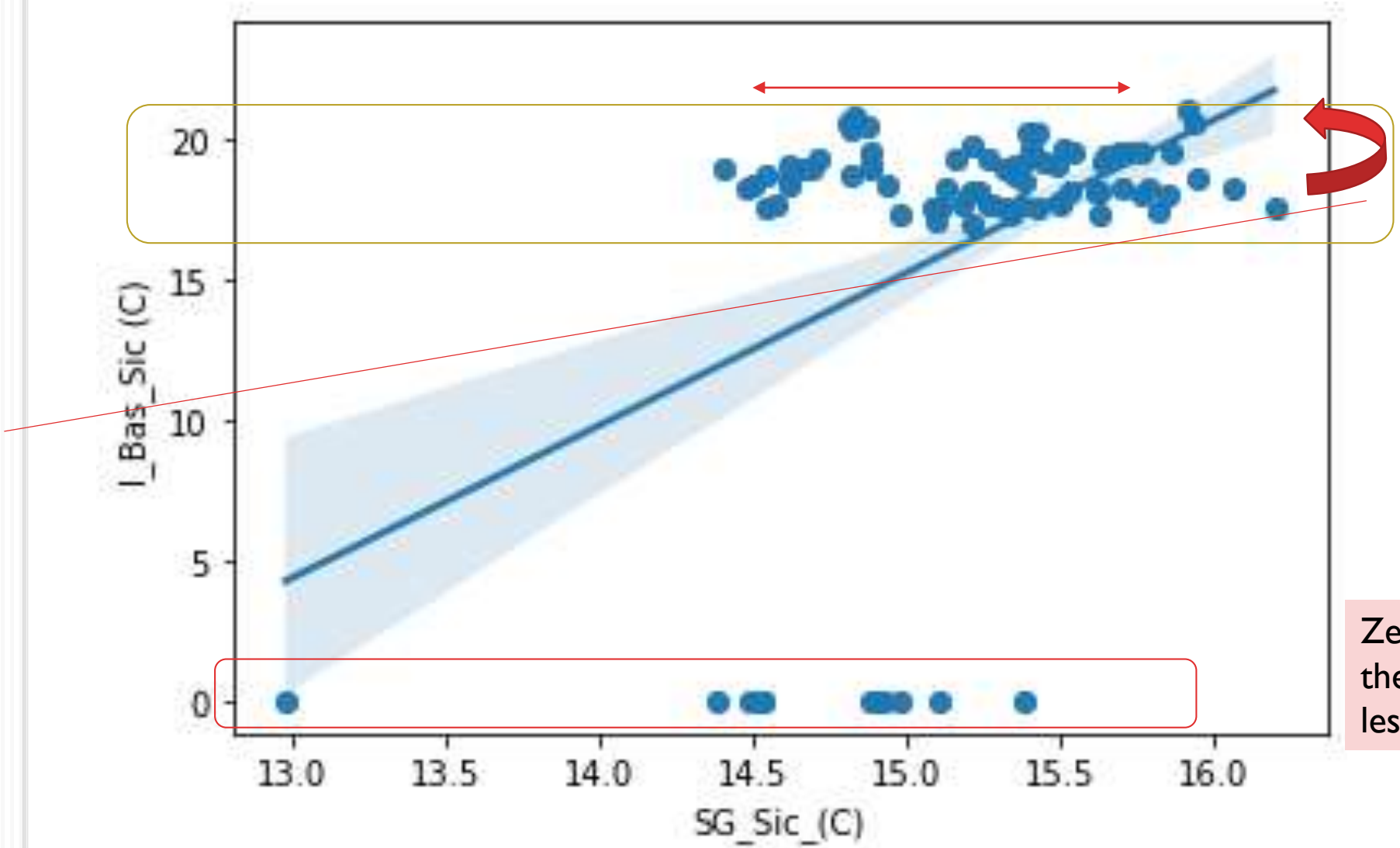






There is a decrease in here, why??

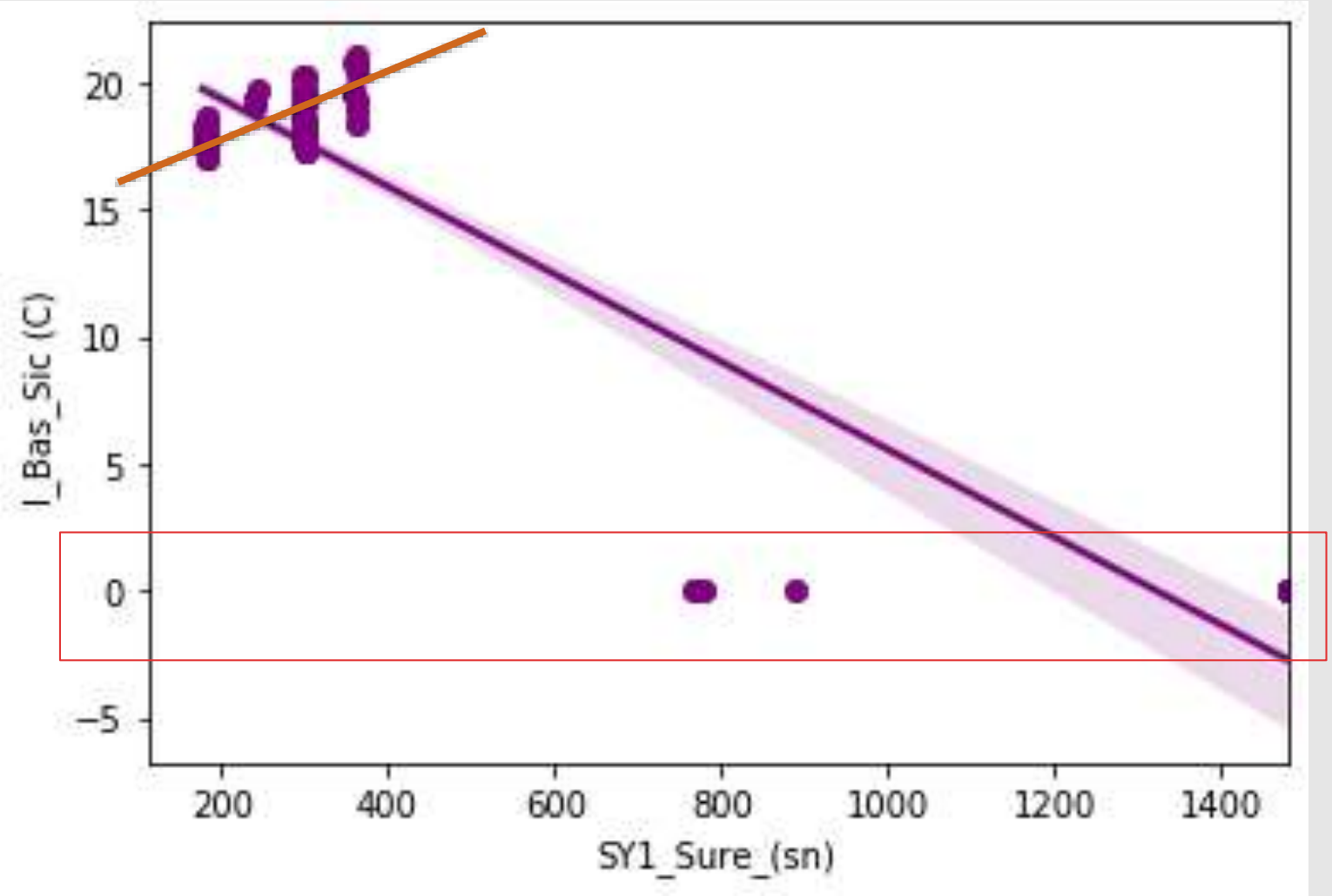




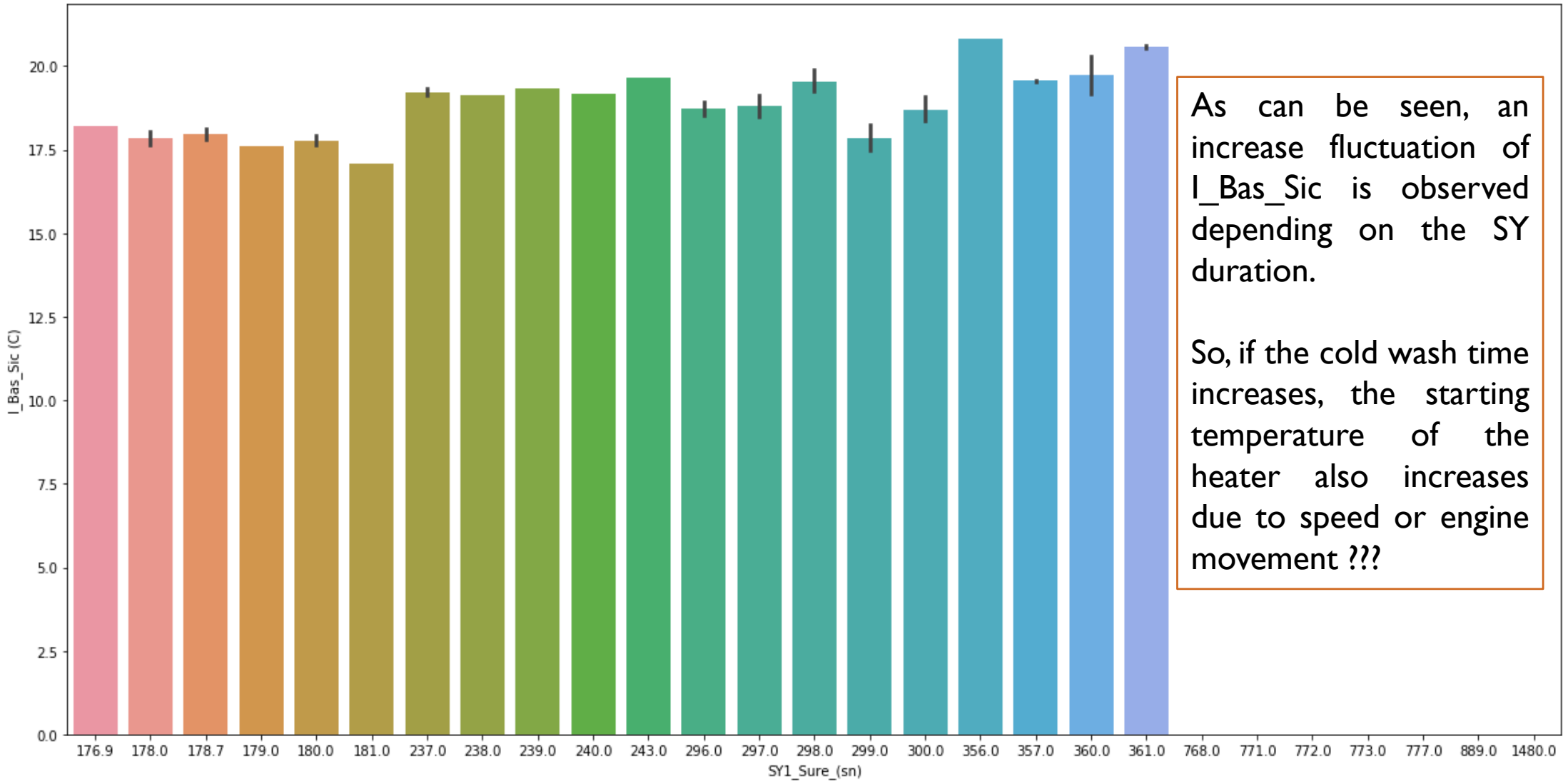
The amount of temperature in the first cold wash increases at the end of the cold wash time. Let's see if the reason for this could be SY_sure_(sn) …

Zero values are a lot, in IQR, they've caused outliers that less than lower bound.

It should have been like this line..

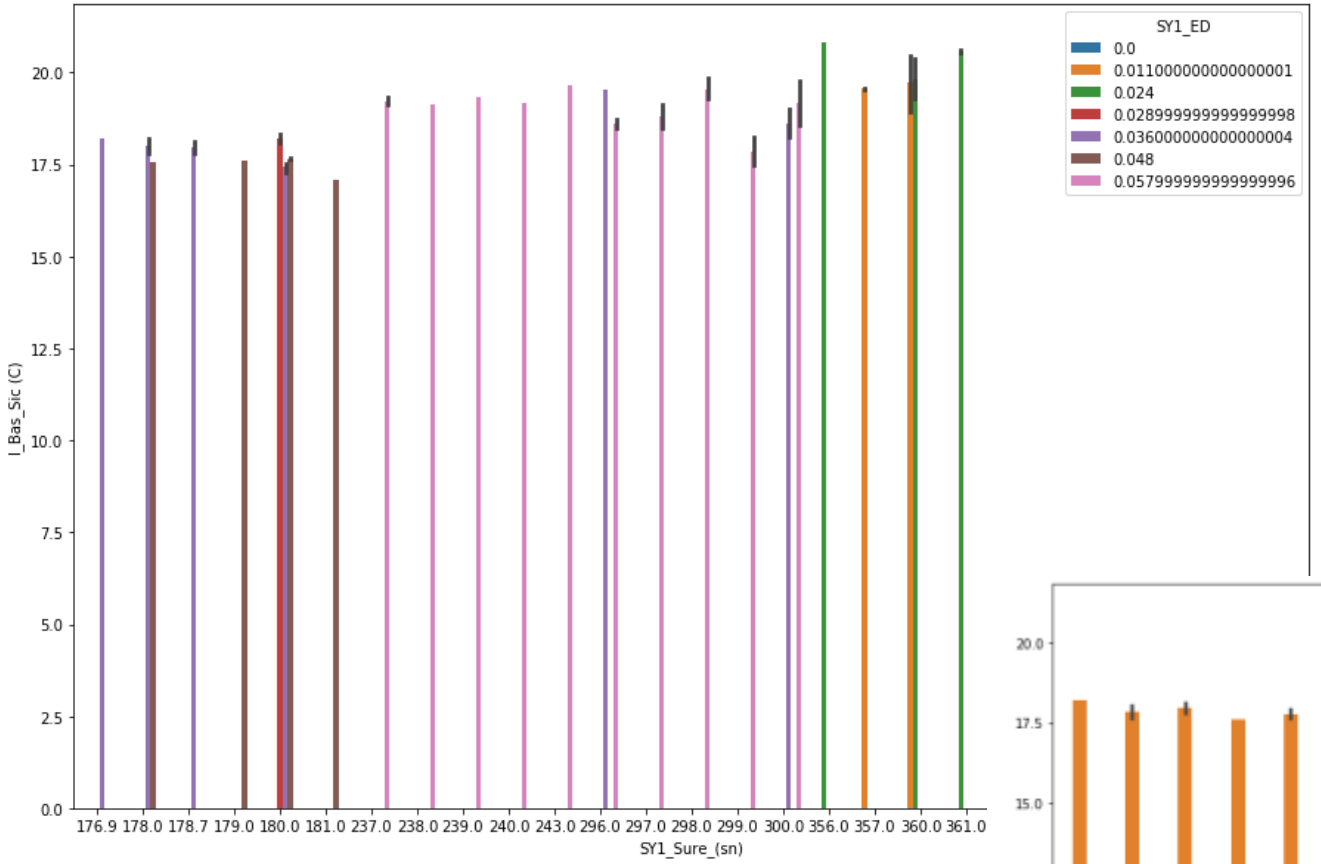


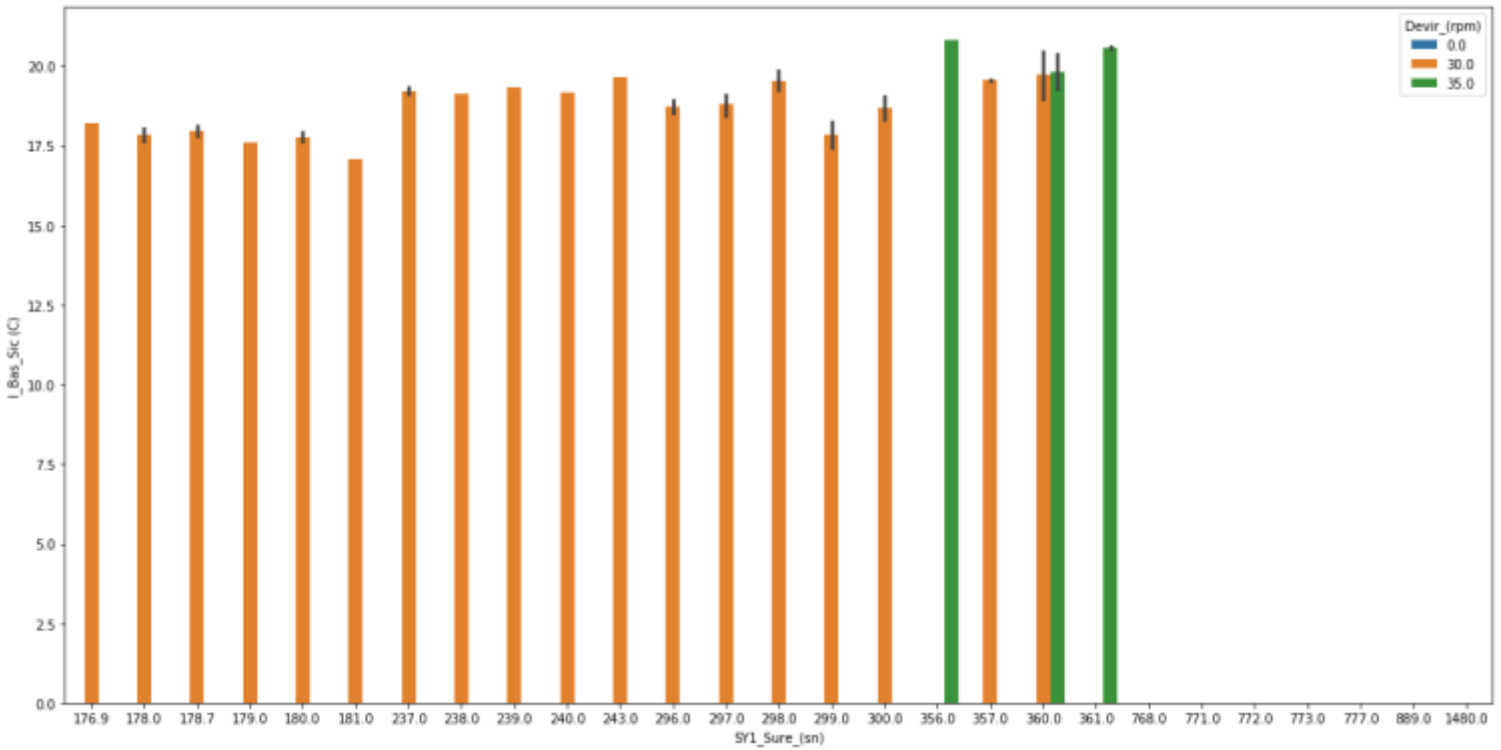
Due to zero outliers, there is a negative correlation between I_Bas_Sic (C) and SY1_Sure_(sn). We have to look the detail info for the best interpretation on the next sheet.

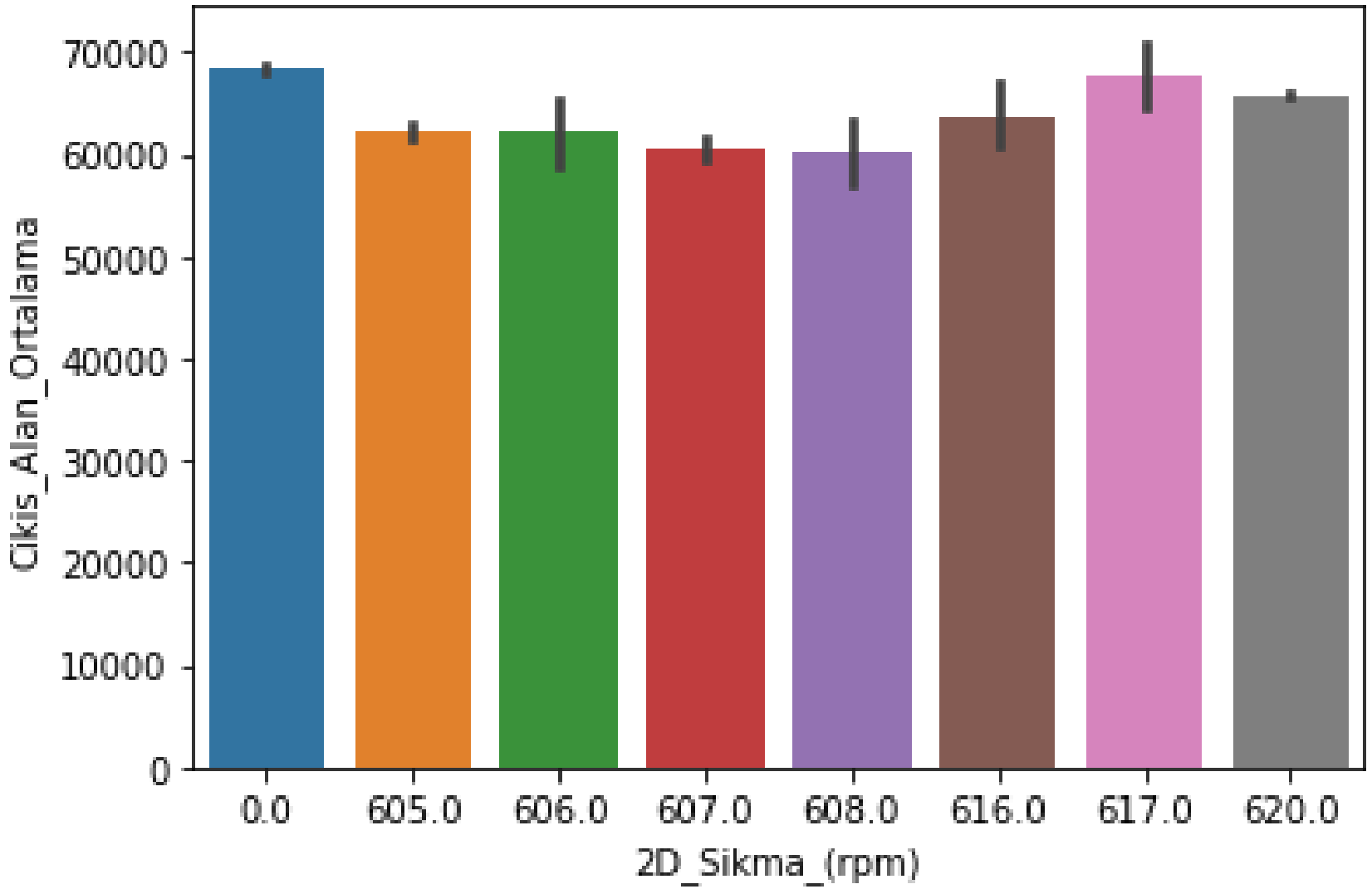


As can be seen, an increase fluctuation of I_Bas_Sic is observed depending on the SY duration.

So, if the cold wash time increases, the starting temperature of the heater also increases due to speed or engine movement ???


SY1_ED
0.0
0.011000000000000001
0.024
0.028999999999999998
0.036000000000000004
0.048
0.057999999999999996
I_Bas_Sic (C)
SY1_Sure_(sn)
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
176.9 178.0 178.7 179.0 180.0 181.0 237.0 238.0 239.0 240.0 243.0 296.0 297.0 298.0 299.0 300.0 356.0 357.0 360.0 361.0

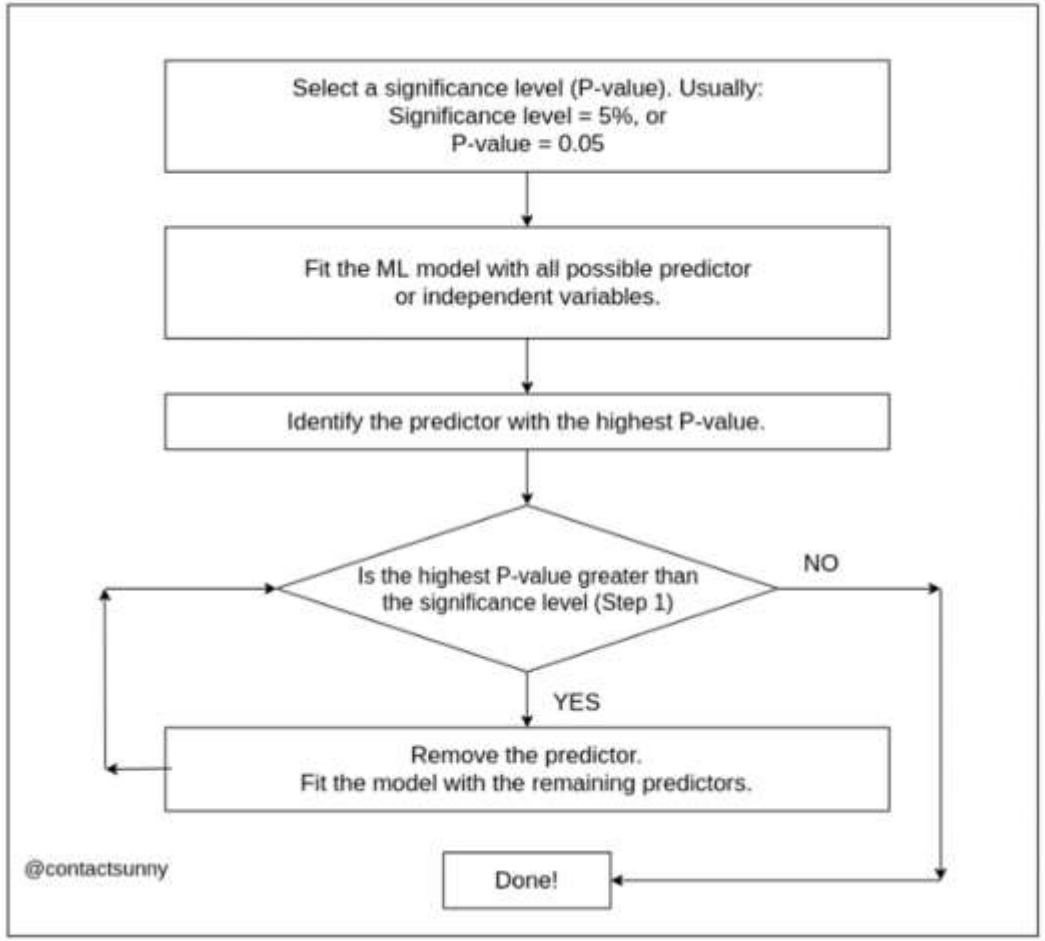

Devir_(rpm)
0.0
30.0
35.0
I_Bas_Sic (C)
SY1_Sure_(sn)
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
176.9 178.0 178.7 179.0 180.0 181.0 237.0 238.0 239.0 240.0 243.0 296.0 297.0 298.0 299.0 300.0 356.0 357.0 360.0 361.0 768.0 771.0 772.0 773.0 777.0 889.0 1480.0



The amount which is the most intense with the shrinkage value, is 608.
0 is too high, when the spin increase, shouldn't the CAO be lower ??

BACKWARD ELIMINATION APPROACH

OLS REGRESSION



If the P value found in a test is below 0.05,
it means that there is a significant difference in comparison.

P değeri	Yorumu
0.01<=p<0.05	İstatistiksel anlamlılık
0.001<=p<0.01	Yüksek düzeyde istatistiksel anlamlılık
p<0.001	Çok yüksek istatistiksel anlamlılık
0.05<=p<0.10	Anlamlılık eğilimi(sınırda anlamlılık)
p>0.10	Fark tesadüften ileri gelmiştir (istatistiksel olarak anlamlı farklılık saptanmamıştır)

	coef	std err	t	P>|t|	[0.025	0.975]
x1	-1411.7982	45.312	-31.157	0.000	-1501.361	-1322.235
x2	-1.55e+04	3844.863	-4.030	0.000	-2.31e+04	-7896.867
x3	-1.614e+04	3917.947	-4.119	0.000	-2.39e+04	-8393.571
x4	-1.559e+04	3798.370	-4.103	0.000	-2.31e+04	-8078.020
x5	-1.628e+04	4055.109	-4.014	0.000	-2.43e+04	-8261.567
x6	-1.613e+04	3905.510	-4.129	0.000	-2.38e+04	-8408.085
x7	-1.63e+04	3957.895	-4.119	0.000	-2.41e+04	-8481.420
x8	177.5966	43.249	4.106	0.000	92.113	263.081
x9	3.057e+05	2.1e+05	1.455	0.148	-1.09e+05	7.21e+05
x10	-3.345e+06	2.07e+06	-1.619	0.108	-7.43e+06	7.38e+05
x11	145.6956	431.173	0.338	0.736	-706.549	997.941
x12	3.335e+04	1.15e+05	0.291	0.772	-1.93e+05	2.6e+05
x13	-24.6923	7.697	-3.208	0.002	-39.907	-9.478
x14	-3.8339	1107.726	-0.003	0.997	-2193.337	2185.670
x15	-3.667e+06	2.65e+06	-1.386	0.168	-8.9e+06	1.56e+06
x16	-366.8278	326.012	-1.125	0.262	-1011.215	277.559
x17	233.5882	267.009	0.875	0.383	-294.174	761.351
x18	-2.8730	6.792	-0.423	0.673	-16.298	10.552
x19	-1826.4961	426.477	-4.283	0.000	-2669.460	-983.532
x20	211.5068	1129.090	0.187	0.852	-2020.224	2443.237
x21	70.5015	76.100	0.926	0.356	-79.917	220.919
x22	-1085.6314	1232.639	-0.881	0.380	-3522.035	1350.772
x23	-7.6903	6.254	-1.230	0.221	-20.051	4.670
x24	75.5238	955.551	0.079	0.937	-1813.194	1964.242
x25	2.094e+06	5.35e+05	3.914	0.000	1.04e+06	3.15e+06
x26	-141.7146	124.476	-1.138	0.257	-387.750	104.321
x27	-1.482e+04	8263.429	-1.794	0.075	-3.12e+04	1511.764
x28	1554.5145	2240.689	0.694	0.489	-2874.375	5983.404
x29	21.9687	44.012	0.499	0.618	-65.025	108.963
x30	-11.8885	1123.194	-0.011	0.992	-2231.965	2208.188
x31	1.92e+07	3.71e+07	0.517	0.606	-5.42e+07	9.26e+07
x32	324.2592	858.464	0.378	0.706	-1372.560	2021.078
x33	-118.6065	45.810	-2.589	0.011	-209.154	-28.059
x34	-11.8493	1123.119	-0.011	0.992	-2231.777	2208.079
x35	416.9914	3559.369	0.117	0.907	-6618.369	7452.352
x36	-4.6607	2.965	-1.572	0.118	-10.521	1.199
x37	572.9766	403.536	1.420	0.158	-224.643	1370.596
x38	18.8322	5.224	3.605	0.000	8.506	29.158
x39	75.4063	955.776	0.079	0.937	-1813.756	1964.569
x40	-1.833e+07	3.55e+07	-0.516	0.606	-8.85e+07	5.18e+07

	coef	std err	t	P>|t|	[0.025	0.975]
x1	-1424.2883	41.546	-34.282	0.000	-1506.354	-1342.222
x2	-1.494e+04	3675.500	-4.066	0.000	-2.22e+04	-7683.499
x3	-1.565e+04	3779.256	-4.140	0.000	-2.31e+04	-8182.817
x4	-1.504e+04	3633.424	-4.141	0.000	-2.22e+04	-7867.729
x5	-1.574e+04	3901.616	-4.034	0.000	-2.34e+04	-8032.835
x6	-1.563e+04	3758.532	-4.158	0.000	-2.31e+04	-8204.565
x7	-1.58e+04	3810.109	-4.146	0.000	-2.33e+04	-8270.164
x8	171.7781	41.533	4.136	0.000	89.739	253.818
x9	3.028e+05	8.43e+04	3.592	0.000	1.36e+05	4.69e+05
x10	-3.3e+06	9.01e+05	-3.662	0.000	-5.08e+06	-1.52e+06
x11	249.7330	75.779	3.296	0.001	100.047	399.419
x12	2.798e+04	5812.875	4.814	0.000	1.65e+04	3.95e+04
x13	-26.5936	5.047	-5.269	0.000	-36.563	-16.625
x14	-3.155e+06	6.15e+05	-5.129	0.000	-4.37e+06	-1.94e+06
x15	-1565.5358	297.796	-5.257	0.000	-2153.769	-977.302
x16	-4.2560	0.943	-4.512	0.000	-6.119	-2.393
x17	1.813e+06	3.69e+05	4.919	0.000	1.08e+06	2.54e+06
x18	-1.027e+04	2158.900	-4.757	0.000	-1.45e+04	-6005.311
x19	41.5232	7.523	5.520	0.000	26.664	56.382
x20	1.519e+07	2.81e+06	5.410	0.000	9.65e+06	2.07e+07
x21	-125.5091	16.597	-7.562	0.000	-158.294	-92.725
x22	-5.9319	0.748	-7.930	0.000	-7.409	-4.454
x23	16.6047	4.412	3.764	0.000	7.890	25.319
x24	-1.437e+07	2.65e+06	-5.418	0.000	-1.96e+07	-9.13e+06

OLS Regression Results

Dep. Variable:	y	R-squared:	0.944
Model:	OLS	Adj. R-squared:	0.930
Method:	Least Squares	F-statistic:	69.09
Date:	Sun, 17 May 2020	Prob (F-statistic):	1.25e-73
Time:	16:20:46	Log-Likelihood:	-1466.8
No. Observations:	180	AIC:	3006.
Df Residuals:	144	BIC:	3121.
Df Model:	35		
Covariance Type:	nonrobust		

Omnibus:	3.959	Durbin-Watson:	0.961
Prob(Omnibus):	0.138	Jarque-Bera (JB):	4.910
Skew:	-0.038	Prob(JB):	0.0859
Kurtosis:	3.806	Cond. No.	1.11e+16

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 7.73e-21. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

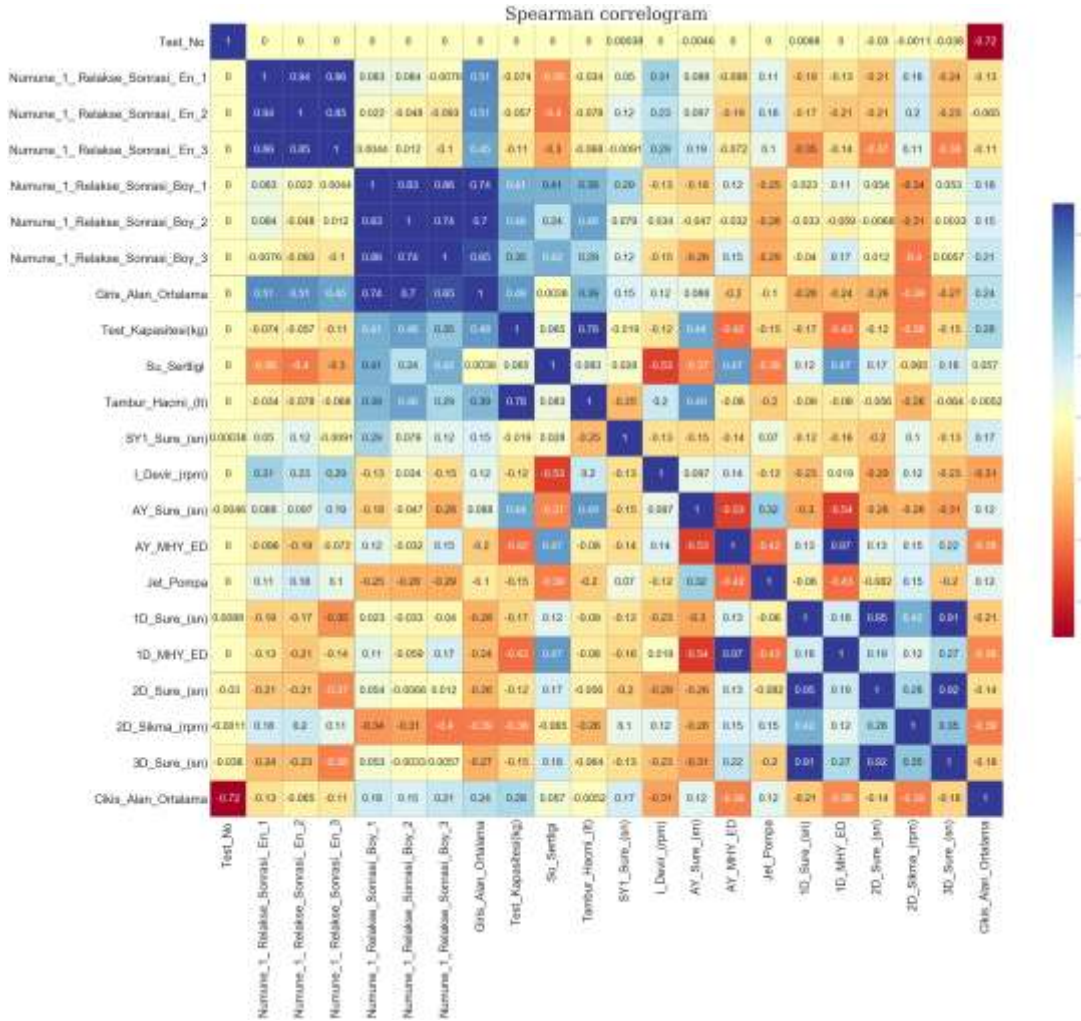
OLS Regression Results

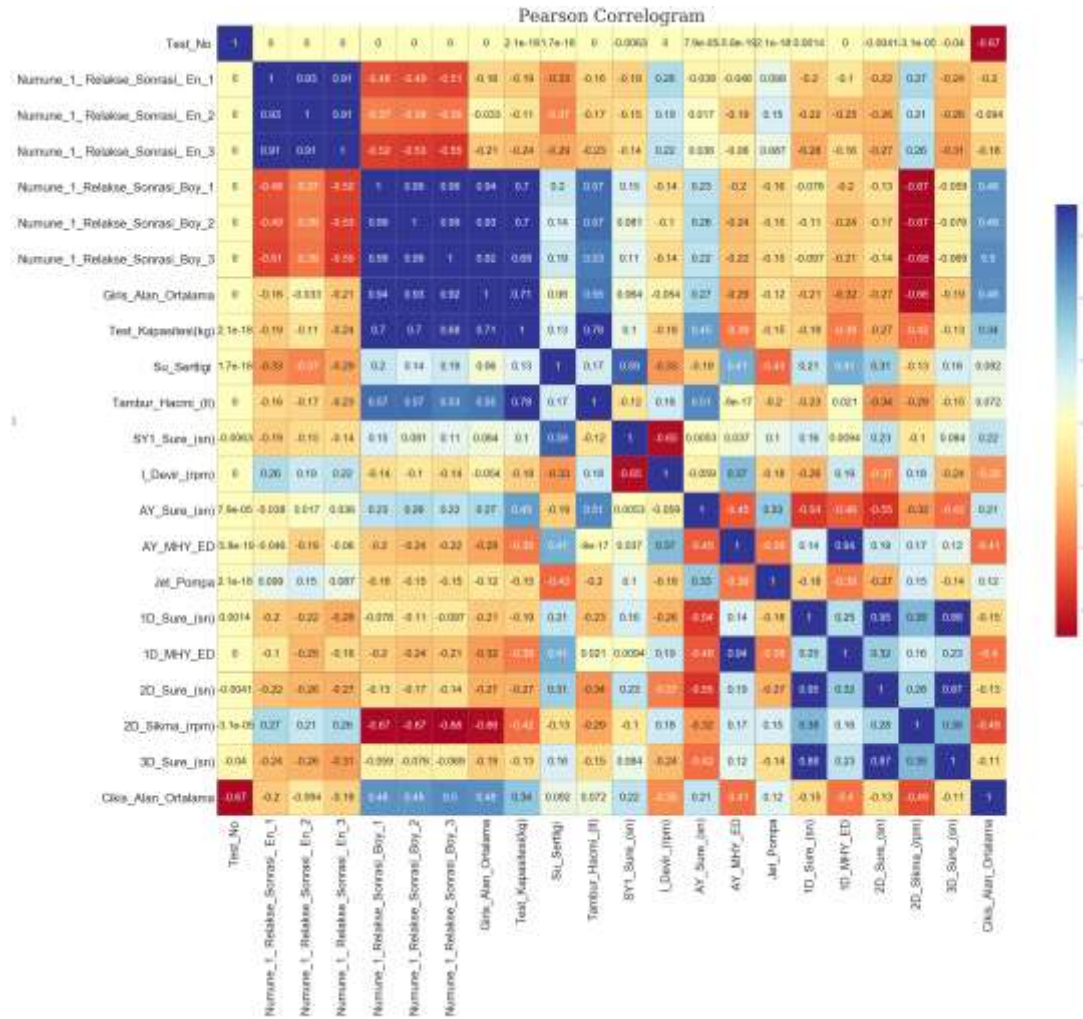
Dep. Variable:	y	R-squared:	0.940
Model:	OLS	Adj. R-squared:	0.931
Method:	Least Squares	F-statistic:	106.4
Date:	Sun, 17 May 2020	Prob (F-statistic):	3.18e-83
Time:	16:15:31	Log-Likelihood:	-1472.6
No. Observations:	180	AIC:	2993.
Df Residuals:	156	BIC:	3070.
Df Model:	23		
Covariance Type:	nonrobust		

Omnibus:	2.875	Durbin-Watson:	0.945
Prob(Omnibus):	0.237	Jarque-Bera (JB):	2.877
Skew:	-0.100	Prob(JB):	0.237
Kurtosis:	3.586	Cond. No.	4.14e+09

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.14e+09. This might indicate that there are
strong multicollinearity or other numerical problems.

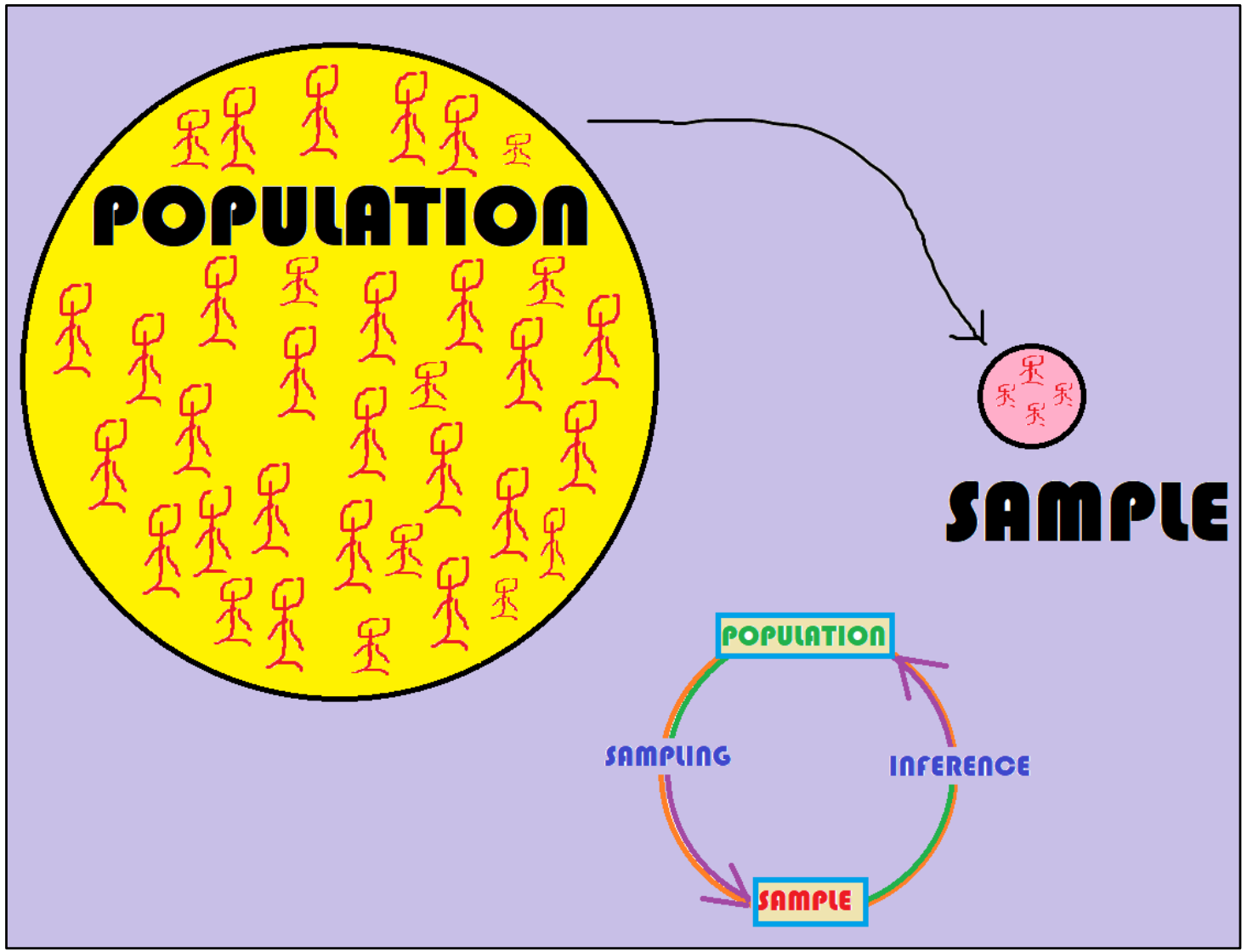
SL=0.05 (Significance Level)		Column	p-value	Değişken adı	
1	iterasyon	13	0.997	SY1_Devir_(rpm)	çıkarılır
2	iterasyon	29	0.981	1D_Devir_(rpm)	çıkarılır
3	iterasyon	33	0.993	2D_Devir_(rpm)	çıkarılır
4	iterasyon	38	0.995	3D_Devir_(rpm)	çıkarılır
5	iterasyon	23	0.981	AY_Devir_(rpm)	çıkarılır
6	iterasyon	34	0.806	2D_MHY_ED	çıkarılır
7	iterasyon	31	0.695	2D_Su_Mik_(lt)	çıkarılır
8	iterasyon	17	0.708	I_Sure_(sn)	çıkarılır
9	iterasyon	19	0.561	I_MHY_ED	çıkarılır
10	iterasyon	21	0.401	AY_Su_Mik_(lt)	çıkarılır
11	iterasyon	20	0.299	I_Tset_(C)	çıkarılır
12	iterasyon	15	0.427	SG_Sic_(C)	çıkarılır
13	iterasyon	25	0.296	AY_Tah_(C)	çıkarılır
14	iterasyon	27	0.209	1D_Su_Mik_(lt)	çıkarılır
15	iterasyon	16	0.217	I_Bas_Sic (C)	çıkarılır
16	iterasyon	36	0.115	3D_Su_Mik_(lt)	çıkarılır


Spearman correlogram
Test_No
Numune_1_ Relakse_Sonrasi_En_1
Numune_1_ Relakse_Sonrasi_En_2
Numune_1_ Relakse_Sonrasi_En_3
Numune_1_Relakse_Sonrasi_Boy_1
Numune_1_Relakse_Sonrasi_Boy_2
Numune_1_Relakse_Sonrasi_Boy_3
Giris_Alan_Ortalama
Test_Kapasitesi(kg)
Su_Sertligi
Tambur_Hacmi_(lt)
SY1_Sure_(sn)
I_Devir_(rpm)
AY_Sure_(sn)
AY_MHY_ED
Jet_Pompa
1D_Sure_(sn)
1D_MHY_ED
2D_Sure_(sn)
2D_Sikma_(rpm)
3D_Sure_(sn)
Cikis_Alan_Ortalama

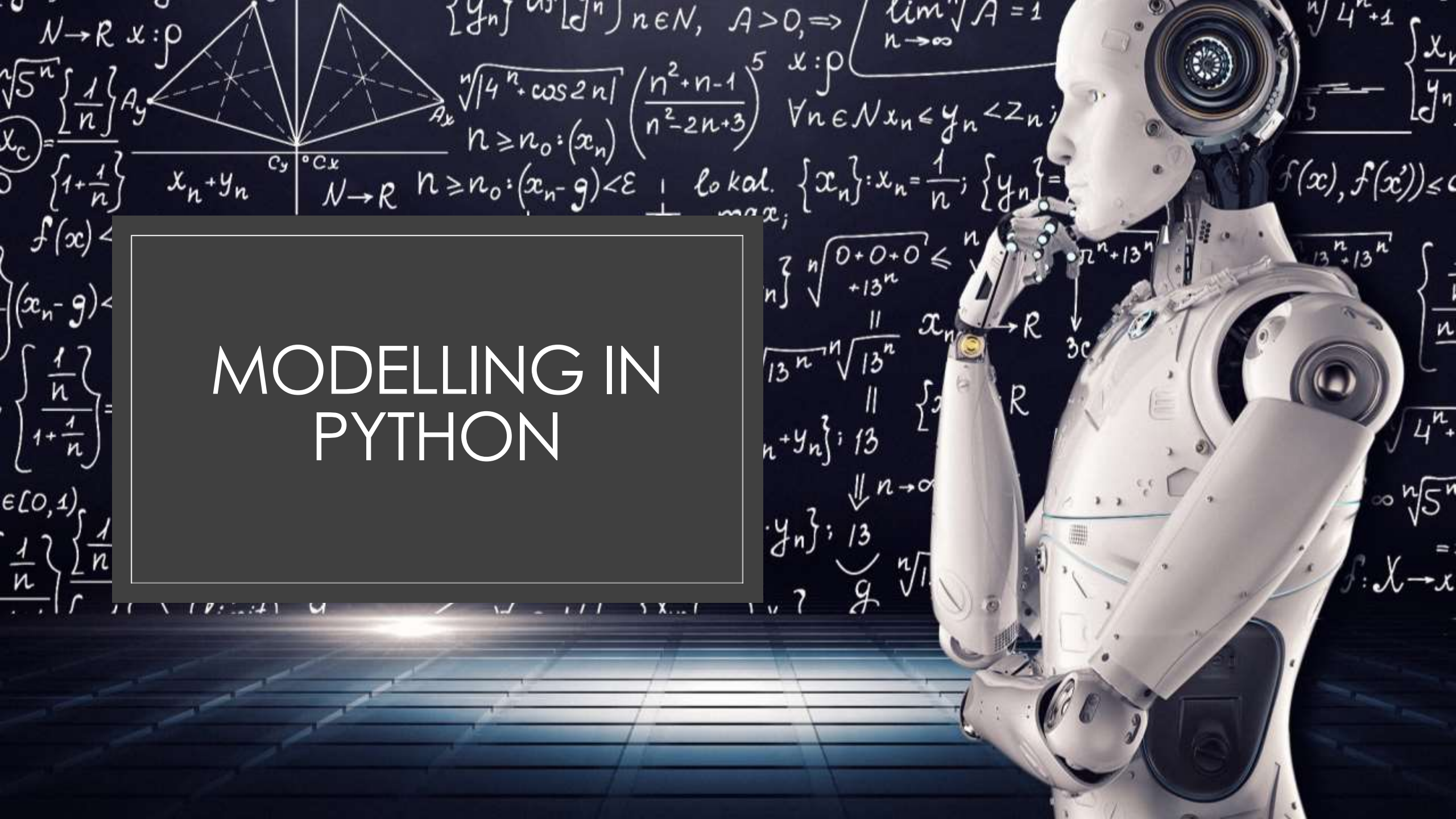

Pearson Correlogram
Test_No
Numune_1_ Relakse_Sonrasi_En_1
Numune_1_ Relakse_Sonrasi_En_2
Numune_1_ Relakse_Sonrasi_En_3
Numune_1_Relakse_Sonrasi_Boy_1
Numune_1_Relakse_Sonrasi_Boy_2
Numune_1_Relakse_Sonrasi_Boy_3
Giris_Alan_Ortalama
Test_Kapasitesi(kg)
Su_Sertligi
Tambur_Hacmi_(lt)
SY1_Sure_(sn)
I_Devir_(rpm)
AY_Sure_(sn)
AY_MHY_ED
Jet_Pompa
1D_Sure_(sn)
1D_MHY_ED
2D_Sure_(sn)
2D_Sikma_(rpm)
3D_Sure_(sn)
Cikis_Alan_Ortalama

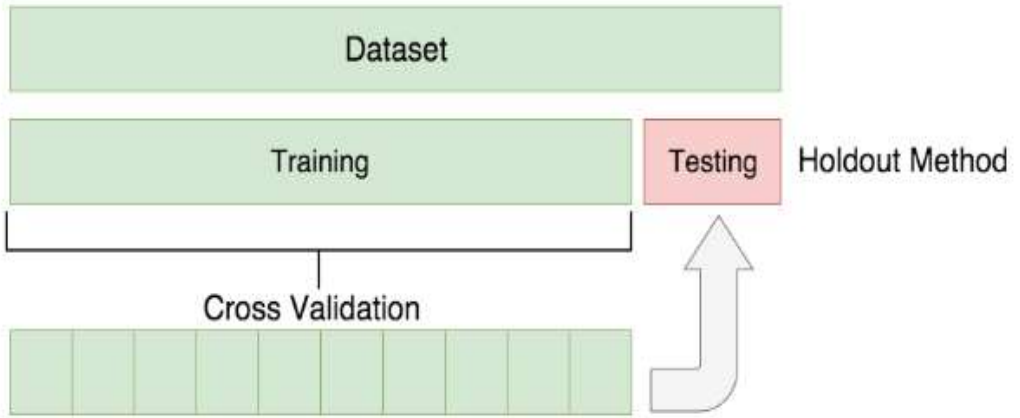
✓ What would be good to reduce p-values?

increase the power of analysis by taking larger sample size and doing better data collection (reducing error-for exp; zero values in our study!!!)



MODELLING IN PYTHON





CROSS VALIDATION

Determining train size for train-test split

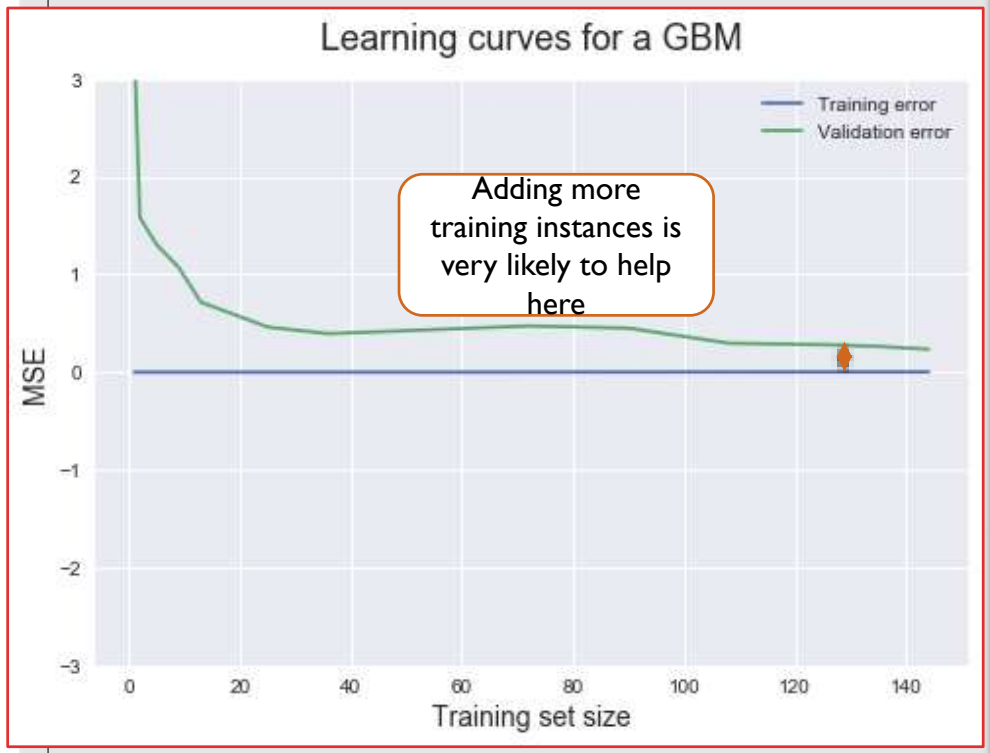
train_sizes = [1, 2, 5, 9,13, 25, 36, 72,90, 108, 126, 135, 144]
(180*0.50 - 180*0.80)

```
##### svr
from sklearn.svm import SVR
from sklearn.model_selection import learning_curve
features = df.columns
target = 'CAO'
train_sizes, train_scores, validation_scores = learning_curve(
estimator = SVR(),
X = df[features],
y = Cikis_AO[target], train_sizes = train_sizes, cv = 10,
scoring = 'neg_mean_squared_error', shuffle=True)

print('Training scores:\n\n', train_scores)
print('\n', '-' * 70) # separator to make the output easy to read
print('\nValidation scores:\n\n', validation_scores)

train_scores_mean = np.abs(train_scores).mean(axis = 1)
validation_scores_mean = np.abs(validation_scores).mean(axis =1 )

print('Mean training scores\n\n', pd.Series(train_scores_mean,
                                           index = train_sizes))
print('\n', '-' * 20) # separator
print('\nMean validation scores\n\n',pd.Series(validation_scores_mean,
                                             index = train_sizes))

import matplotlib.pyplot as plt

plt.style.use('seaborn')
plt.plot(train_sizes, train_scores_mean, label = 'Training error')
plt.plot(train_sizes, validation_scores_mean, label = 'Validation error')
plt.ylabel('MSE', fontsize = 14)
plt.xlabel('Training set size', fontsize = 14)
plt.title('Learning curves for a SVR', fontsize = 18 , y = 1.03)
plt.legend()
plt.ylim(-3,3)
```

```
############# GBM

from sklearn.ensemble import GradientBoostingRegressor
from sklearn.model_selection import learning_curve
features = df.columns
target = 'CAO'
train_sizes, train_scores, validation_scores = learning_curve(
estimator = GradientBoostingRegressor(),
X = df[features],
y = Cikis_AO[target], train_sizes = train_sizes, cv = 10,
scoring = 'neg_mean_squared_error', shuffle=True)

print('Training scores:\n\n', train_scores)
print('\n', '-' * 70) # separator to make the output easy to read
print('\nValidation scores:\n\n', validation_scores)

train_scores_mean = np.abs(train_scores).mean(axis = 1)
validation_scores_mean = np.abs(validation_scores).mean(axis =1 )

print('Mean training scores\n\n', pd.Series(train_scores_mean,
                                           index = train_sizes))
print('\n', '-' * 20) # separator
print('\nMean validation scores\n\n',pd.Series(validation_scores_mean,
                                             index = train_sizes))

import matplotlib.pyplot as plt

plt.style.use('seaborn')
plt.plot(train_sizes, train_scores_mean, label = 'Training error')
plt.plot(train_sizes, validation_scores_mean, label = 'Validation error')
plt.ylabel('MSE', fontsize = 14)
plt.xlabel('Training set size', fontsize = 14)
plt.title('Learning curves for a GBM', fontsize = 18 , y = 1.03)
plt.legend()
plt.ylim(-3,3)
```

Determining train size for train-test split

- Adding more training instances
- Adding more features
- Feature selection
- Hyperparameter optimization

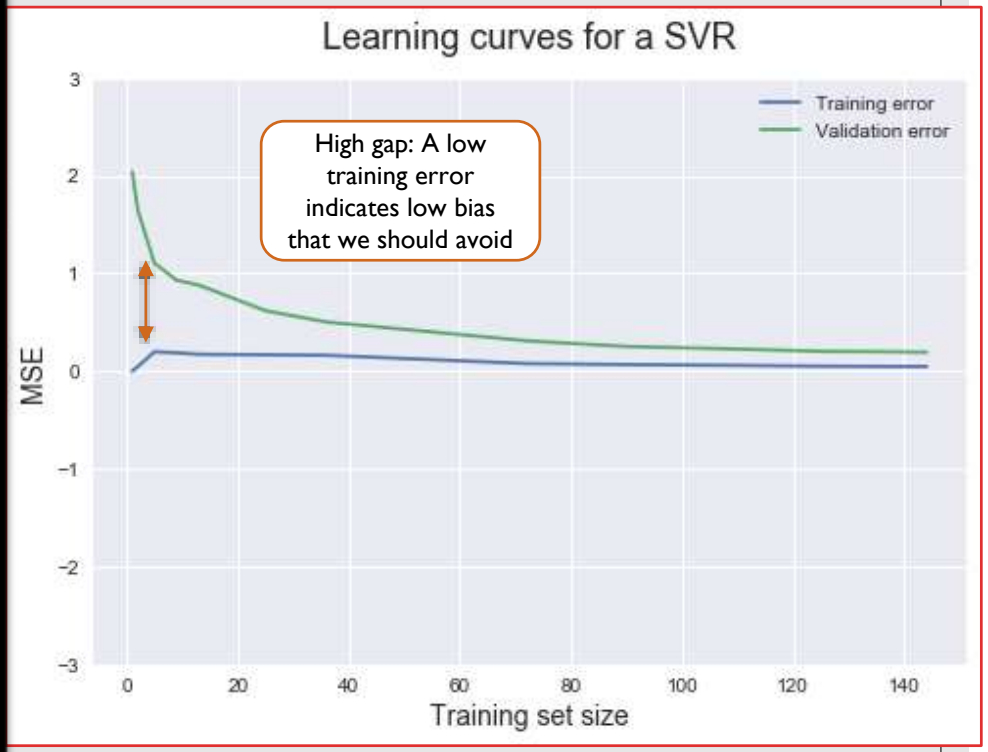


	Mean training scores	Mean validation scores
1	0.000000e+00	3.344291
2	9.147134e-08	1.583766
5	9.340545e-08	1.307148
9	9.644352e-08	1.071880
13	9.622787e-08	0.717403
25	2.324368e-06	0.462318
36	6.548008e-05	0.394996
72	8.765447e-04	0.473935
90	1.677802e-03	0.451119
108	2.348792e-03	0.297287
126	2.804317e-03	0.280765
135	3.075798e-03	0.264330
144	3.163698e-03	0.235232

dtype: float64

	Mean training scores	Mean validation scores
1	0.000000	2.039337
2	0.048002	1.646903
5	0.198824	1.105167
9	0.188524	0.931225
13	0.172998	0.881341
25	0.168690	0.618940
36	0.162902	0.503384
72	0.078362	0.311459
90	0.068138	0.253115
108	0.060219	0.229787
126	0.051240	0.202319
135	0.049136	0.200815
144	0.048359	0.193558

dtype: float64



Let's split 70:30

126

Data multiplexing, multiplexing variables extracted from df before fitting

```
df.drop(['Durulama_Sayisi',
         'SY1_Devir_(rpm)',
         '1D_Devir_(rpm)',
         '2D_Devir_(rpm)',
         '3D_Devir_(rpm)',
         'Deterjan_Miktarı(gr)',
         '3D_MHY_ED',
         'SY1_ED'
        ], axis=1, inplace=True)
```

GBM – CROSS VALIDATION MEAN SCORE: 93,16% / STD: 0.04

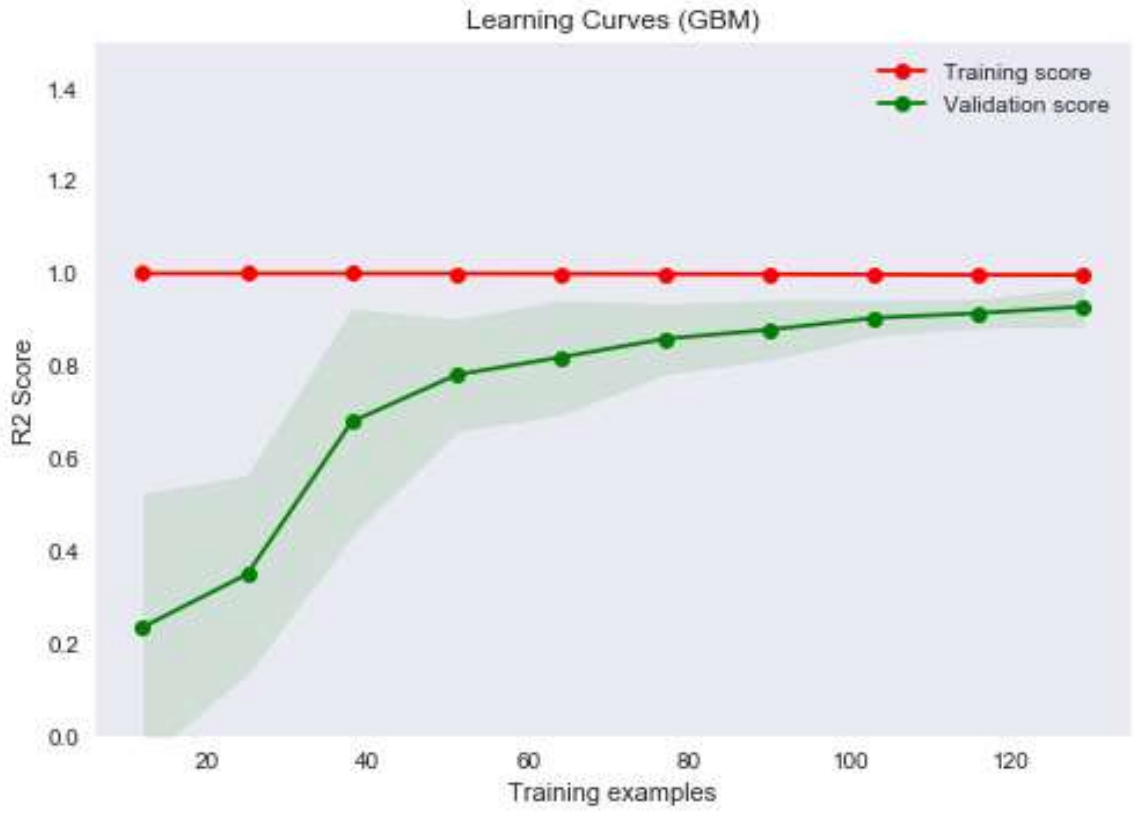
SVR – CROSS VALIDATION MEAN SCORE: 90,36% / STD: 0.04

RF – CROSS VALIDATION MEAN SCORE: 87,59% / STD: 0.05

GLM – CROSS VALIDATION MEAN SCORE: 87,79% / STD: 0.06

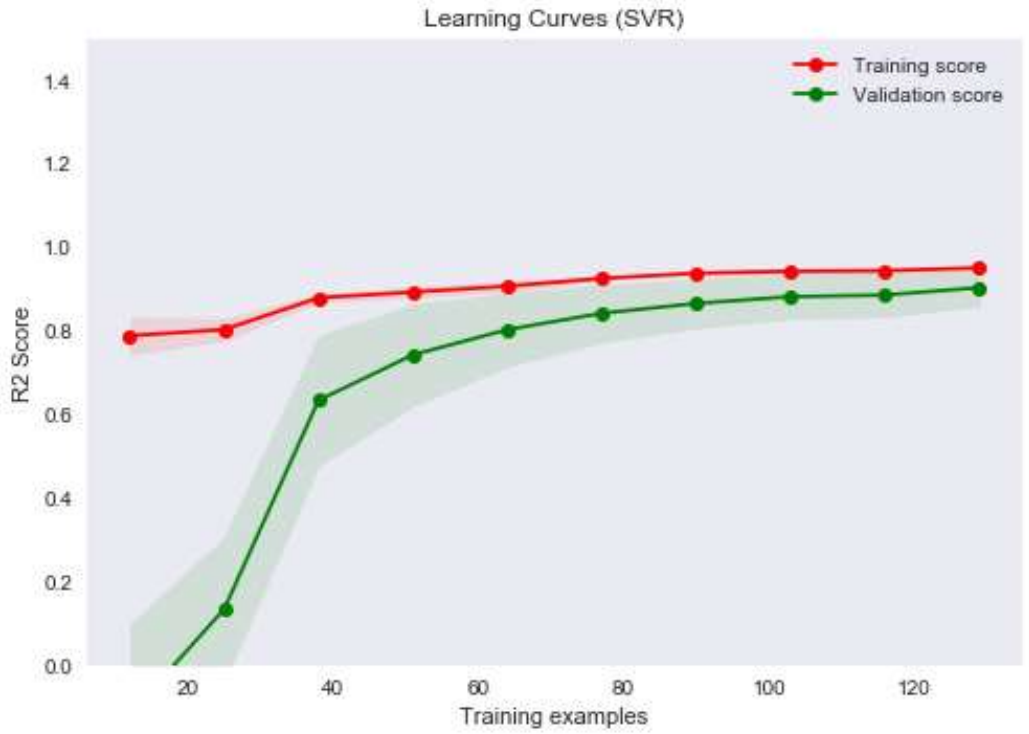
AdaBoost – CROSS VALIDATION MEAN SCORE: 82,21% / STD: 0.07

GBM – CROSS VALIDATION MEAN SCORE: 93,16% / STD: 0.04



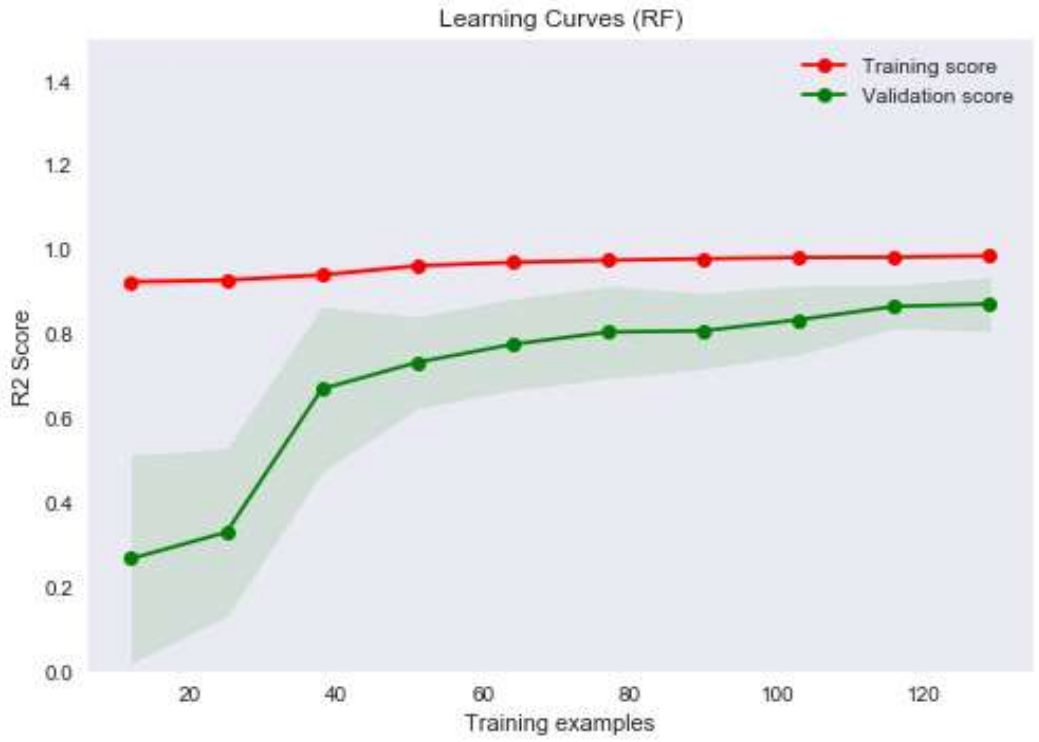

```
#model5 - Gradient Descent

from sklearn.ensemble import GradientBoostingRegressor
from sklearn.model_selection import cross_val_score
gb_reg = GradientBoostingRegressor()
gb_scores = cross_val_score(gb_reg, x_train, y_train, cv=10)
print(gb_scores)
# [0.88337849 0.95637685 0.94947434 0.96500836 0.96981316 0.84037879
#  0.97163019 0.87310394 0.93085526 0.976291  ]

print(gb_scores.mean())
# 0.9316310368054921
print(gb_scores.std())
# 0.046020003468920455

from sklearn.model_selection import learning_curve
title = "Learning Curves (GBM)"
cv = 10
plot_learning_curve(gb_reg, title,
                    x_train, y_train, ylim=(0, 1.5),
                    cv=cv, n_jobs=-1);
```

SVR – CROSS VALIDATION MEAN SCORE: 90,36% / STD: 0.04



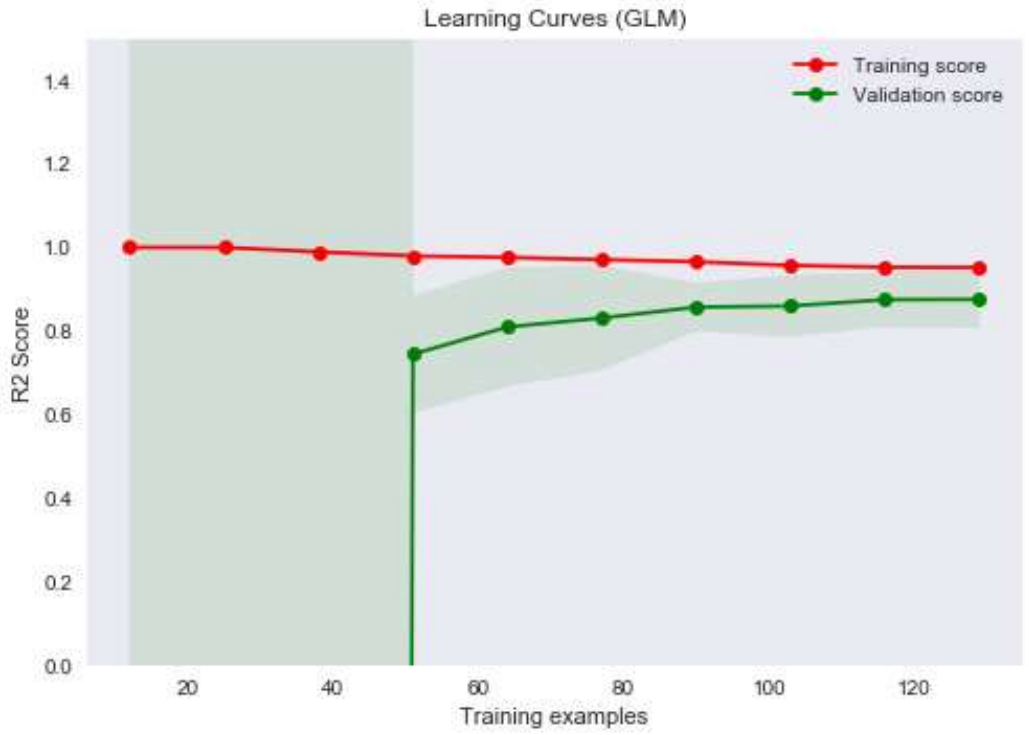

```
#model2 - SVR
from sklearn.svm import SVR

from sklearn.model_selection import cross_val_score
svr_reg = SVR()
svr_scores = cross_val_score(svr_reg, x_train, y_train, cv=10)
print(svr_scores)
# [0.82345469 0.94368955 0.84945493 0.89961801 0.88704056 0.89142335
#  0.89996566 0.8452074  0.95932952 0.97146793]
print(svr_scores.mean())
# 0.8970651587841191
print(svr_scores.std())
# 0.047019328549225234


# Assessing model performance
# Plot learning curve
from sklearn.model_selection import learning_curve
title = "Learning Curves (SVR)"
cv = 10
plot_learning_curve(svr_reg, title,
                    x_train, y_train, ylim=(0, 1.5),
                    cv=cv, n_jobs=-1);
```

RF – CROSS VALIDATION MEAN SCORE: 87,59% / STD: 0.05



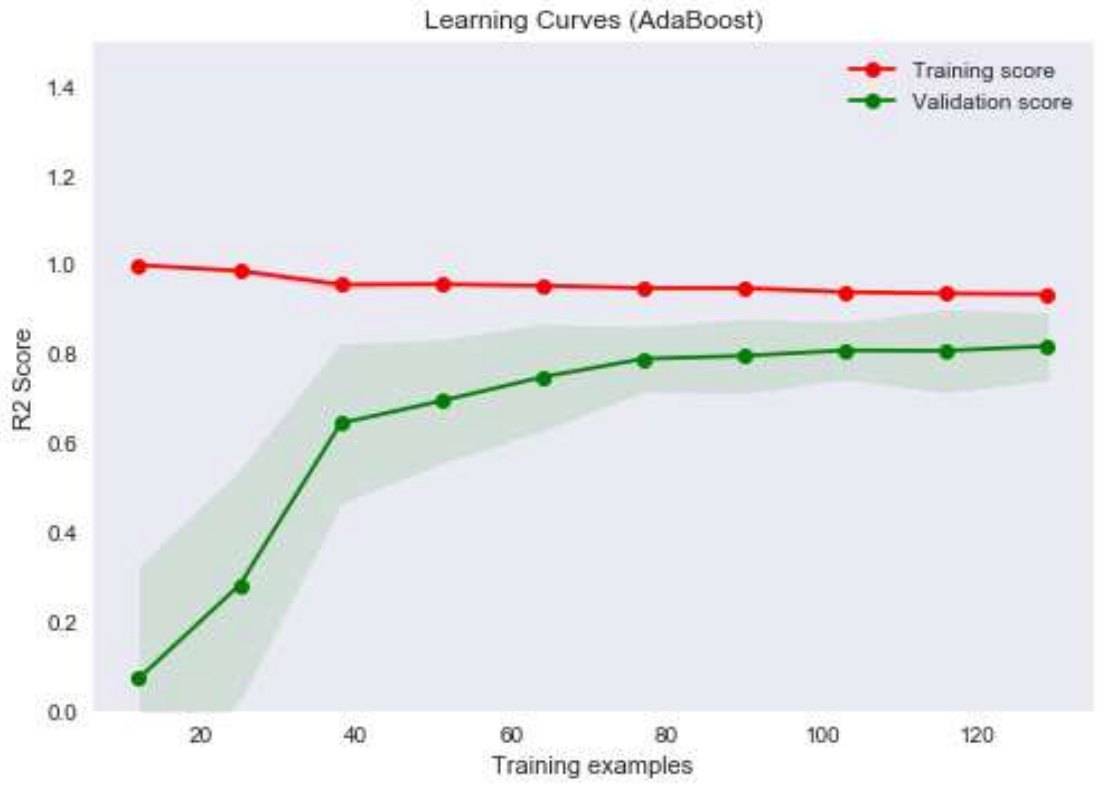

```
# model3- RandomForest
from sklearn.ensemble import RandomForestRegressor

from sklearn.model_selection import cross_val_score
rf_reg = RandomForestRegressor()
rf_scores = cross_val_score(rf_reg, x_train, y_train, cv=10)

print(rf_scores)
# [0.81600877 0.92538804 0.87121921 0.92250344 0.88830655 0.75914512
#  0.90155346 0.82045205 0.81484911 0.94420484]

print(rf_scores.mean())
# 0.8663630577214168
print(rf_scores.std())
# 0.057646548431513824

from sklearn.model_selection import learning_curve
title = "Learning Curves (RF)"
cv = 10
plot_learning_curve(rf_reg, title,
                    x_train, y_train, ylim=(0, 1.5),
                    cv=cv, n_jobs=-1);
```

GLM – CROSS VALIDATION MEAN SCORE: 87,79% / STD: 0.06




```
#model1- linear regression

from sklearn.linear_model import LinearRegression

from sklearn.model_selection import cross_val_score
lr = LinearRegression()
lr_scores = cross_val_score(lr, x_train, y_train, cv=10)
print(lr_scores)
# [0.80768775 0.94315428 0.929207   0.95287587 0.91778322 0.88557883
#  0.62466177 0.69620303 0.8999114  0.9549427 ]

print(lr_scores.mean())
# 0.8612005846129721
print(lr_scores.std())
# 0.10948336320814078

from sklearn.model_selection import learning_curve
title = "Learning Curves (GLM)"
cv = 10
plot_learning_curve(lr, title,
                    x_train, y_train, ylim=(0.7, 1.0),
                    cv=cv, n_jobs=-1);
```

AdaBoost – CROSS VALIDATION MEAN SCORE: 82,21% / STD: 0.07




```
#model4 - AdaBoost

from sklearn.ensemble import AdaBoostRegressor

from sklearn.model_selection import cross_val_score
ab_reg = AdaBoostRegressor()
ab_scores = cross_val_score(ab_reg, x_train, y_train, cv=10)
print(ab_scores)
# [0.77385224 0.86439727 0.8469751  0.92960135 0.81351651 0.75816734
#  0.82270168 0.64031523 0.6504978  0.90786072]

print(ab_scores.mean())
# 0.8007885251110226
print(ab_scores.std())
# 0.09260657075819591

from sklearn.model_selection import learning_curve
title = "Learning Curves (AdaBoost)"
cv = 10
plot_learning_curve(ab_reg, title,
                    x_train, y_train, ylim=(0, 1.5),
                    cv=cv, n_jobs=-1);
```

INTERPRETATION OF LEARNING CURVES

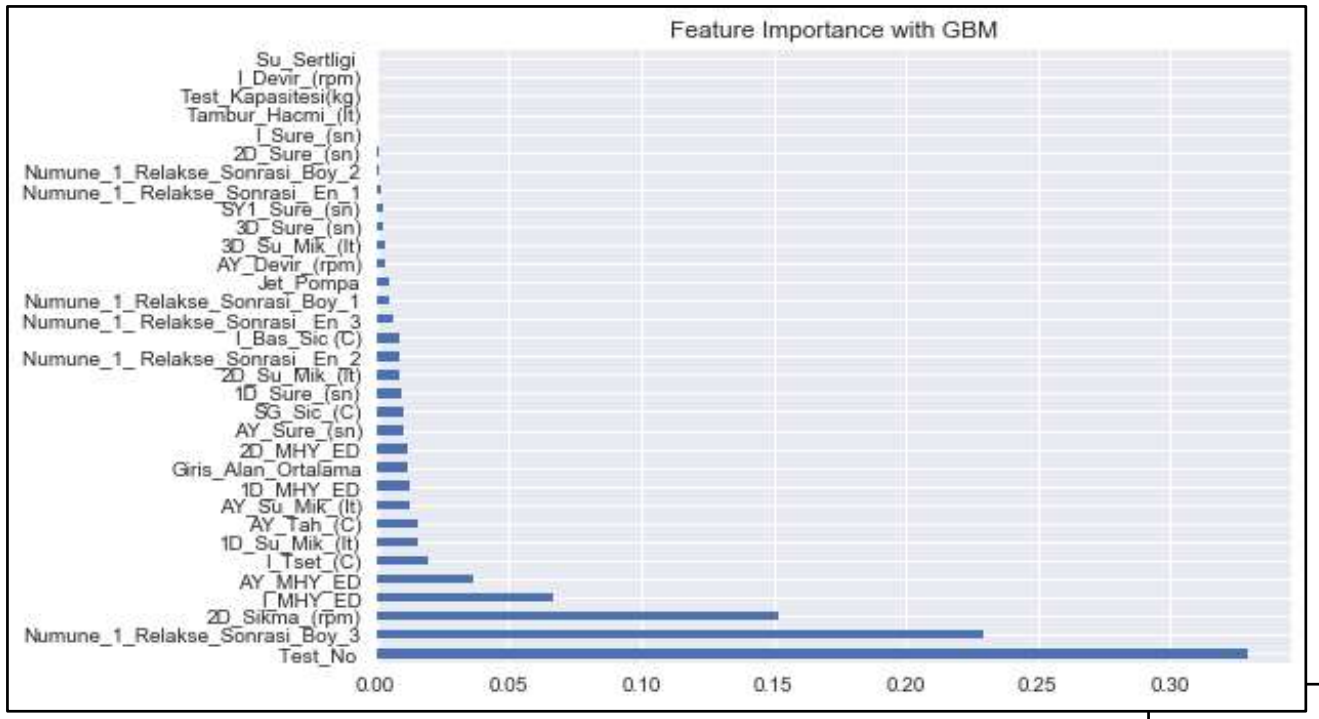
- When the model **overfits**, it means that it performs well on the training set, but not on the validation set. Accordingly, the model is not able to generalize to unseen data. If the model is overfitting, the learning curve will present **a gap** between the training and validation scores. Two common solutions for overfitting are reducing the complexity of the model and/or collect more data.
- On the other hand, **underfitting** means that the model is not able to perform well in either training or validations sets. In those cases, the learning curves will converge to a low score value. When the model underfits, gathering more data is not helpful because the model is already not being able to learn the training data. Therefore, the best approaches for these cases are to improve the model (e.g., tuning the hyperparameters) or to improve the quality of the data (e.g., collecting a different set of features).

Hyperparameter optimization – GBM/XGBoost

```python
#model5 - Gradient Descent

from sklearn.ensemble import GradientBoostingRegressor
from sklearn.model_selection import GridSearchCV
model_gbm = GradientBoostingRegressor()

grid_param_gbm = [{'max_depth': [4,5,6,7,8,10,12,15],
                   'learning_rate': [0.05,0.02,0.01,0.1,0.15,0.2,0.3,0.4,0.5],
                   'n_estimators': [10,30,50,100,200,300,500],
               }]


gs_gbm = GridSearchCV(model_gbm,
                  param_grid=grid_param_gbm,
                  scoring = 'neg_root_mean_squared_error',
                  cv=10,
                  n_jobs = -1)


grid_search_gbm = gs_gbm.fit(x_train, y_train)
best_model = grid_search_gbm.best_score_
best_parameters = grid_search_gbm.best_params_
print(best_model)
# 0.19228813093340386

print(best_parameters)
# {'learning_rate': 0.15, 'max_depth': 4, 'n_estimators': 500}
# {'learning_rate': 0.1, 'max_depth': 4, 'n_estimators': 200}

model_gbm = GradientBoostingRegressor(learning_rate=0.1,
                                      max_depth = 4,
                                      n_estimators=200)
model_gbm.fit(x_train, y_train)
pred_gbm = model_gbm.predict(x_test)
```

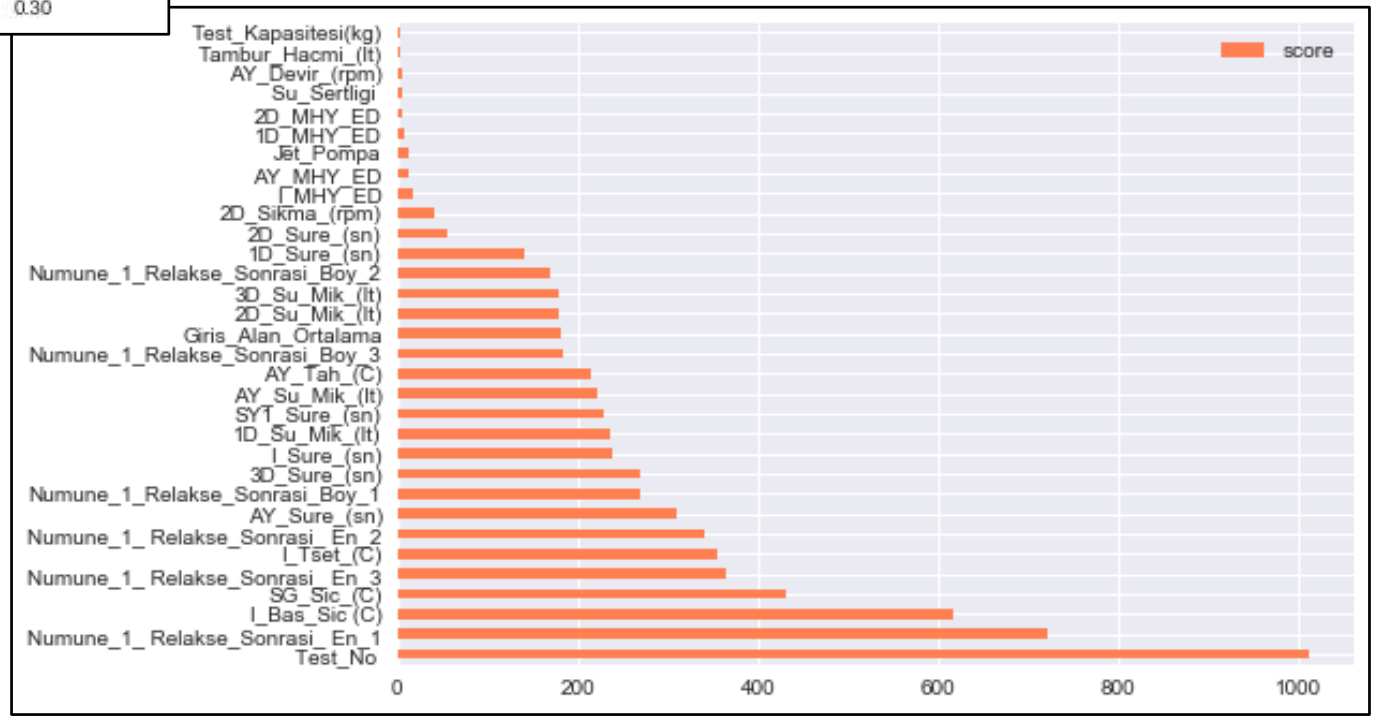
```python
from sklearn.model_selection import GridSearchCV
from xgboost import XGBModel

estimator = XGBModel(objective= 'reg:linear',
                     nthread=4,
                     seed=12345
                     )

grid_param_xgb = [{'max_depth': range (2, 10, 1),
                   'n_estimators': range(60, 300, 40),
                   'learning_rate': [0.05,0.01,0.1,0.3],
                   'gamma': [0,0.3,0.6,1],
                   'booster': ['gbtree', 'dart']
                  }]

grid_search = GridSearchCV(estimator=estimator,
                           param_grid=grid_param_xgb,
                           scoring = 'neg_root_mean_squared_error',
                           n_jobs = 20,
                           cv = 10,
                           verbose=True
                           )

grid_search.fit(sparse_train,label_train)

grid_search.best_estimator_
grid_search.best_params_
```

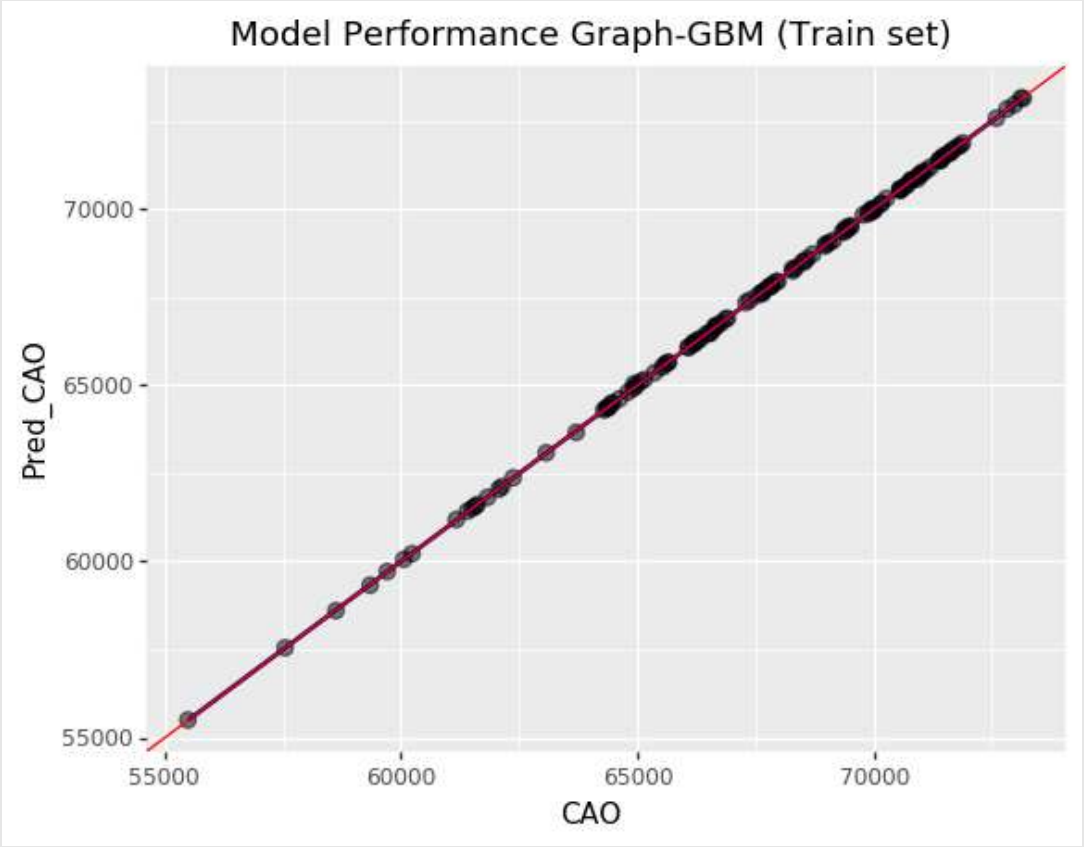
```python
xgb_params = {'booster': 'gbtree',
              'gamma': 0.6,
              'learning_rate': 0.1,
              'max_depth': 4,
              'n_estimators': 260,
              'objective':'reg:linear'
              }

xgb_params['eval_metric'] = "rmse"
evallist = [(dense_test, 'eval'), (dense_train, 'train')]
num_round = 5000
bst = xgb.train(xgb_params, dense_train, num_round, evallist,early_stopping_rounds=10)

bst.save_model('1.model')
```

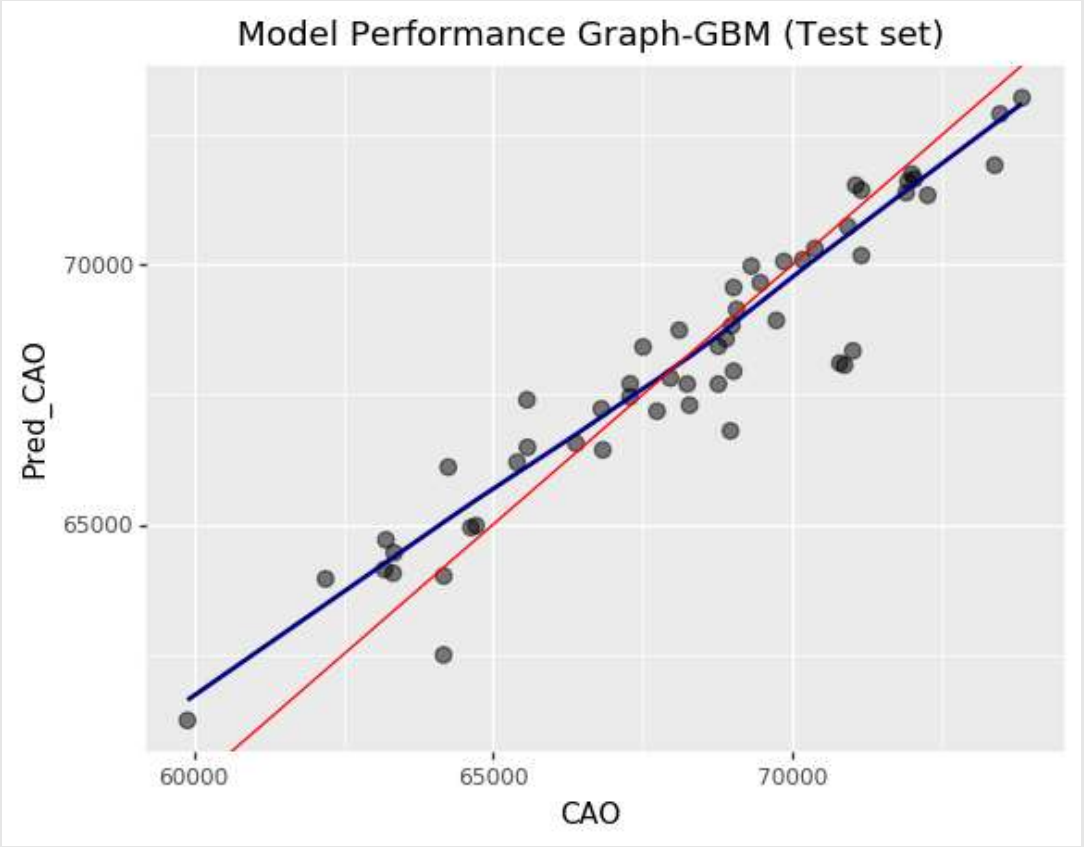



```
##### feature importance #####
model_gbm.fit(x_train, y_train)
print(model_gbm.feature_importances_)
#plot graph of feature importances for better visualization
feat_importances_gb = pd.Series(model_gbm.feature_importances_, index=x_test.columns)
feat_importances_gb.nlargest(33).plot(kind='barh')
plt.title('Feature Importance with GBM')
plt.show()
```

```
#prediction
ypred = bst.predict(dense_test, ntree_limit=bst.best_ntree_limit)

ypred.shape, label_test.shape

ypred_train = bst.predict(dense_train, ntree_limit=bst.best_ntree_limit)


# feature_imp_xgb=pd.DataFrame(bst.get_fscore().items(), columns=['feature','importance']).sort_values('impor
# #plot graph of feature importances for better visualization

feature_important = bst.get_score(importance_type='weight')
keys = list(feature_important.keys())
values = list(feature_important.values())
data = pd.DataFrame(data=values, index=keys, columns=["score"]).sort_values(by = "score", ascending=False)
data.plot(kind='barh', color='coral')
data.plot.title('Feature Importance with XGBoost',fontsize=50, fontfamily='serif')
data.plot.xticks(rotation=90, size=25)
data.plot.show()
```



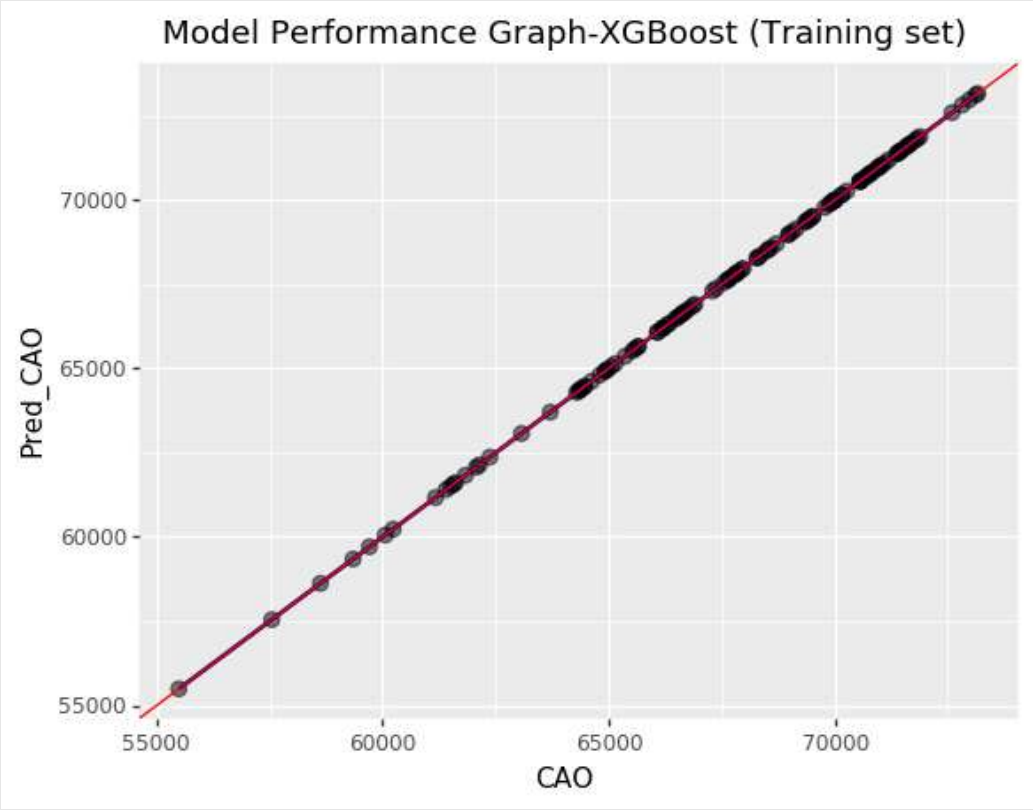


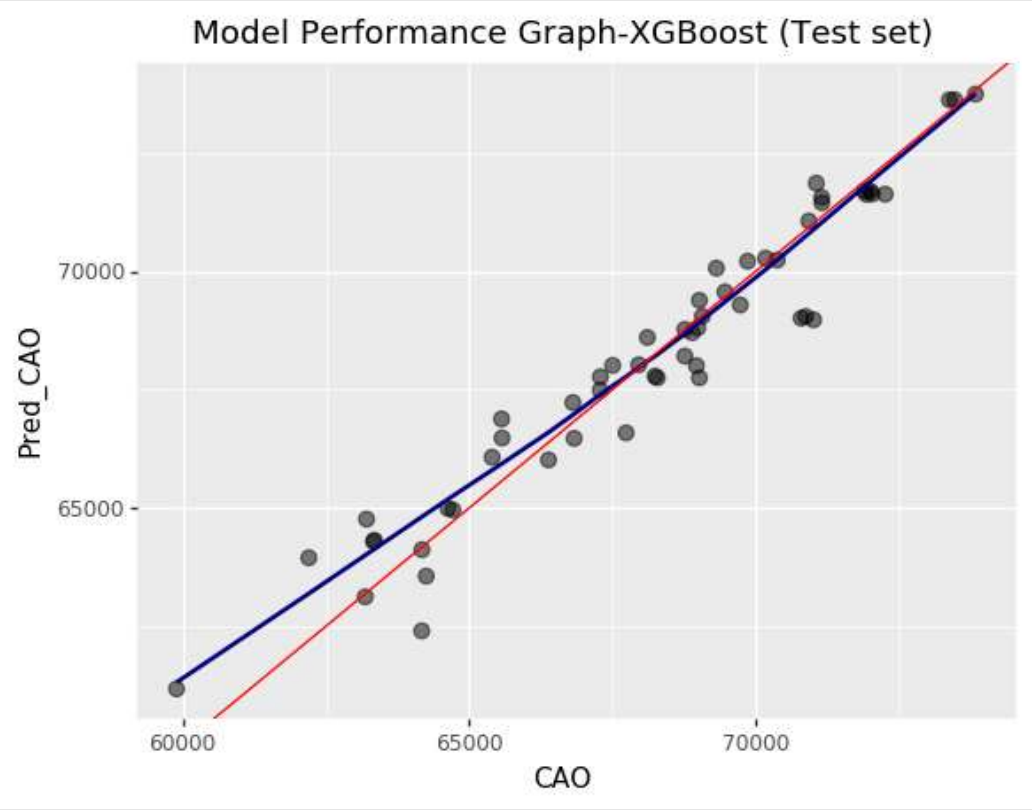


R-squared: 1.00
Mean Squared Error (MSE): 690.61
Root Mean Squared Error (RMSE): 26.28
Mean Absolute Error (MAE): 19.87

R-squared: 0.89
Mean Squared Error (MSE): 1137282.24
Root Mean Squared Error (RMSE): 1066.43
Mean Absolute Error (MAE): 776.27

XGBOOST



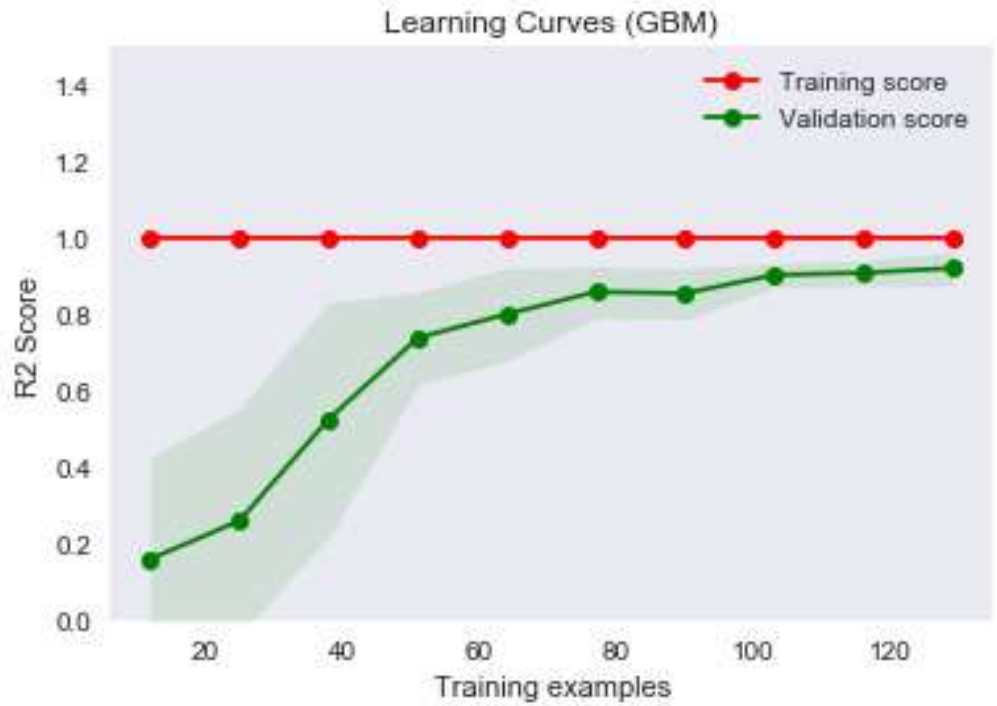


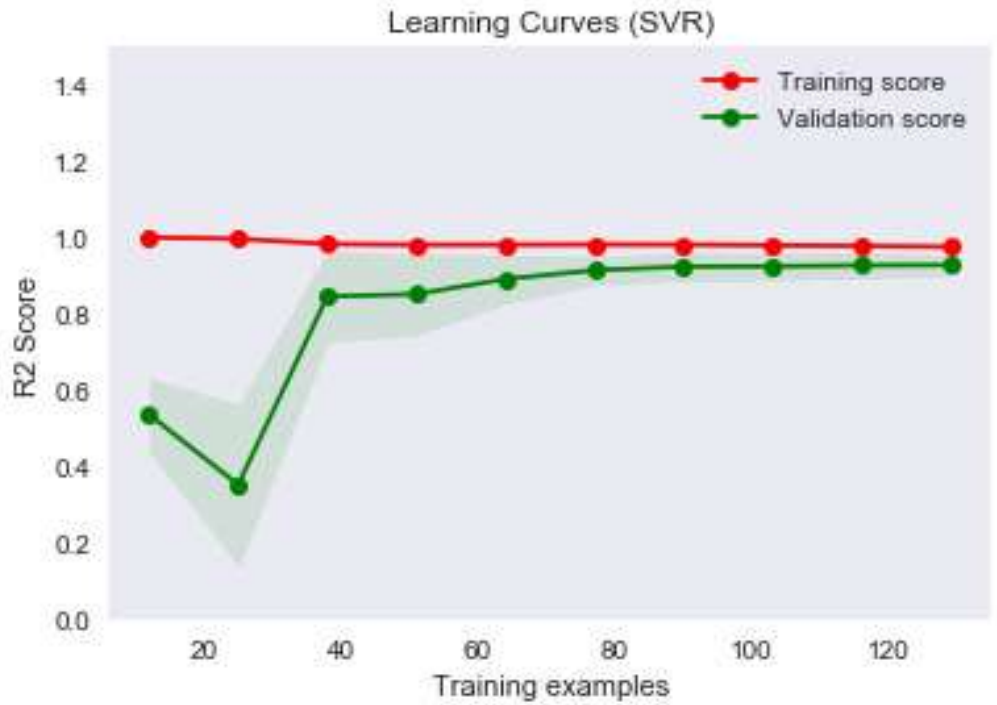
R-squared: 1.00
Mean Squared Error (MSE): 0.93
Root Mean Squared Error (RMSE): 0.97
Mean Absolute Error (MAE): 0.71

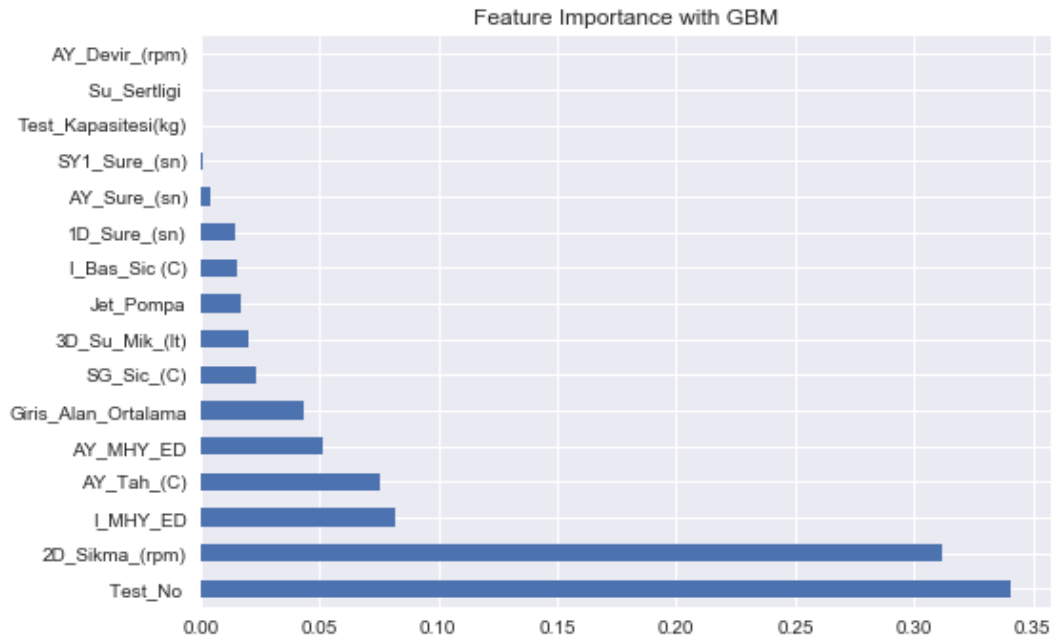
R-squared: 0.93
Mean Squared Error (MSE): 653156.78
Root Mean Squared Error (RMSE): 808.18
Mean Absolute Error (MAE): 605.77

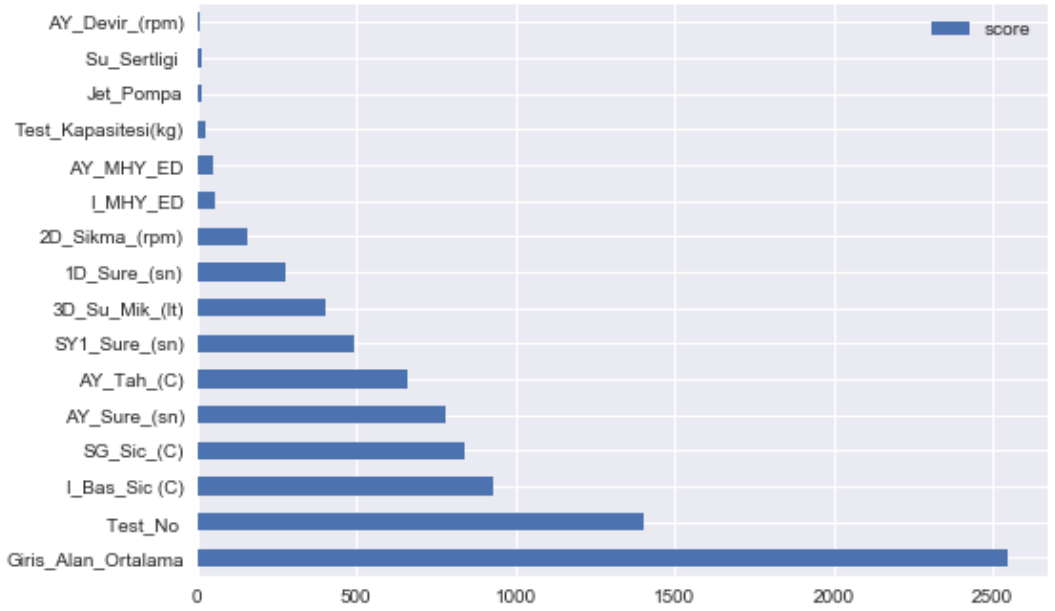
- Feature selection (extraction)

```
#adım 2: p value 0.05 den büyük değişkenler çıkarılır...

#OLS vriabls to remove

df.drop(['Numune_1_ Relakse_Sonrasi_ En_1',
         'Numune_1_ Relakse_Sonrasi_ En_2',
         'Numune_1_ Relakse_Sonrasi_ En_3',
         'Numune_1_Relakse_Sonrasi_Boy_1',
         'Numune_1_Relakse_Sonrasi_Boy_2',
         'Numune_1_Relakse_Sonrasi_Boy_3',
         '2D_MHY_ED',
         '2D_Su_Mik_(lt)',
         'I_Sure_(sn)',
          # 'I_MHY_ED', #
         'AY_Su_Mik_(lt)',
         'I_Tset_(C)',
          # 'SG_Sic_(C)', #
          # 'AY_Tah_(C)', #
         '1D_Su_Mik_(lt)', #
          # 'I_Bas_Sic (C)', #
          # '3D_Su_Mik_(lt)' #

          'Tambur_Hacmi_(lt)',
          '1D_MHY_ED',
          '2D_Sure_(sn)',
          '3D_Sure_(sn)',
          'I_Devir_(rpm)'], axis=1, inplace=True)
```

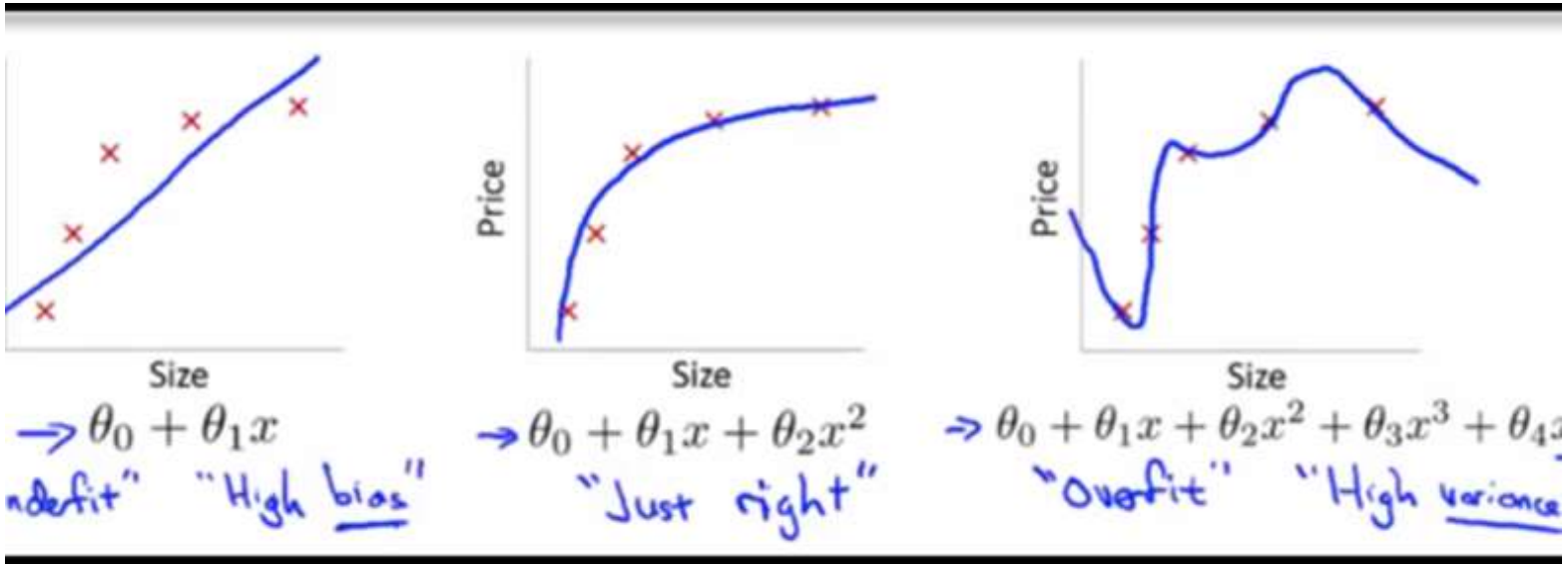

Learning Curves (GBM)
Training score
Validation score
R2 Score
Training examples
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
20
40
60
80
100
120


Learning Curves (SVR)
Training score
Validation score
R2 Score
Training examples
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
20
40
60
80
100
120

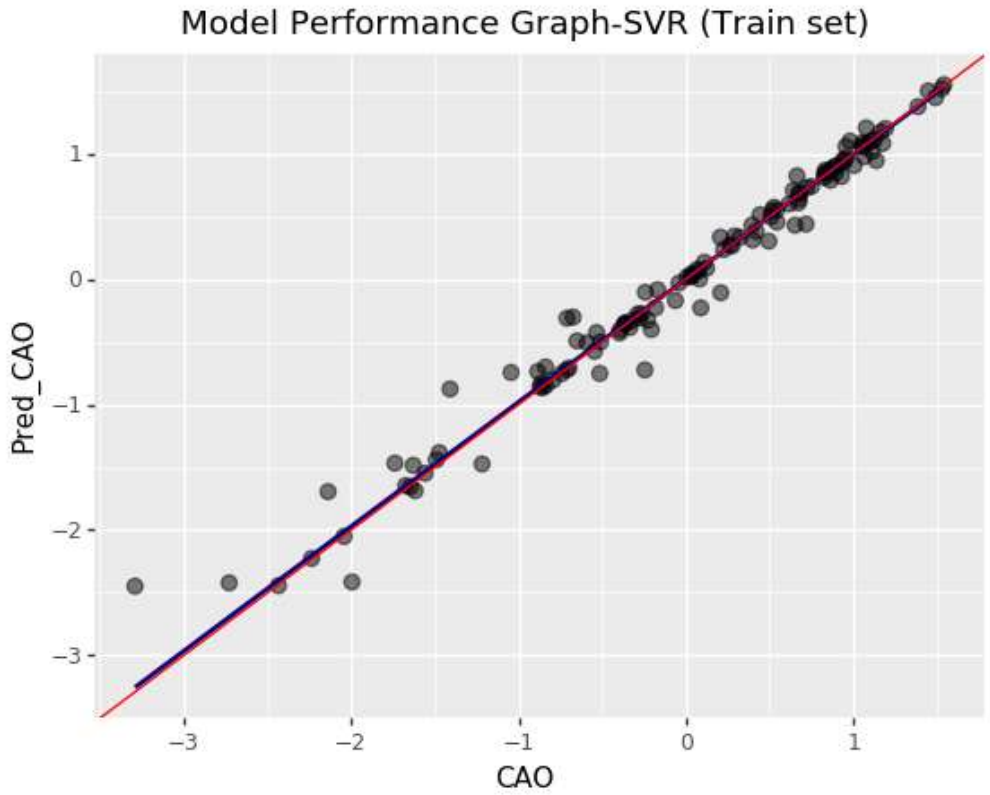


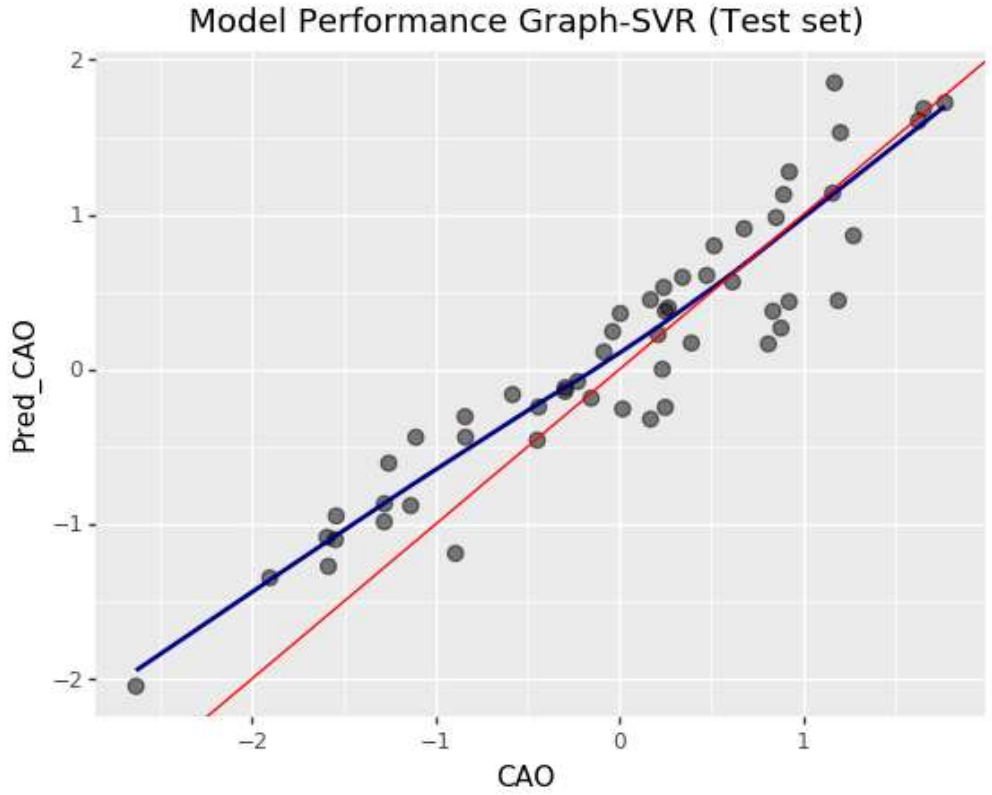


FEATURE IMPORTANCE FOR GBM, XGBOOST



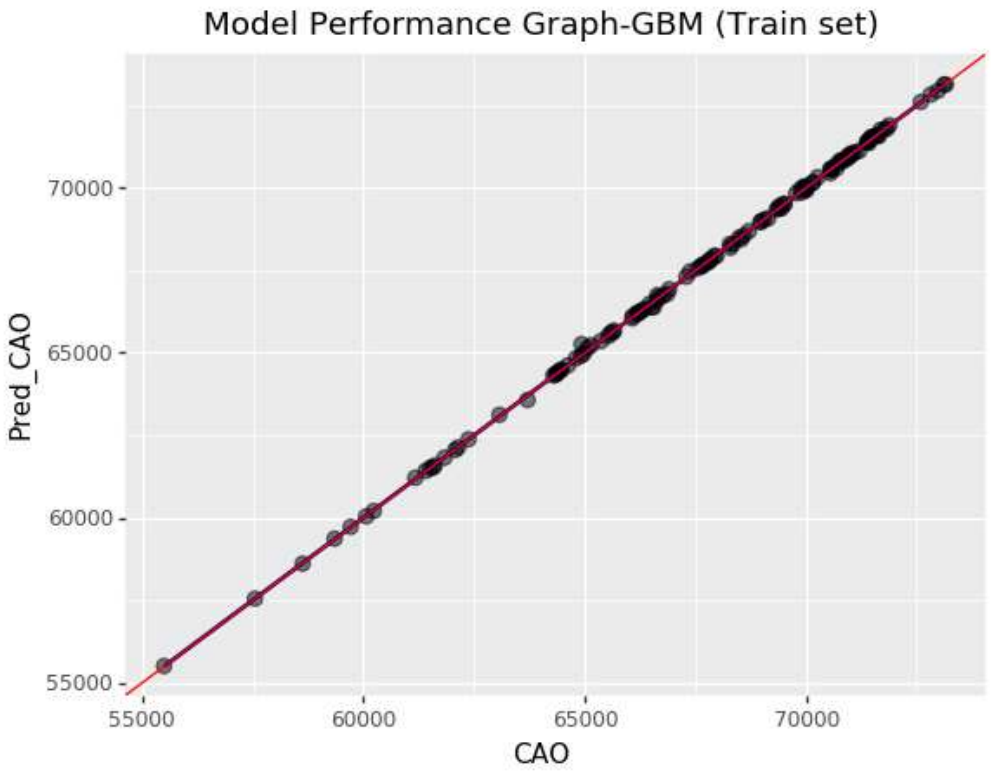
MODEL PERFORMANCE GRAPHS



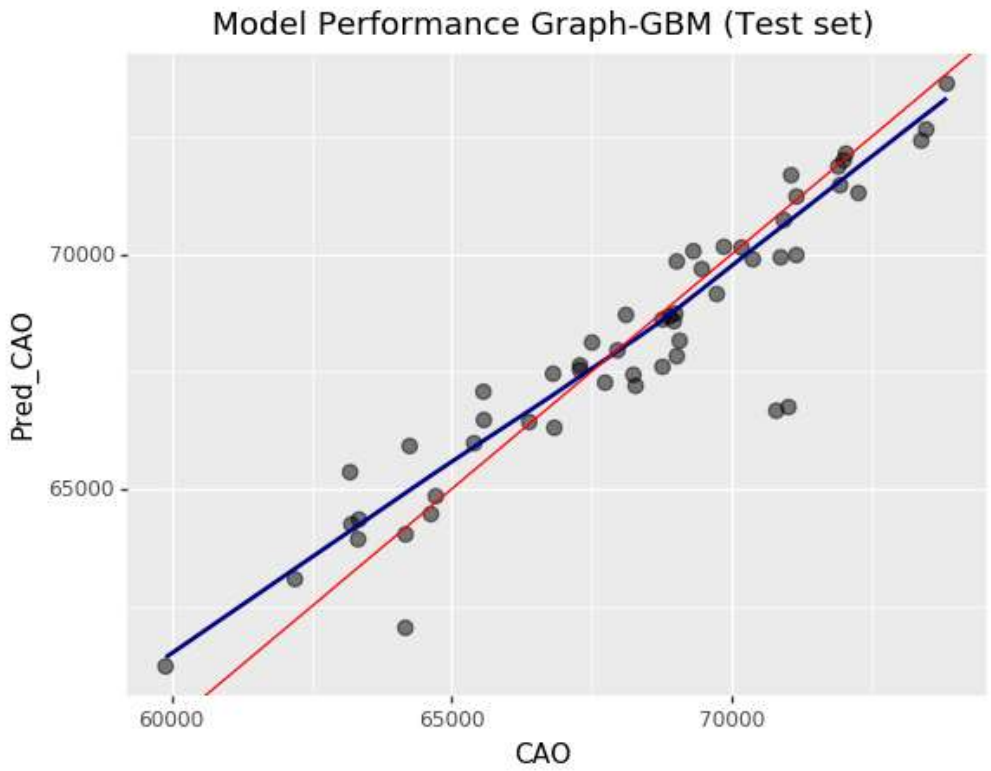


R-squared: 0.98
Mean Squared Error (MSE): 0.02
Root Mean Squared Error (RMSE): 0.16
Mean Absolute Error (MAE): 0.09

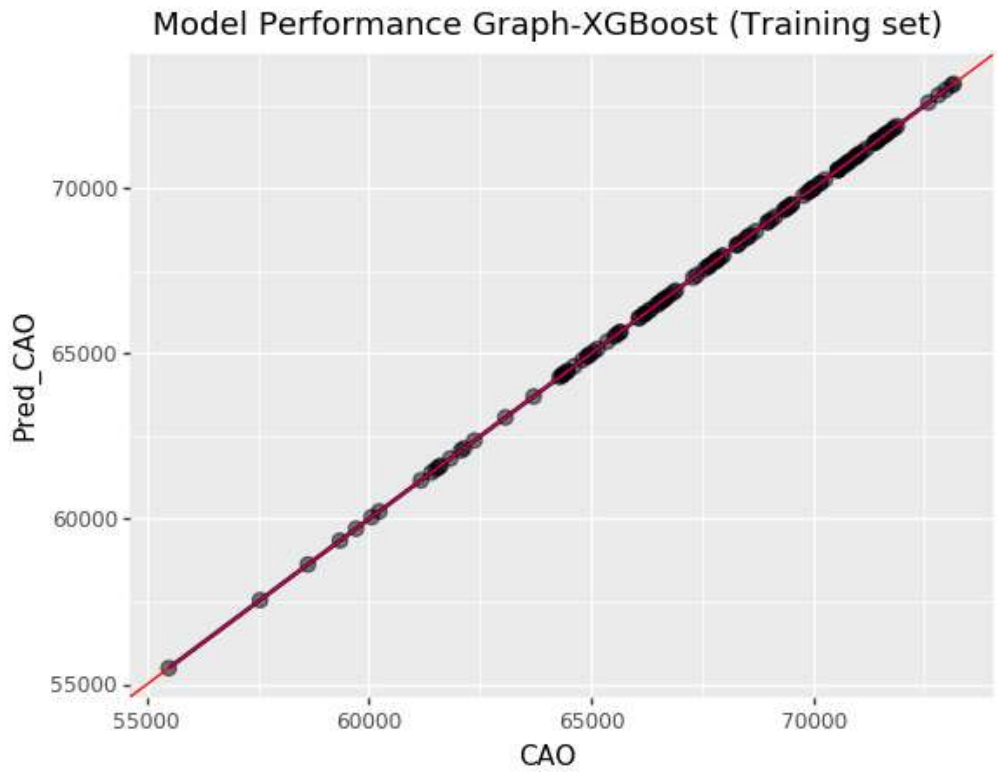
R-squared: 0.86
Mean Squared Error (MSE): 0.14
Root Mean Squared Error (RMSE): 0.38
Mean Absolute Error (MAE): 0.32



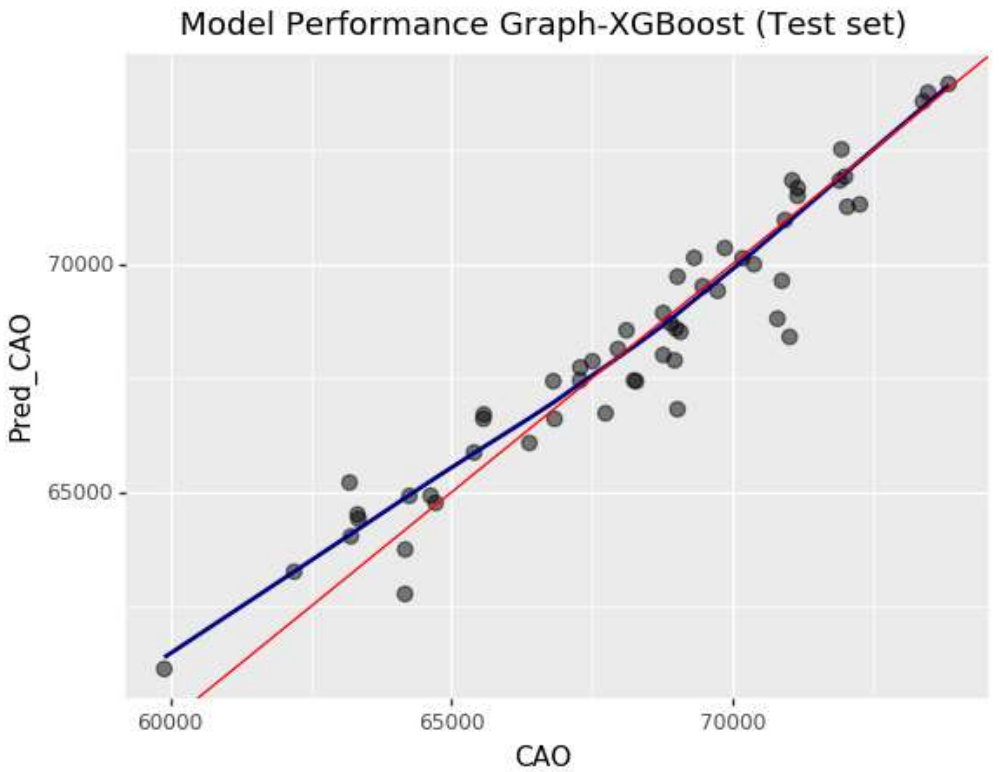
R-squared: 1.00
Mean Squared Error (MSE): 3109.46
Root Mean Squared Error (RMSE): 55.76
Mean Absolute Error (MAE): 36.68



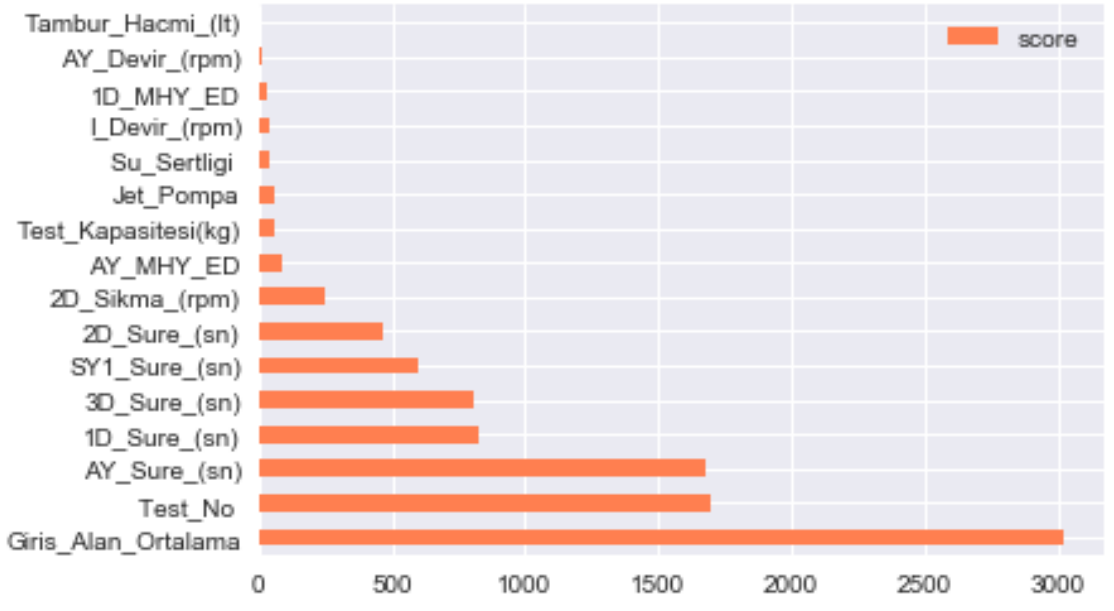
R-squared: 0.87
Mean Squared Error (MSE): 1311147.36
Root Mean Squared Error (RMSE): 1145.05
Mean Absolute Error (MAE): 779.59



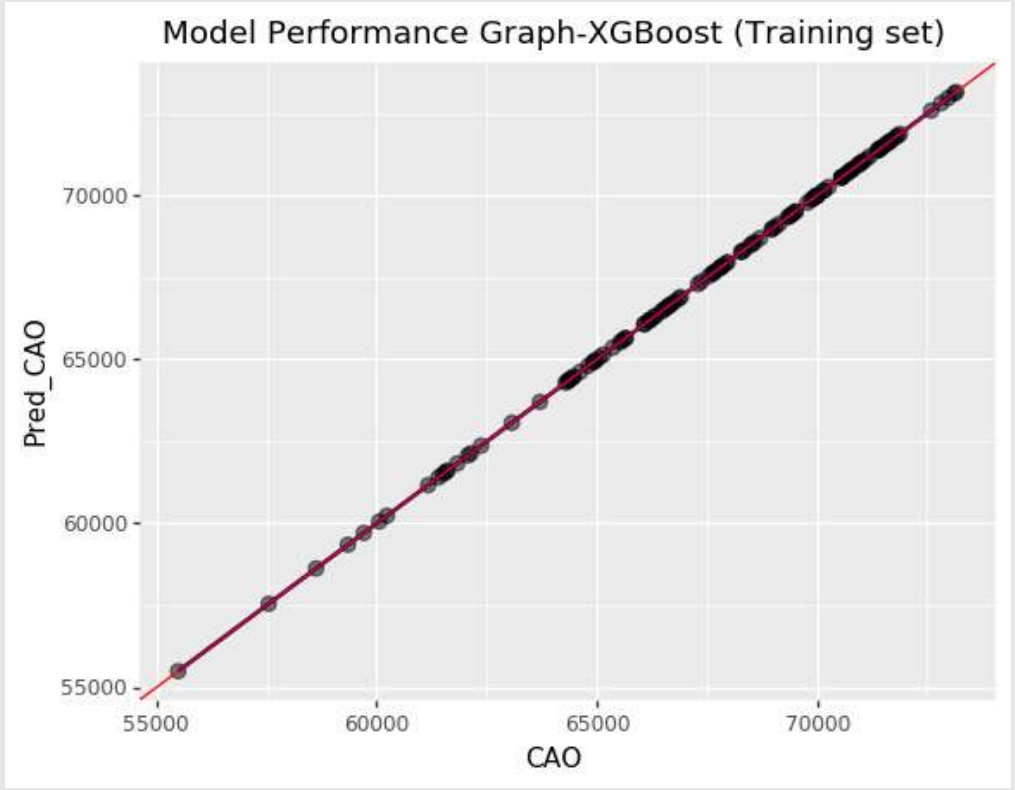
R-squared: 1.00
Mean Squared Error (MSE): 0.93
Root Mean Squared Error (RMSE): 0.97
Mean Absolute Error (MAE): 0.72

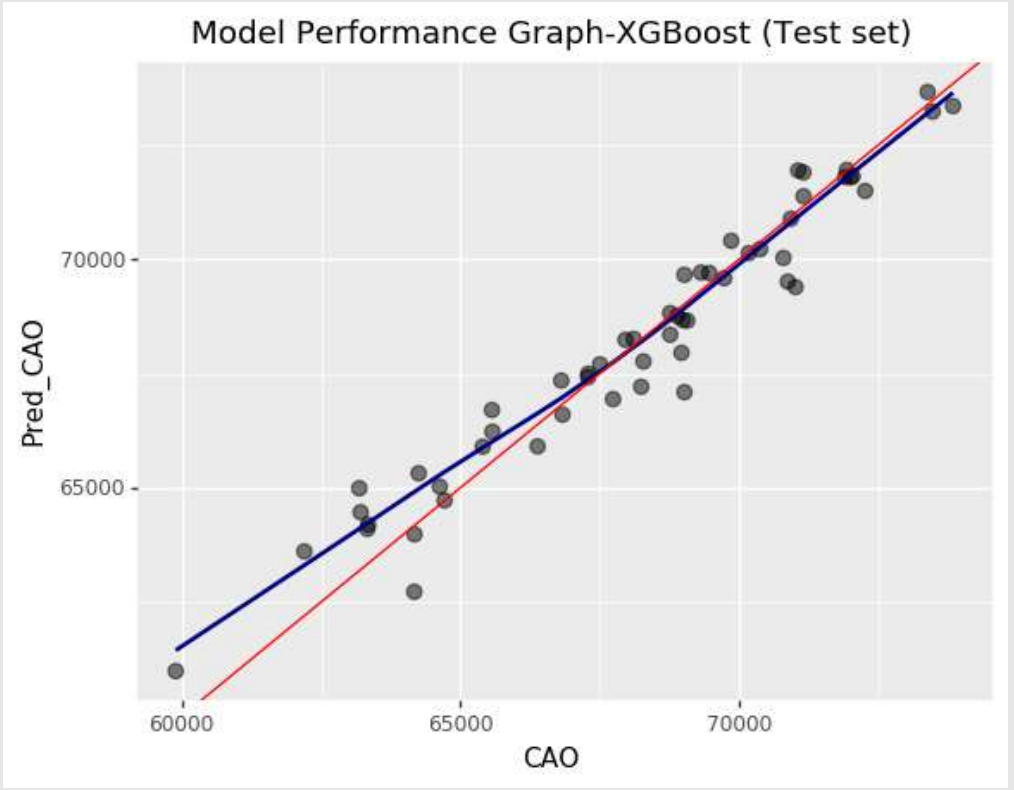


R-squared: 0.92
Mean Squared Error (MSE): 793557.76
Root Mean Squared Error (RMSE): 890.82
Mean Absolute Error (MAE): 684.89




```
#adım 2: p value 0.05 den büyük değişkenler çıkarılır...

#OLS vriabls to remove

df.drop(['Numune_1_ Relakse_Sonrasi_ En_1',
         'Numune_1_ Relakse_Sonrasi_ En_2',
         'Numune_1_ Relakse_Sonrasi_ En_3',
         'Numune_1_Relakse_Sonrasi_Boy_1',
         'Numune_1_Relakse_Sonrasi_Boy_2',
         'Numune_1_Relakse_Sonrasi_Boy_3',
         '2D_MHY_ED',
         '2D_Su_Mik_(lt)',
         'I_Sure_(sn)',
          'I_MHY_ED', #
         'AY_Su_Mik_(lt)',
         'I_Tset_(C)',
          'SG_Sic_(C)', #
          'AY_Tah_(C)', #
         '1D_Su_Mik_(lt)', #
          'I_Bas_Sic (C)', #
          '3D_Su_Mik_(lt)' #

          # 'Tambur_Hacmi_(lt)',
          # '1D_MHY_ED',
          # '2D_Sure_(sn)',
          # '3D_Sure_(sn)',
          # 'I_Devir_(rpm)'
         ], axis=1, inplace=True)
```






```
R-squared: 1.00
Mean Squared Error (MSE): 2.63
Root Mean Squared Error (RMSE): 1.62
Mean Absolute Error (MAE): 1.11
```

```
R-squared: 0.94
Mean Squared Error (MSE): 577380.88
Root Mean Squared Error (RMSE): 759.86
Mean Absolute Error (MAE): 584.50
```

CONCLUSION

- In this project, five machine learning algorithms used for prediction in Python program. Model performance was measured with R^2 score, MSE, RMSE and MAE KPI forecast error types. After examination of KPI values for test samples and also model performance graphs, the XGBoost algorithm was chosen as an optimal model for this case. The best result has been gotten with removing variables that have bigger p-values than significance level(0.05), in that way, test MAE, RMSE and R^2 score values were found 584.50, 759.86, 94% respectively.
- As a result, the XGBoost model predicted well. GBM and SVR model followed it. In other words, the aim of the project could be reached.