Forecasting with Regression Analysis

PROJ 302 Final Report

Summer 2018

Hüseyin Ender Sarı – 24302

Philip Morris SA, Philip Morris Sabancı Pazarlama ve Satış A.Ş

Internship start and end dates: 25.06.18 – 14.09.18

Internship Supervisor: Burak Kocuk

Submission Date: 14/09/18

Faculty of Engineering and Natural Sciences

Industrial Engineering

ABSTRACT

I completed the internship at Philip Morris International/Sabanci. As a leading tobacco company, Philip Morris has a head quarter in Istanbul and a factory in Izmir. [redacted] my project consisted of designing a web-site to forecast the sales figures for all products and retailers in Maslak/Istanbul region. I planned to use regression analysis for 6-month forecast of the demand of the retailers and I had the 18-month of the past demand data. To achieve the best forecast, I tried to analyze the data with different mathematical methods. These are some mathematical methods such as Power, Logarithmic, Exponential, and Least Squares Methods with different degrees. According to these methods, I tried to obtain the best forecast by comparing their relative errors. The best method for any product varies in respect to the data therefore, I had a chance to reach the accurate forecast.

[redacted]

Table of Contents

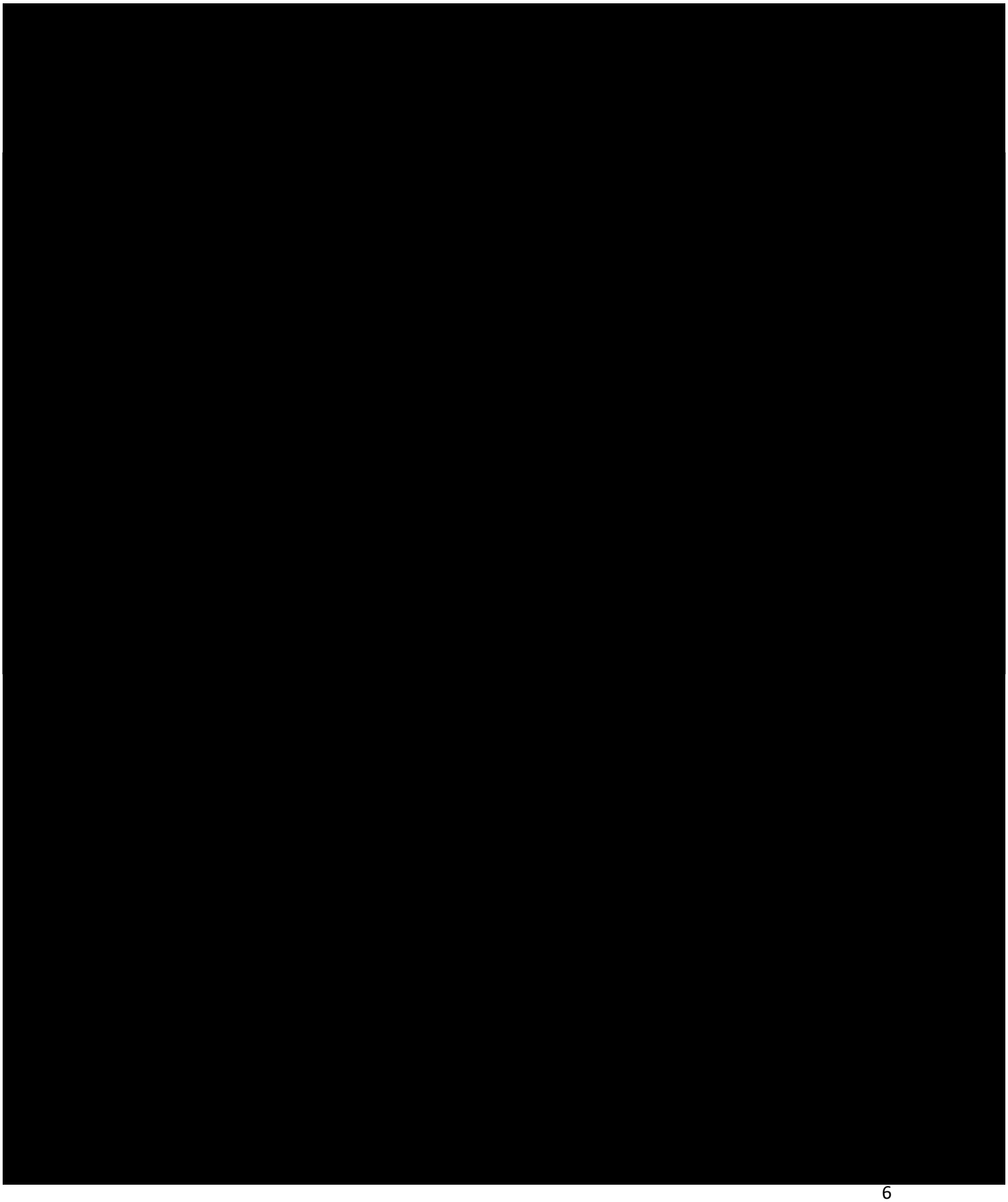
1. Introduction

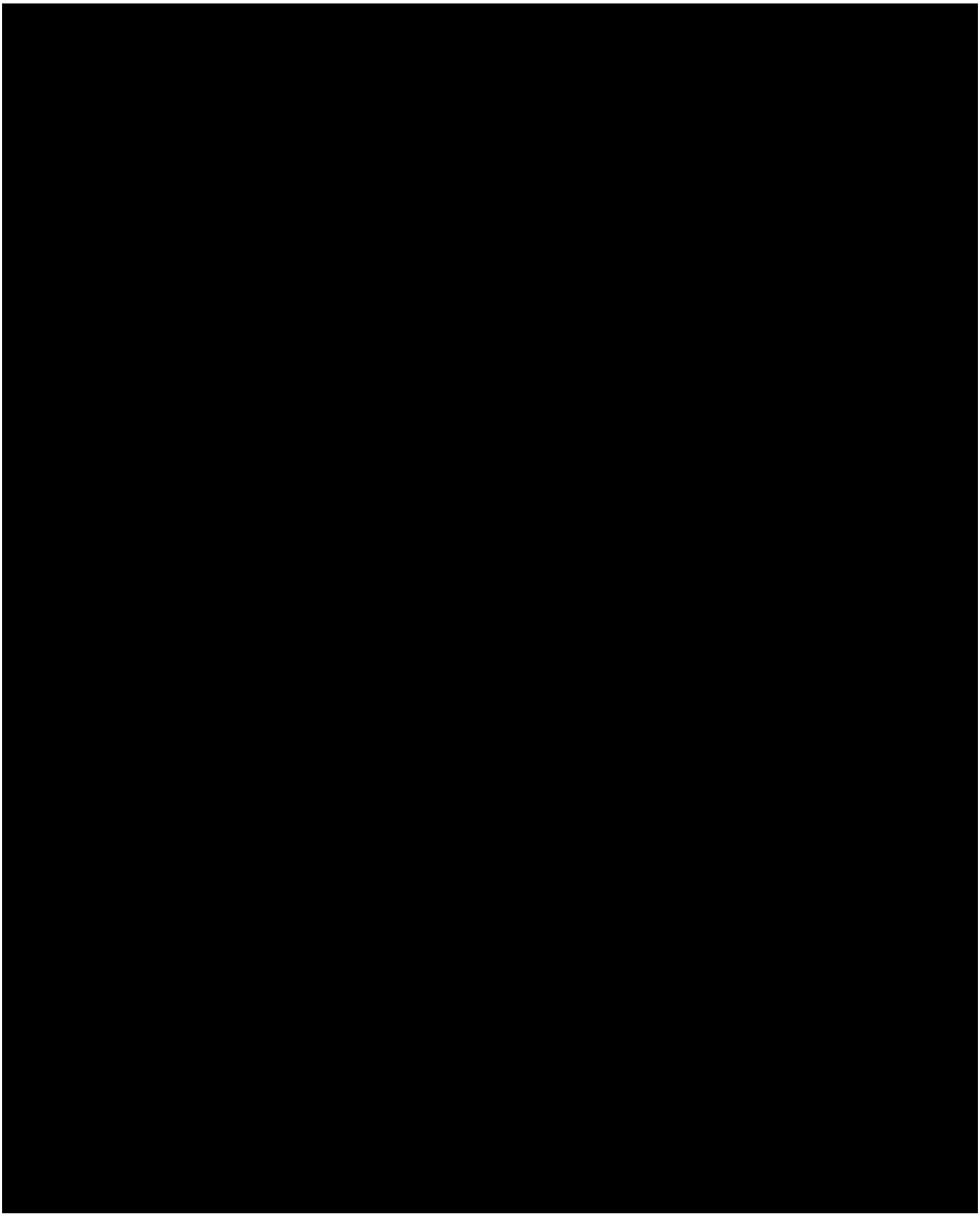
This report allows the reader to obtain information about both Philip Morris International/Sabanci and my internship. Since I completed the internship in 3 months, I had a chance to observe almost all departments.

Besides these, I dedicated my time to self-improvement about coding, analyzing and communication. Most of the time, I spent the time while coding. There were a variety of difficulties, however, I handled many of them. I tried to do my best to decrease the error rate for forecasting.

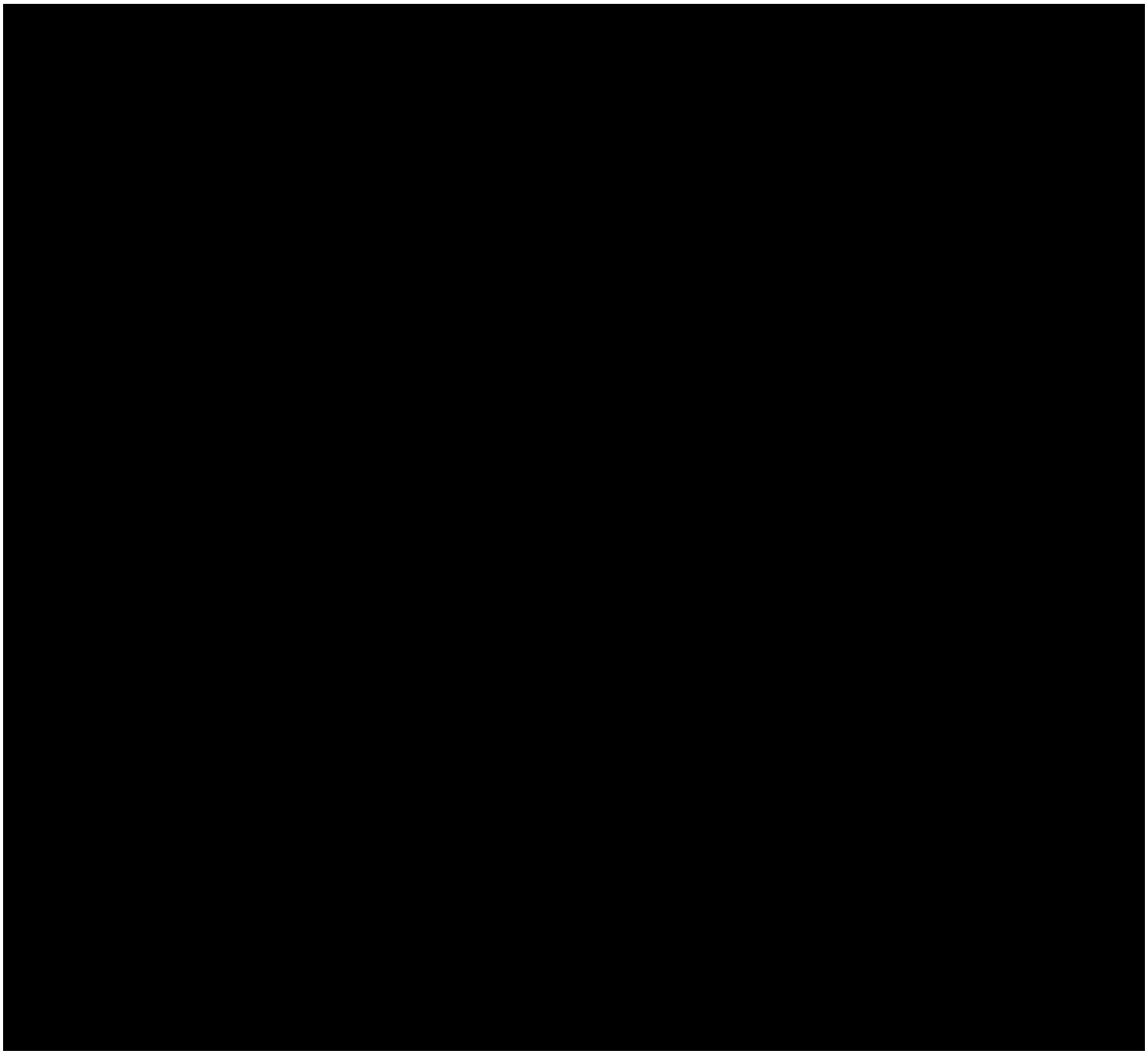
At the end of my project, I designed a local website to forecast the sales figures, also anyone can see the relationship analysis for the products and top ten products in terms of their relative errors by benefiting from six different mathematical methods.

2. Company Information

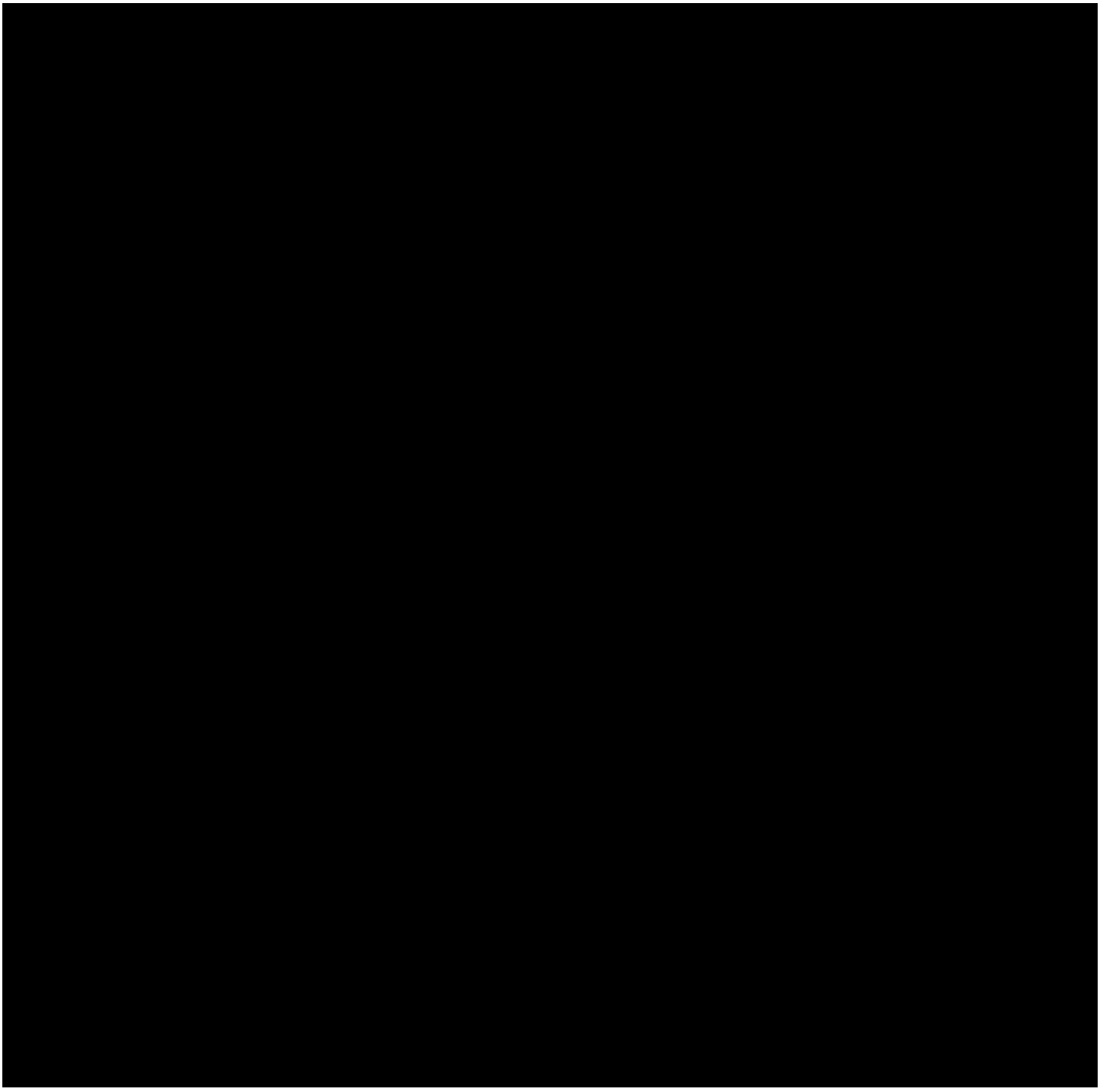




3. Project Background



3.2. Status of the Project or the Problem at the beginning



3.4. Related Literature

The method that is used by the company is simple moving averages. “A moving average is a non-parametric estimation technique that uses weighted averages” (Brocklebank,2018). The disadvantage of the moving average, this method neglects the significant changes and pattern behaviors of the data. Also, there are many forecasting methods such as simple exponential smoothing, Holts method and Winters method. Simple exponential smoothing method can be used for the data which has no patterns. Holts method can be ideal if the data has a trend and Winters method considers both trend and seasonality. (Sobol,1993,p.4) However, for my

forecasting project, I decided to use regression analysis with different mathematical methods to obtain their trendlines and regression equations and compare their relative errors with the original data. The regression equations are necessary because I want to predict the future observations and it gives a general description of the data structure (FARAWAY, 2002, p.7). To close with the original data, there are a variety of lines, but I chose to use Least Squares Method Degree 1, Least Squares Method Degree 2, Least Squares Method Degree 3, Power Method, Logarithmic Method and Exponential Method. The method of Least Squares can obtain the best fit line to the data and this method is prevalent while working on a regression equation. (Miller,2006)

Least Square method degree 1 has a general formula as $y= ax+b$ LSM degree 2 has a general formula $y=ax^2+bx+c$ and degree 3 has $y=ax^3+bx^2+cx+d$. Power Method has a general formula $y=ax^b$. Exponential method has $y = ae^{bx}$. Logarithmic method can be shown as $y = a + b \ln x$. (Mathews,1992,p.253) By obtaining these regression lines, there is a chance to compare them with the original data to find relative error which is calculated as (absolute error/original data). The minimum relative error gives the more accurate result for the forecast. By benefiting from these mathematical equations, it is easy to reach the best forecast.

As a clothing firm Zara also needs forecasting to determine the production rate and stocking the items basically. However, their forecasting methodology consists of mainly external things such as weather condition, specific events , holidays etc. The motivation for me to work for forecasting the PMI products and retailers is that Zara's forecasting has a potential to reduce firm's lost sales by 24%.(García,2014, p.7) I think that it shows the power of the forecasting, but it should be developed with different algorithms.

4. Project Background

4.1. Project Objective

4.2. My Responsibilities

At the beginning, I import and export daily selling data to the business department via SAP. However, after the necessity for the forecasting, I just do it for only my project. The activities that I conducted are:

1-Collecting data

2-Coding to find the regression equations according to different mathematical methods

3-Using SQL for data layer visualstudio

4-Thinking about dashboards, front-end and back-end

4.3. Methodology/ Tools

For the forecasting project, I designed a local website by using different programming languages such as Java, C#, html, CSS. As for framework I used entity framework and as a library I used anychart, bootstrap and math. As two main programs I used SQL Server Management System and Visual Studio.

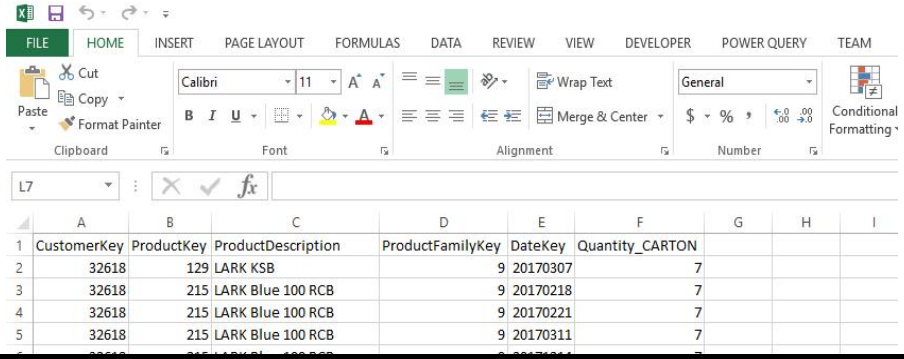
4.4. Expected Outcome and Deliverables

The expectations from my algorithm is that decreasing the error rate, obtaining the accurate forecast compared to simple moving average method. In the future, it can be used as a local forecasting tool for PMSA. The IS analysists would like to improve the tool with adding different mathematical algorithms and methods.

4.5. Details

1-) I took the 18-month data from SAP in excel format. This data was from 2nd January 2017 to 30th June 2018.

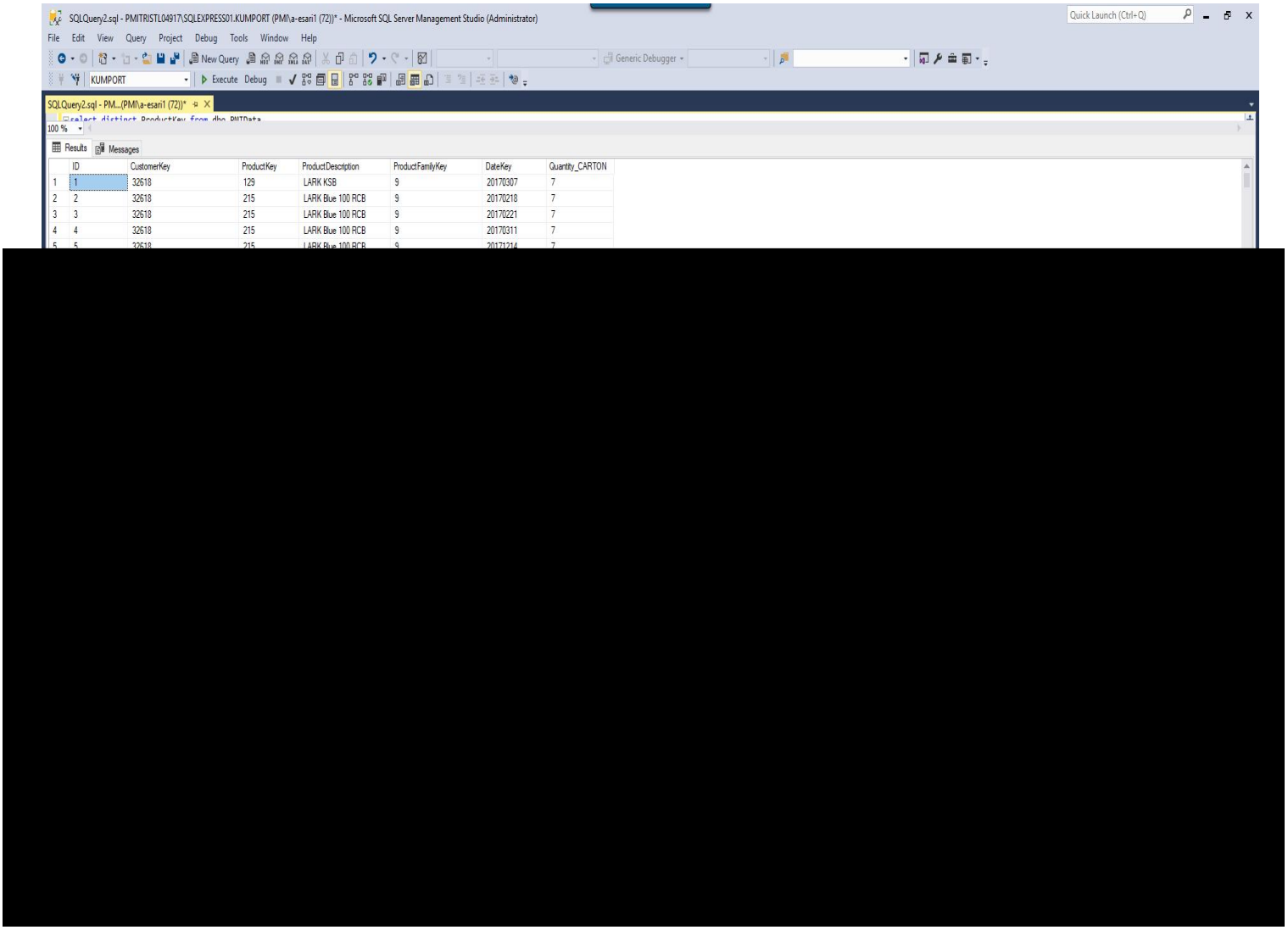
[redacted]



	A	B	C	D	E	F
1	CustomerKey	ProductKey	ProductDescription	ProductFamilyKey	DateKey	Quantity_CARTON
2	32618	129	LARK KSB	9	20170307	7
3	32618	215	LARK Blue 100 RCB	9	20170218	7
4	32618	215	LARK Blue 100 RCB	9	20170221	7
5	32618	215	LARK Blue 100 RCB	9	20170311	7

[redacted]

2-) I imported the excel file to SQL Server Management Studio and I gave ID for all the rows.



	ID	CustomerKey	ProductKey	ProductDescription	ProductFamilyKey	DateKey	Quantity_CARTON
1	1	32618	129	LARK KSB	9	20170307	7
2	2	32618	215	LARK Blue 100 RCB	9	20170218	7
3	3	32618	215	LARK Blue 100 RCB	9	20170221	7
4	4	32618	215	LARK Blue 100 RCB	9	20170311	7
5	5	32618	215	LARK Blue 100 RCB	9	20171214	7

3-) In order to connect the database to my application, I made a scheme. The scheme consisted of different tables that I designed these tables with using entity framework because adding, removing, queueing, joining to database directly was faster. In other words, when I changed something, the database was affected.

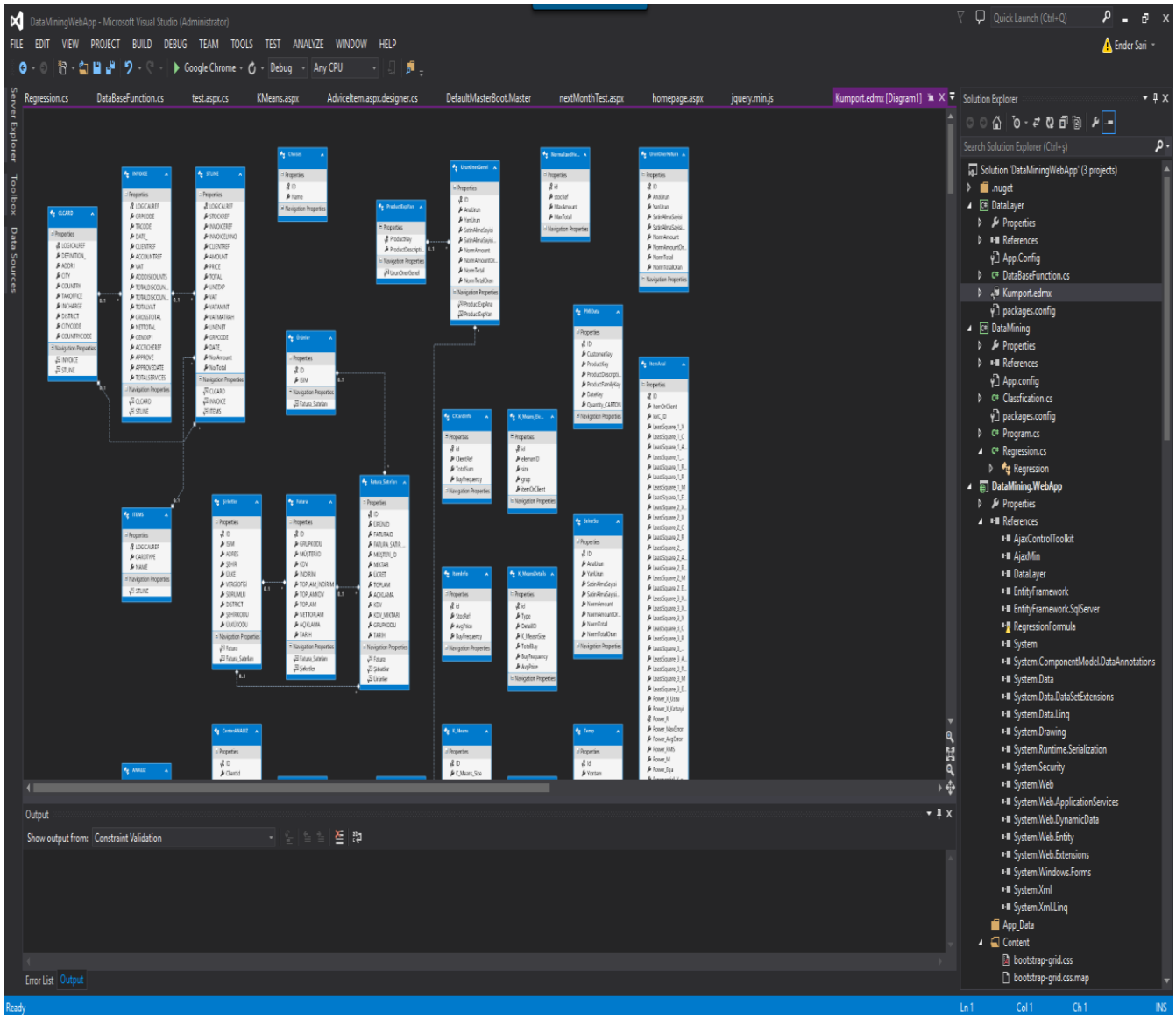


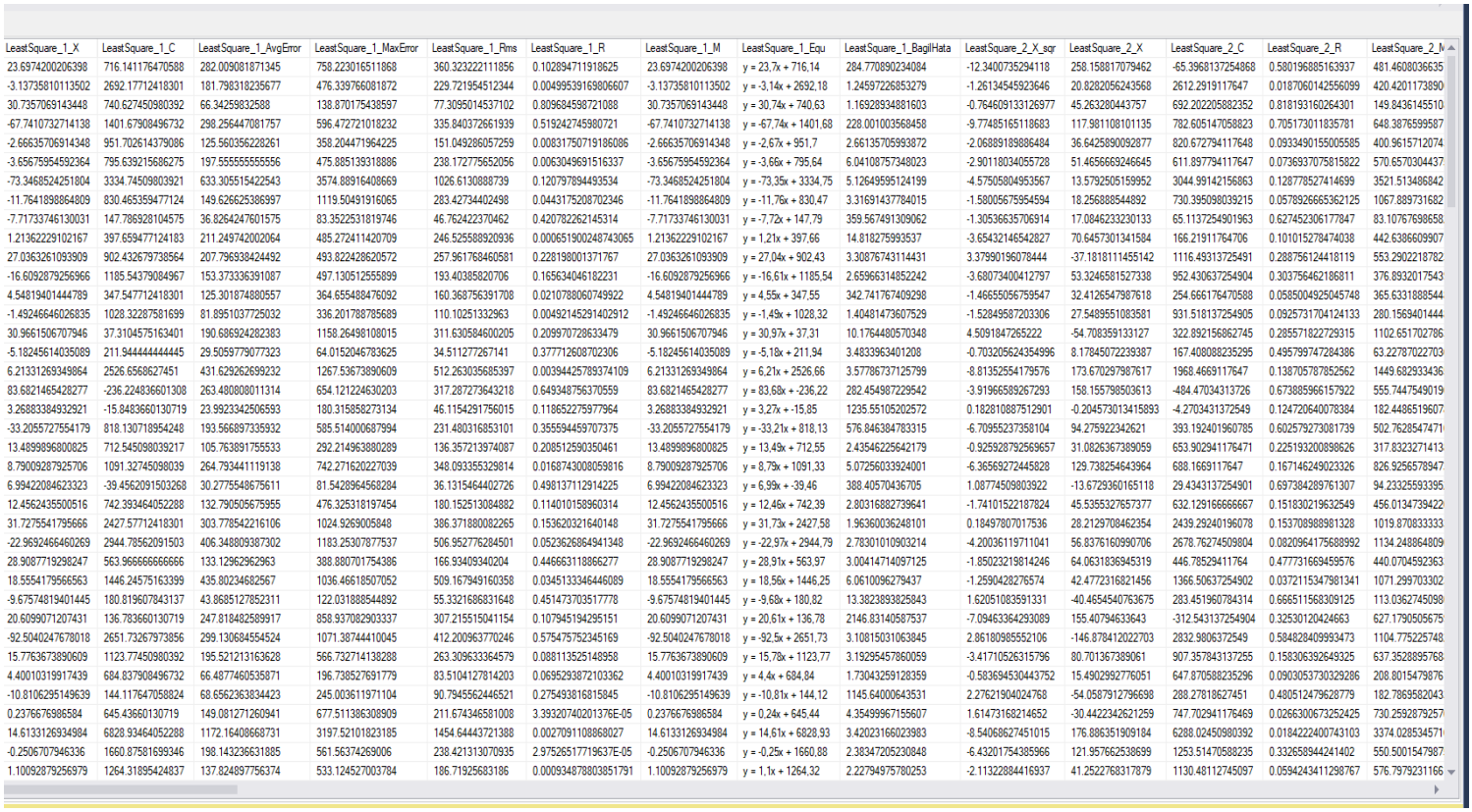
Figure 3. Scheme Entity Framework

4-) According to the mathematical methods, I coded them to find their regression lines. I used C# to find the regression lines. I coded all the mathematical methods as a function to call them. After that, I composed their regression equations and transferred them to the SQL. Also, I added the whole regression and relative error code to the appendix.

```
public ANALIZ Least_Sqaure_1(ANALIZ ANALIZ, double[]x , double[] y )
      {
         // y = ax+b;
         double Total = 0double sumX = 0;
         double SumY = 0;
         double sumX_2 = 0;
         double sumX_Y = 0;
         size=x.length;

         for (int i = 0; i < y.Length; i++)
         {
            sumX += x[i];
            SumY += y[i];
            sumX_2 += Math.Pow(x[i], 2);
            sumX_Y += x[i] * y[i];
         }
         double temp = (sumX_2 * size) - (sumX * sumX);
         double A = (sumX_Y * size - sumX * SumY) / temp;
         double B = (sumX_2 * SumY - sumX * sumX_Y) / temp;

         ANALIZ.LeastSquare_1_X = A;
         ANALIZ.LeastSquare_1_C = B;
         ANALIZ.LeastSquare_1_M = A;
         ANALIZ.LeastSquare_1_Equ = "y = " + Math.Round(A, 2) + "x + " + Math.Round(B, 2);

         return ANALIZ;
      }
```

Figure 4. Least Square Degree 1 Method Code

5-) I consisted all the regression lines and calculated their relative errors by using SQL. My code calculated their coefficients and show the equation inside of the program.



LeastSquare_1_X	LeastSquare_1_C	LeastSquare_1_AvgError	LeastSquare_1_MaxError	LeastSquare_1_Rms	LeastSquare_1_R	LeastSquare_1_M	LeastSquare_1_Equ	LeastSquare_1_BagilHata	LeastSquare_2_X_sqr	LeastSquare_2_X	LeastSquare_2_C	LeastSquare_2_R	LeastSquare_2_M
23.6974200206398	716.141176470588	282.009081871345	758.223016511868	360.323222111856	0.102894711918625	23.6974200206398	y = 23,7x + 716,14	284.770890234084	-12.3400735294118	258.158817079462	-65.3968137254868	0.580196885163937	481.4608036635
-3.13735810113502	2692.17712418301	181.798318235677	476.339766081872	229.721954512344	0.00499539169806607	-3.13735810113502	y = -3,14x + 2692,18	1.24597226853279	-1.26134545923646	20.8282056243568	2612.2919117647	0.0187060142556099	420.4201173890
30.7357069143448	740.627450980392	66.34259832588	138.870175438597	77.3095014537102	0.809684598721088	30.7357069143448	y = 30,74x + 740,63	1.16928934881603	-0.764609133126977	45.263280443757	692.202205882352	0.818193160264301	149.8436145510
-67.7410732714138	1401.67908496732	298.256447081757	596.472721018232	335.840372661939	0.519242745980721	-67.7410732714138	y = -67,74x + 1401,68	228.001003568458	-9.77485165118683	117.981108101135	782.605147058823	0.705173011835781	648.3876599587
-2.66635706914348	951.702614379086	125.560356228261	358.204471964225	151.049286057259	0.00831750719186086	-2.66635706914348	y = -2,67x + 951,7	2.66135705993872	-2.06889189886484	36.6425890092877	820.672794117648	0.0933490155005585	400.9615712074
-3.65675954592364	795.639215686275	197.555555555556	475.885139318886	238.172775652056	0.0063049691516337	-3.65675954592364	y = -3,66x + 795,64	6.04108757348023	-2.90118034055728	51.4656669246645	611.897794117647	0.0736937075815822	570.6570304437
-73.3468524251804	3334.74509803921	633.305515422543	3574.88916408669	1026.61630888739	0.120797894493534	-73.3468524251804	y = -73,35x + 3334,75	5.12649595124199	-4.57505804953567	13.5792505159952	3044.99142156863	0.128778527414699	3521.513486842
-11.7641898864809	830.465359477124	149.626625386997	1119.50491916065	283.42734402498	0.0443175208702346	-11.7641898864809	y = -11,76x + 830,47	3.31691437784015	-1.58005675954594	18.2568885448892	730.395098039215	0.0578926665362125	1067.889731682
-7.71733746130031	147.786928104575	36.8264247601575	83.3522531819746	46.762422370462	0.420782262145314	-7.71733746130031	y = -7,72x + 147,79	359.567491309062	-1.30536635706914	17.0846233230133	65.1137254901963	0.627452306177847	83.10767698658
1.21362229102167	397.659477124183	211.249742002064	485.272411420709	246.525588920936	0.000651900248743065	1.21362229102167	y = 1,21x + 397,66	14.818275993537	-3.65432146542827	70.6457301341584	166.21911764706	0.101015278474038	442.6386609907
27.0363261093909	902.432679738564	207.796938424492	493.822428620572	257.961768460581	0.228198001371767	27.0363261093909	y = 27,04x + 902,43	3.30876743114431	3.37990196078444	-37.1818111455142	1116.49313725491	0.288756124418119	553.2902218782
-16.6092879256966	1185.54379084967	153.373336391087	497.130512555899	193.403858207 06	0.165634064182231	-16.6092879256966	y = -16,61x + 1185,54	2.65966314852242	-3.68073400412797	53.3246581527338	952.430637254904	0.303756462186811	376.8932017543
4.54819401444789	347.547712418301	125.301874880557	364.655488476092	160.368756391708	0.0210788060749922	4.54819401444789	y = 4,55x + 347,55	342.741767409298	-1.46655056759547	32.4126547987618	254.666176470588	0.0585004925045748	365.6331888544
-1.49246646026835	1028.32287581699	81.8951037725032	336.201788785689	110.10251332963	0.00492145291402912	-1.49246646026835	y = -1,49x + 1028,32	1.40481473607529	-1.52849587203306	27.5489551083581	931.518137254905	0.0925731704124133	280.1569401444
30.9661506707946	37.3104575163401	190.686924282383	1158.26498108015	311.630584600205	0.209970728633479	30.9661506707946	y = 30,97x + 37,31	10.1764480570348	4.5091847265222	-54.7083591331 27	322.892156862745	0.285571822729315	1102.651702786
-5.18245614035089	211.944444444445	29.5059779077323	64.0152046783625	34.511277267141	0.377712608702306	-5.18245614035089	y = -5,18x + 211,94	3.4833963401208	-0.703205624354996	8.17845072239387	167.408088235295	0.495799747284386	63.22787022703
6.21331269349864	2526.6568627451	431.629262699232	1267.53673890609	512.263035685397	0.00394425789374109	6.21331269349864	y = 6,21x + 2526,66	3.57786737125799	-8.81352554179576	173.670297987617	1968.4669117647	0.138705787852562	1449.682933436
83.6821465428277	-236.224836601308	263.480808011314	654.121224630203	317.287273643218	0.649348756370559	83.6821465428277	y = 83,68x + -236,22	282.454987229542	-3.91966589267293	158.155798503613	-484.47034313726	0.673885966157922	555.7447549019
3.26883384932921	-15.8483660130719	23.9923342506593	180.315858273134	46.1154291756015	0.118652275977964	3.26883384932921	y = 3,27x + -15,85	1235.55105202572	0.182810887512901	-0.204573013415893	-4.2703431372549	0.124720640078384	182.4486519607
-33.2055727554179	818.130718954248	193.566897335932	585.514000687994	231.480316853101	0.355594459707375	-33.2055727554179	y = -33,21x + 818,13	576.846384783315	-6.70955237358104	94.275922342621	393.192401960785	0.602579273081739	502.7628547471
13.4899896800825	712.545098039217	105.763891755533	292.214963880289	136.357213974087	0.208512590350461	13.4899896800825	y = 13,49x + 712,55	2.43546225642179	-0.925928792569657	31.0826367389059	653.902941176471	0.225193200898626	317.8323271413
8.79009287925706	1091.32745098039	264.793441119138	742.271620227039	348.093355329814	0.0168743008059816	8.79009287925706	y = 8,79x + 1091,33	5.07256033924001	-6.36569272445828	129.738254643964	688.1669117647	0.167146249023326	826.9256578947
6.99422084623323	-39.4562091503268	30.2775548675611	81.5428964568284	36.1315464402726	0.498137112914225	6.99422084623323	y = 6,99x + -39,46	388.40570436705	1.08774509803922	-13.6729360165118	29.4343137254901	0.697384289761307	94.23325593395
12.4562435500516	742.393464052288	132.790505675955	476.325318197454	180.152513084882	0.114010158960314	12.4562435500516	y = 12,46x + 742,39	2.80316882739641	-1.74101522187824	45.5355327657377	632.129166666667	0.151830219632549	456.0134739422
31.7275541795666	2427.57712418301	303.778542216106	1024.9269005848	386.371880082265	0.153620321640148	31.7275541795666	y = 31,73x + 2427,58	1.96360036248101	0.18497807017536	28.2129708462354	2439.29240196078	0.153708988981328	1019.870833333
-22.9692466460269	2944.78562091503	406.348809387302	1183.25307877537	506.952776284501	0.0523626864941348	-22.9692466460269	y = -22,97x + 2944,79	2.78301010903214	-4.20036119711041	56.8376160990706	2678.76274509804	0.0820964175688992	1134.248864809
28.9087719298247	563.966666666666	133.12962962963	388.880701754386	166.93409340204	0.446663118866277	28.9087719298247	y = 28,91x + 563,97	3.00414714097125	-1.85023219814246	64.0631836945319	446.78529411764	0.477731669459576	440.0704592363
18.5554179566563	1446.24575163399	435.80234682567	1036.46618507052	509.167949160358	0.0345133346446089	18.5554179566563	y = 18,56x + 1446,25	6.0610096279437	-1.2590428276574	42.4772316821456	1366.50637254902	0.0372115347981341	1071.299703302
-9.67574819401445	180.819607843137	43.8685127852311	122.031888544892	55.3321686831648	0.451473703517778	-9.67574819401445	y = -9,68x + 180,82	13.3823893825843	1.62051083591331	-40.4654540763675	283.451960784314	0.666511568309125	113.0362745098
20.6099071207431	136.783660130719	247.818482589917	858.937082903337	307.215515041154	0.107945194295151	20.6099071207431	y = 20,61x + 136,78	2146.83140587537	-7.094633642930 89	155.40794633643	-312.543137254904	0.325301204246 63	627.1790505675
-92.5040247678018	2651.73267973856	299.130684554524	1071.38744410045	412.200963770246	0.575475752345169	-92.5040247678018	y = -92,5x + 2651,73	3.10815031063845	2.86180985552106	-146.878412022703	2832.9806372549	0.584828409993473	1104.775225748
15.7763673890609	1123.77450980392	195.521213163628	566.732714138288	263.309633364579	0.088113525148958	15.7763673890609	y = 15,78x + 1123,77	3.19295457860059	-3.41710526315796	80.7013673890 61	907.357843137255	0.158306392649325	637.3528895768
4.40010319917439	684.837908496732	66.4877460535871	196.738527691779	83.5104127814203	0.0695293872103362	4.40010319917439	y = 4,4x + 684,84	1.73043259128359	-0.583694530443752	15.4902992776051	647.870588235296	0.0903053703029286	208.8015479876
-10.8106295149639	144.117647058824	68.6562363834423	245.003611971104	90.7945562446521	0.275493816815845	-10.8106295149639	y = -10,81x + 144,12	1145.64000643531	2.27621904024768	-54.0587912796698	288.27818627451	0.480512479628779	182.7869582043
0.2376676986584	645.436601307 19	149.081271260941	677.511386308909	211.674346581008	3.39320740201376E-05	0.2376676986584	y = 0,24x + 645,44	4.35499967155607	1.61473168214652	-30.4422342621259	747.702941176469	0.0266300673252425	730.2592879257
14.6133126934984	6828.93464052288	1172.16408668731	3197.52101823185	1454.64443721388	0.00270911088680 27	14.6133126934984	y = 14,61x + 6828,93	3.42023166023983	-8.540686274510 15	176.886351909184	6288.02450980392	0.0184222400743103	3374.0285345 71
-0.2506707946336	1660.87581699346	198.143236631885	561.56374269006	238.421313070935	2.97526517719637E-05	-0.2506707946336	y = -0,25x + 1660,88	2.38347205230848	-6.43201754385966	121.957662538699	1253.51470588235	0.332658944241402	550.5001547987
1.10092879256979	1264.31895424837	137.824897756374	533.124527003784	186.71925683186	0.000934878803851791	1.10092879256979	y = 1,1x + 1264,32	2.22794975780253	-2.11322884416937	41.2522768317879	1130.48112745097	0.0594243411298767	576.7979231166

Figure 5. Regression Equations

6-) I would like to show the working principle of forecasting with regression analysis. The homepage consisted of three parts which are trendline, find the product and top ten.

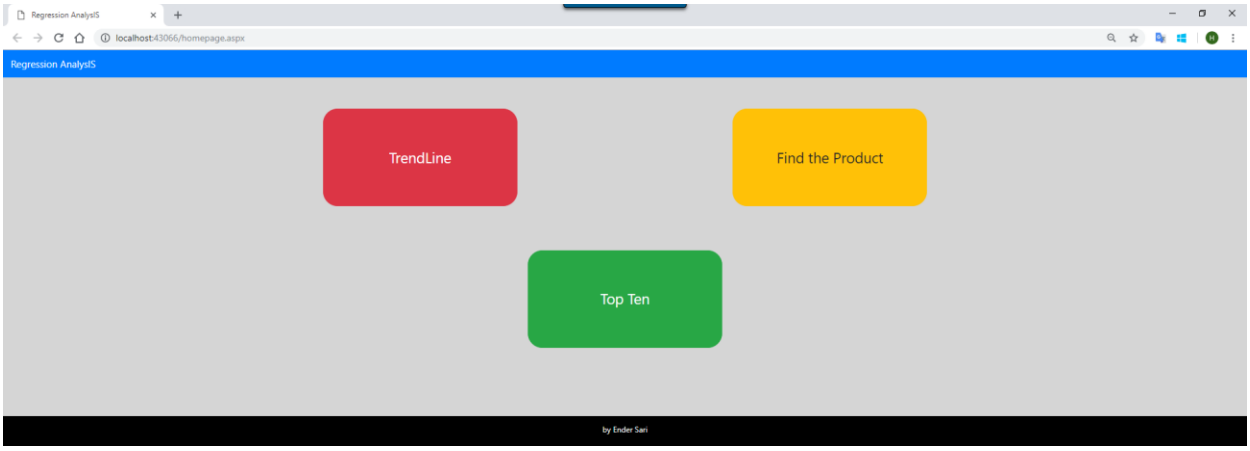


Figure 6. Homepage

When click the Trendline button, you can see the regression equations, the relative errors and the forecast for both retailer and any product. This gives the trendlines and equations for customer ID is [redacted]

Figure 7. Regression Graphs

For the product forecasting, there are similar screens but I consisted a sliding for products.

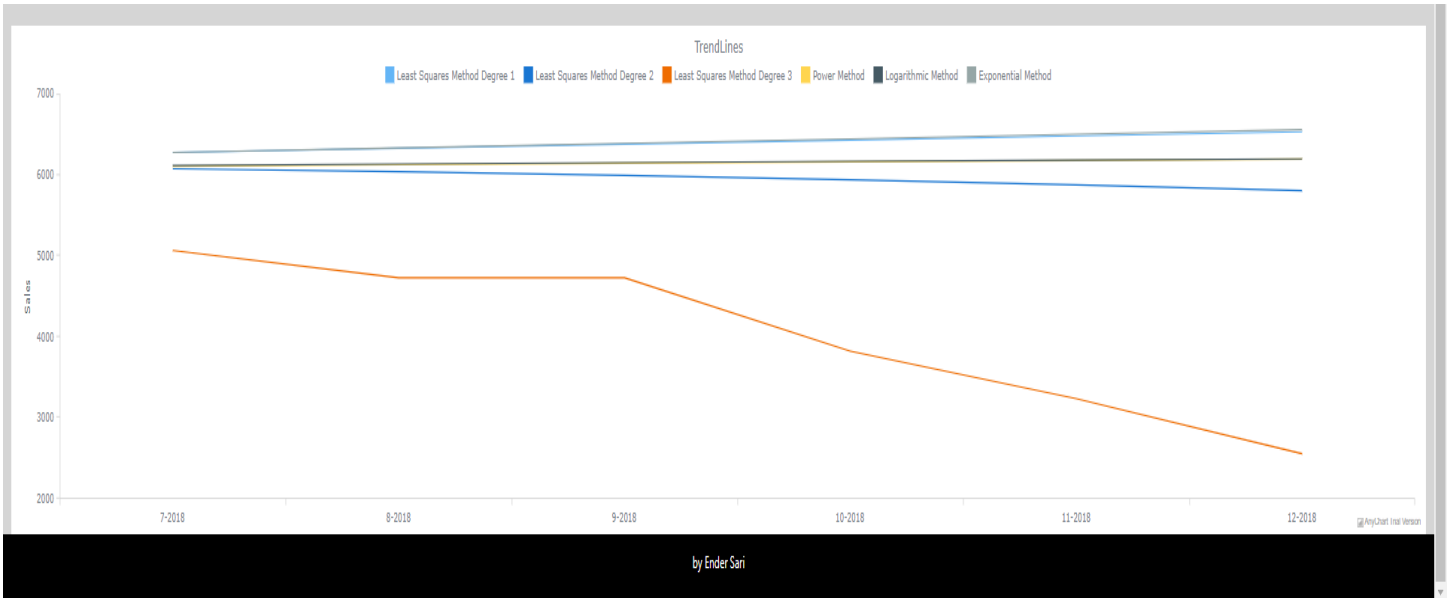


Figure 8. Forecasting

For the forecasting from June to December 2018, there is a graph and it gives for the forecasting for all methods.

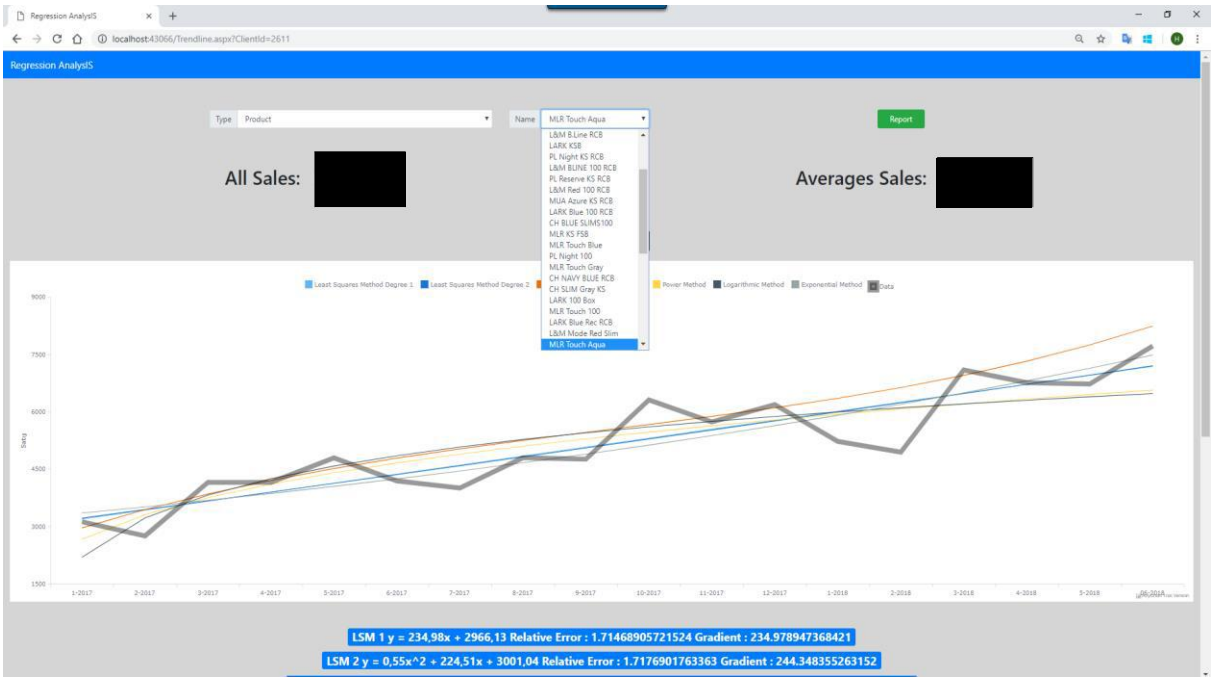


Figure 9. Regression Graphs for Marlboro Touch Aqua

Figure 10. Forecasting for Marlboro Touch Aqua

7-) When click the button find the product from homepage, it is a relationship analysis and it shows that the product which is purchased by any retailer with the products existing the right column and it also gives purchasing frequency. I made this analysis because it is necessary to know which products should be loaded to the van together to reduce the cost.

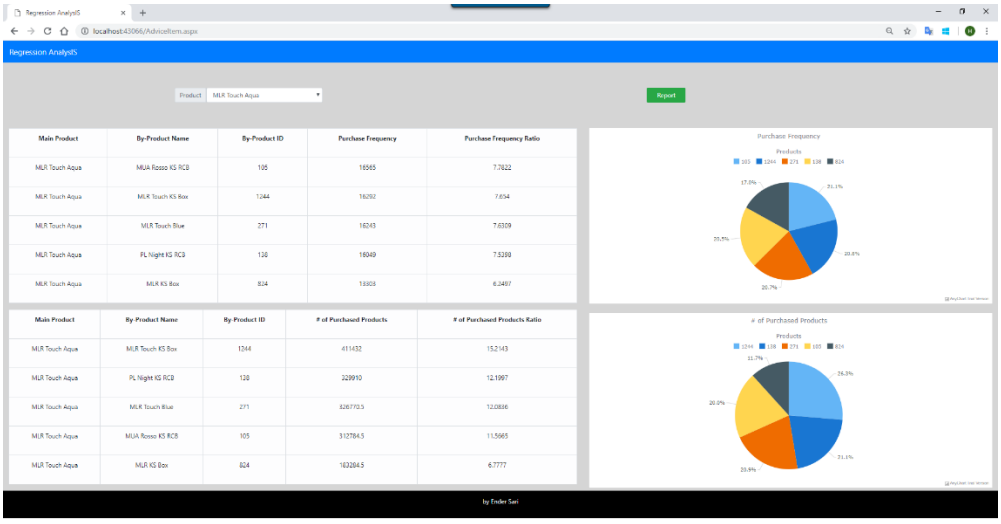


Figure 10. Relationship Analysis

8-) For the top ten button, both the retailers and products were examined in respect to reduced ones, increasing ones and the most successful ones according to their relative error rates according to these different mathematical methods. Also, there is a sliding to choose the method. The screen can be shown like

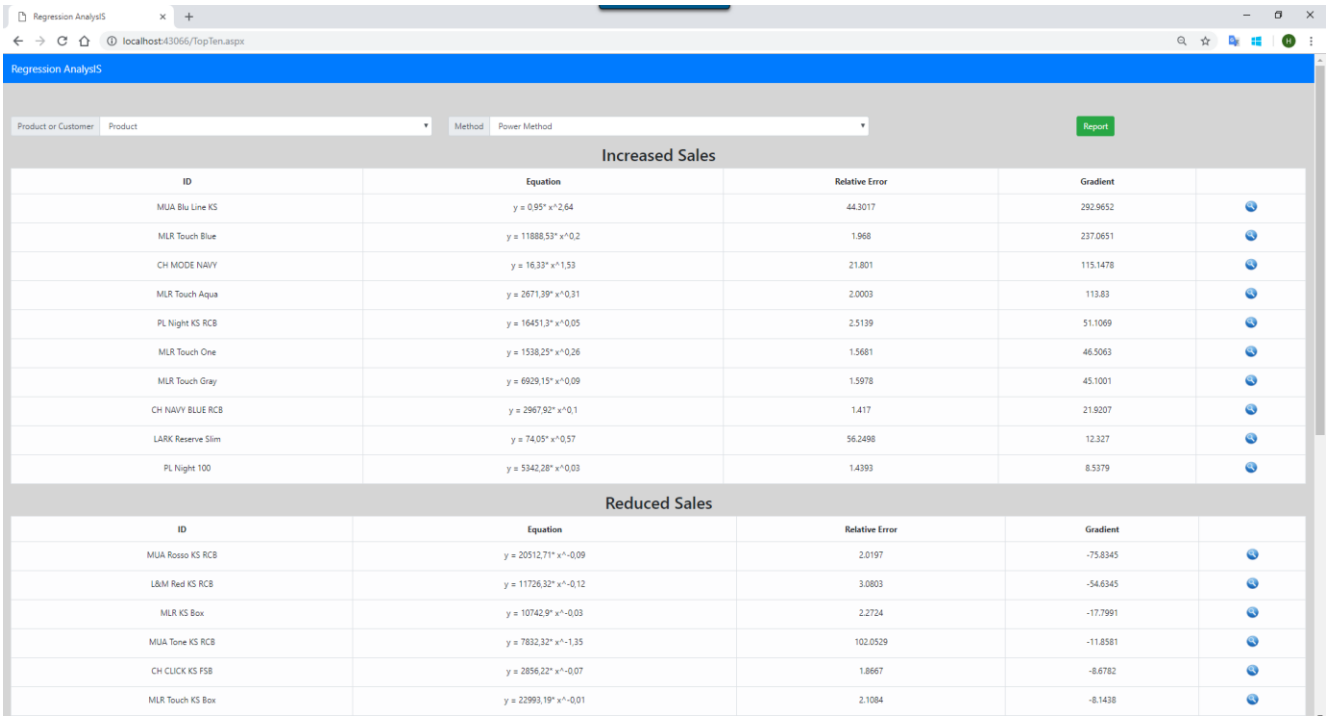


Figure 11. Top Ten products

4.6. Results

I completed my project successfully and my forecasting with regression analysis project help them to think forecasting from different perspectives.

5. Internship Experience

5.1. Learning

I learned many different things during my internship. As for the technical perspective, I contributed to myself by improving my coding skills and merged them with the forecasting. Besides that, after the three months internship period, I learned how to use SQL Server Management System. My career plan didn't change. Still, I would like to get a master's degree with data science.

5.2. Relation to undergraduate education

Knowledge from the computing classes such as CS201 and CS204 were very useful for me. Since the content for these classes was C++, the programming algorithms continue same, just the syntax changes. Besides these, IE401 and ENS208 had the forecasting content. By thinking the learnings from these classes I tried to do something different. There is a database course (CS306), I wish I learned the content at the university, because this class is about SQL and it was very useful during my internship.

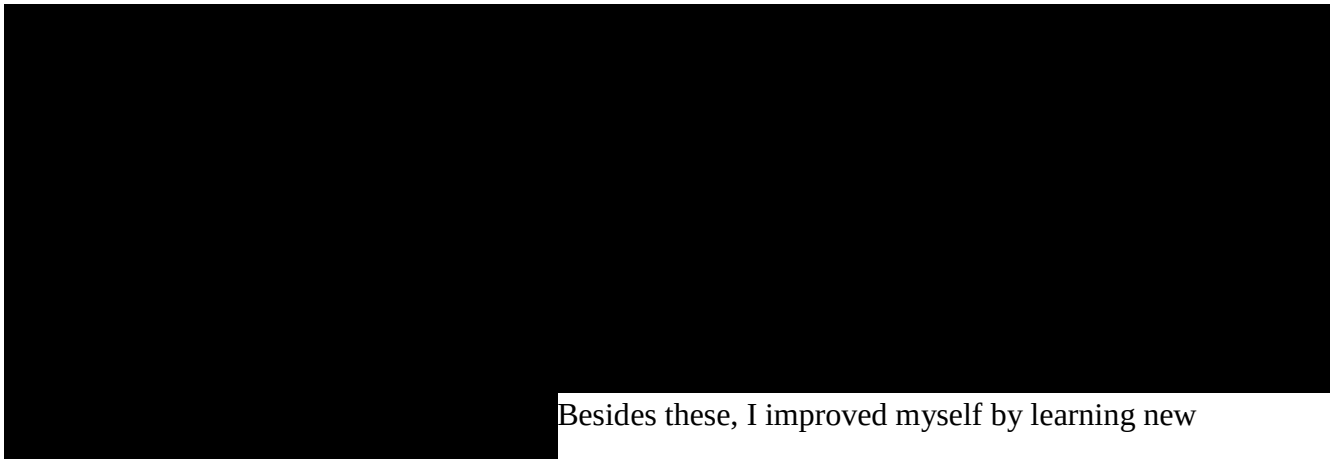
5.3. Difficulties

The first difficulty for me, I couldn't connect the SQL with the Visual Studio, there was a database error. To fix this, I took an admin role from the global support team. It lasted around 5 days. The second challenge for me, while coding, I didn't have any proper plan to code because it was my first experience since I tried to design a local website, I spent so much time to learn the things. Thirdly, I tried to code front-end and back-end simultaneously, when changing a piece of code from back-end I lost some of the features therefore before the changing something I started to back up.

5.4. A Typical Day

I started work at 7.45 in the morning. After 2 hours I was giving a tea break. Some of the time I was going to eat at 12.00. However, If I had a job or a meeting, I was going to eat at 12.30. According to our descent hour, we were one hour away. Typically, I spent time coding the forecasting program. Generally, I worked until 15.00 then again I gave a tea break about 10 or 15 minutes, then I could leave at 17.15.

6. Conclusions



Besides these, I improved myself by learning new programming languages.

7. Recommendations

I think that future Project 302 students should apply global companies to understand the corporate life, because it is a really good way to feel it. For the assessment and interview days, they shouldn't support their one persona, they should try to give information about themselves from different respects. Also, they should be ready to work outside of their comfort zone.

As for work culture considerations, as far as I understand from this internship period, some of the time, you may feel hierarchy but some of the time you may feel fully free. It depends rely on the situation. However, Philip Morris International/Sabanci is a good place to become an intern but you should be ready for the challenges.

8. References

1-Mathews, J. M., 1992. Numerical Method for Mathematics, Science, and Engineering, Prentice-Hall Inc, 253-277

2-FARAWAY, J. J. (2002). Practical Regression and Anova using R. Bath: University of Bath.

3-García, J.M. (2014). Demand Forecasting at Zara : A Look at Seasonality , Product Lifecycle and.

4-Brocklebank, J. C. (2018). *SAS for Forecasting Time Series, Third Edition, 3rd Edition*. S.l.: SAS Institute.

5-Sobol, M.G. (1993) . A Simple Method to Adjust Exponential Smoothing Forecasts for Trend and Seasonality : Southern Methodist University

6- Millet, S. J. (2006) The Method of Least Squares : Brown University

7- Philip Morris Sabancı – Departments (2018)

9. Appendices

(Figure 4. Least Square Degree 1 Method Code) – All codes for the regression and relative error

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using MathNet.Numerics.LinearAlgebra;
using MathNet.Numerics.LinearAlgebra.Double;
using DataLayer;

namespace DataMining
{
   public class Regression
   {
      private static Regression instance;

      private Regression()
      {
      }


      public static Regression GetInstance()
      {
         lock (typeof(Regression))
         {
            if (instance == null)
            {
               instance = new Regression();
            }
            return instance;
         }
      }


      public ANALIZ MakeAllTrendline(ANALIZ ANALIZ, double[] x, double[] OrjData)
      {
         ANALIZ = Least_Sqaure_1(ANALIZ, x, OrjData);
         ANALIZ = Least_Sqaure_2(ANALIZ, x, OrjData);
         ANALIZ = Least_Sqaure_3(ANALIZ, x, OrjData);
         ANALIZ = Power_Method(ANALIZ, x, OrjData);
         ANALIZ = Exponential(ANALIZ, x, OrjData);
         ANALIZ = Logarithmic_Method(ANALIZ, x, OrjData);
         ANALIZ = Equations(ANALIZ, OrjData);

         return ANALIZ;
      }

      public struct errors
      {
         public double max;
         public double avg;
         public double Rms;
         public double SSR;
         public double SSE;
         public double SSTO;
         public double R;
         public double R_error;
      }

      public errors FoundError(double[] OrjData, double[] RegData)
      {
         double mean = 0;
         errors Error = new errors();
```

```
    Error.avg = 0;
    Error.Rms = 0;
    Error.R = 0;
    Error.max = 0;
    Error.R_error = 0;
    foreach (double veri in OrjData)
    {
        mean += veri;
    }
    mean = mean / OrjData.Length;


    for (int i = 0; i < OrjData.Length; i++)
    {
        if (Math.Abs(OrjData[i] - RegData[i]) > Error.max)
        {
            Error.max = Math.Abs(OrjData[i] - RegData[i]);
        }
        Error.avg += Math.Abs(OrjData[i] - RegData[i]);
        Error.Rms += Math.Pow((OrjData[i] - RegData[i]), 2);
        Error.SSR += Math.Pow((RegData[i]-mean),2);  // total
        Error.SSE += Math.Pow((OrjData[i] - RegData[i]), 2); //hata
        Error.SSTO += Math.Pow((OrjData[i] - mean), 2);
        Error.R_error += (Math.Abs(OrjData[i] - RegData[i])) / OrjData[i];
        //Error.R_error += (Math.Abs(OrjData[i] - RegData[i])) / RegData[i];


    }
    Error.R = 1 - (Error.SSE / Error.SSTO);  // orj
    Error.avg = Error.avg / OrjData.Length;
    Error.Rms = Math.Pow(Error.Rms / OrjData.Length, 0.5);
    return Error;

}


public ANALIZ Least_Sqaure_1(ANALIZ ANALIZ, double[]x , double[] y )
{
    // y = ax+b;
    double Total = 0;
    double sumX = 0;
    double SumY = 0;
    double sumX_2 = 0;
    double sumX_Y = 0;
    int size = 18;

    for (int i = 0; i < y.Length; i++)
    {
        sumX += x[i];
        SumY += y[i];
        sumX_2 += Math.Pow(x[i], 2);
        sumX_Y += x[i] * y[i];
    }
    double temp = (sumX_2 * size) - (sumX * sumX);
    double A = (sumX_Y * size - sumX * SumY) / temp;
    double B = (sumX_2 * SumY - sumX * sumX_Y) / temp;

    ANALIZ.LeastSquare_1_X = A;
    ANALIZ.LeastSquare_1_C = B;
    ANALIZ.LeastSquare_1_M = A;
    ANALIZ.LeastSquare_1_Equ = "y = " + Math.Round(A, 2) + "x + " + Math.Round(B, 2);

    return ANALIZ;
}


public ANALIZ Least_Sqaure_2(ANALIZ ANALIZ, double[]x , double[] y)
{
    // y =ax'2+bx+c  Y türev = 2*a*x+b
    double[] Sum_X = new double[6];
```

```
    double SumY = 0;
    double sumX_Y = 0;
    double sum_x_2_y = 0;
    int size = 18;

    for (int i = 0; i < y.Length; i++)
    {
        for (int a = 1; a < 5; a++)
        {
            Sum_X[a] += Math.Pow(x[i], a);
        }
        sumX_Y += x[i] * y[i];
        SumY += y[i];
        sum_x_2_y += y[i] * x[i] * x[i];

    }

    #region matrix
    Matrix<double> temp = DenseMatrix.OfArray(new double[,] {
{Sum_X[4],Sum_X[3],Sum_X[2]},
{Sum_X[3],Sum_X[2],Sum_X[1]},
{Sum_X[2],Sum_X[1],size}});

    Matrix<double> A_Matrix = DenseMatrix.OfArray(new double[,] {
{Sum_X[3],Sum_X[2],sum_x_2_y},
{Sum_X[2],Sum_X[1],sumX_Y},
{Sum_X[1],size,SumY}});

    Matrix<double> B_Matrix = DenseMatrix.OfArray(new double[,] {
{Sum_X[4],Sum_X[2],sum_x_2_y},
{Sum_X[3],Sum_X[1],sumX_Y},
{Sum_X[2],size,SumY}});

    Matrix<double> C_Matrix = DenseMatrix.OfArray(new double[,] {
{Sum_X[4],Sum_X[3],sum_x_2_y},
{Sum_X[3],Sum_X[2],sumX_Y},
{Sum_X[2],Sum_X[1],SumY}});

    #endregion

    double Temp_Det = temp.Determinant();
    double A = A_Matrix.Determinant() / Temp_Det;
    double B = -B_Matrix.Determinant() / Temp_Det;
    double C = C_Matrix.Determinant() / Temp_Det;

    ANALIZ.LeastSquare_2_X_sqr = A;
    ANALIZ.LeastSquare_2_X = B;
    ANALIZ.LeastSquare_2_C = C;
    ANALIZ.LeastSquare_2_M = 2 * A * 18 + B;
    ANALIZ.LeastSquare_2_Equ = "y = " + Math.Round(A, 2) + "x^2 + " + Math.Round(B, 2)+"x + "+Math.Round(C,2);

    return ANALIZ;
}


public ANALIZ Least_Sqaure_3(ANALIZ ANALIZ, double[] x, double[] y)
{
    //ax'3+bx'2+cx+d   y'= 3ax'2+2bx+c
    double[] Sum_X = new double[18];
    double SumY = 0;
    double sum_y_x_2 = 0;
    double sum_y_x_3 = 0;
    double sum_y_x_1 = 0;
    int size = 18;

    for (int i = 0; i < y.Length; i++)
    {
        for (int a = 1; a < 18; a++)
        {
            Sum_X[a] += Math.Pow(x[i], a);
```

```
            }
            sum_y_x_1 += x[i] * y[i];
            SumY += y[i];
            sum_y_x_2 += y[i] * x[i] * x[i];
            sum_y_x_3 += y[i] * Math.Pow(x[i], 3);
        }

        #region matrix
        Matrix<double> temp = DenseMatrix.OfArray(new double[,] {
    {Sum_X[6],Sum_X[5],Sum_X[4],Sum_X[3]},
    {Sum_X[5],Sum_X[4],Sum_X[3],Sum_X[2]},
    {Sum_X[4],Sum_X[3],Sum_X[2],Sum_X[1]},
    {Sum_X[3],Sum_X[2],Sum_X[1],size}});

        Matrix<double> A_Matrix = DenseMatrix.OfArray(new double[,] {
    {Sum_X[5],Sum_X[4],Sum_X[3],sum_y_x_3},
    {Sum_X[4],Sum_X[3],Sum_X[2],sum_y_x_2},
    {Sum_X[3],Sum_X[2],Sum_X[1],sum_y_x_1},
    {Sum_X[2],Sum_X[1],size,SumY}});

        Matrix<double> B_Matrix = DenseMatrix.OfArray(new double[,] {
    {Sum_X[6],Sum_X[4],Sum_X[3],sum_y_x_3},
    {Sum_X[5],Sum_X[3],Sum_X[2],sum_y_x_2},
    {Sum_X[4],Sum_X[2],Sum_X[1],sum_y_x_1},
    {Sum_X[3],Sum_X[1],size,SumY}});

        Matrix<double> C_Matrix = DenseMatrix.OfArray(new double[,] {
    {Sum_X[6],Sum_X[5],Sum_X[3],sum_y_x_3},
    {Sum_X[5],Sum_X[4],Sum_X[2],sum_y_x_2},
    {Sum_X[4],Sum_X[3],Sum_X[1],sum_y_x_1},
    {Sum_X[3],Sum_X[2],size,SumY}});

        Matrix<double> D_Matrix = DenseMatrix.OfArray(new double[,] {
    {Sum_X[6],Sum_X[5],Sum_X[4],sum_y_x_3},
    {Sum_X[5],Sum_X[4],Sum_X[3],sum_y_x_2},
    {Sum_X[4],Sum_X[3],Sum_X[2],sum_y_x_1},
    {Sum_X[3],Sum_X[2],Sum_X[1],SumY}});

        #endregion

        double Temp_Det = temp.Determinant();
        double A = -A_Matrix.Determinant() / Temp_Det;
        double B = B_Matrix.Determinant() / Temp_Det;
        double C = -C_Matrix.Determinant() / Temp_Det;
        double D = D_Matrix.Determinant() / Temp_Det;

        ANALIZ.LeastSquare_3_X_kup = A;
        ANALIZ.LeastSquare_3_X_sqr = B;
        ANALIZ.LeastSquare_3_X = C;
        ANALIZ.LeastSquare_3_C = D;
        ANALIZ.LeastSquare_3_M = (3 * A * 18 * 18) + (2 * B * 18) + C;
        ANALIZ.LeastSquare_3_Equ = "y = " + Math.Round(A, 2) + "x^3 + " + Math.Round(B, 2) + "x^2 + " + Math.Round(C, 2)+"x
+"+Math.Round(D,2);


        return ANALIZ;

    }


    public ANALIZ Logarithmic_Method(ANALIZ ANALIZ, double[] x, double[] y)
    {
        //logarithmic trendline y = a*ln(x) + b.
        double sumX = 0;
        double SumY = 0;
        double sumX_2 = 0;
        double sumX_Y = 0;
        int size = 18;

        double[] NewX = new double[18];
```

```
    for (int a = 0; a < 18; a++)
    {
        NewX[a] = Math.Log(x[a]);
    }
    for (int i = 0; i < y.Length; i++)
    {
        sumX += NewX[i];
        SumY += y[i];
        sumX_2 += Math.Pow(NewX[i], 2);
        sumX_Y += NewX[i] * y[i];
    }
    double temp = (sumX_2 * size) - (sumX * sumX);
    double A = (sumX_Y * size - sumX * SumY) / temp;
    double B = (sumX_2 * SumY - sumX * sumX_Y) / temp;

    ANALIZ.Logarithmic_X_ln = A;
    ANALIZ.Logarithmic_C = B;
    ANALIZ.Logarithmic_M = A / 18;
    ANALIZ.Logarithmic_Equ = "y = " + Math.Round(A, 2) + "*ln(x) + " + Math.Round(B, 2);
    return ANALIZ;

}


public ANALIZ Exponential(ANALIZ ANALIZ, double[] x, double[] y)
{
    //exponential trendline  y = a * e ^(b * x).
    double total = 0;
    double sumX = 0;
    double SumY = 0;
    double sumX_2 = 0;
    double sumX_Y = 0;
    int size = 18;
    double min = 0;
    #region Update_Zeros

    for (int f = 0; f < 18; f++)
    {
        if (y[f] != 0)
        {
            min = y[f];
            break;
        }
    }
    for (int a = 0; a < 18; a++)
    {
        if (y[a] < min)
        {
            min = y[a];
        }
    }
    for (int a = 0; a < 18; a++)
    {
        if (y[a] == 0)
        {
            y[a] = min / 100;
        }
    }

    #endregion

    double[] NewY = new double[18];
    for (int a = 0; a < 18; a++)
    {
        NewY[a] = Math.Log(y[a]);
    }
    for (int i = 0; i < NewY.Length; i++)
    {
        sumX += x[i];
        SumY += NewY[i];
```

```
        sumX_2 += Math.Pow(x[i], 2);
        sumX_Y += x[i] * NewY[i];
    }
    double temp = (sumX_2 * size) - (sumX * sumX);
    double A = (sumX_Y * size - sumX * SumY) / temp;
    double B = Math.Exp((sumX_2 * SumY - sumX * sumX_Y) / temp);
    ANALIZ.Exponential_X_eUzeri = A;
    ANALIZ.Exponential_eKatsayi = B;
    ANALIZ.Exponential_M = A * B * Math.Exp(18 * A);
    ANALIZ.Exponential_Equ = "y = " + Math.Round(B, 2) + "*e^(" + Math.Round(A, 2)+"*x)";

    return ANALIZ;
}


public ANALIZ Power_Method(ANALIZ ANALIZ, double[] x, double[] y)
{
    // power trendline  y = a * x^b.
    double sumX = 0;
    double SumY = 0;
    double sumX_2 = 0;
    double sumX_Y = 0;
    int size = 18;
    double min = 0;

    #region Update_Zeros

    for (int f = 0; f < 18; f++)
    {
        if (y[f] != 0)
        {
            min = y[f];
            break;
        }
    }
    for (int a = 0; a < 18; a++)
    {
        if (y[a] != 0)
        {
            if (y[a] < min)
            {
                min = y[a];
            }
        }

    }
    for (int a = 0; a < 18; a++)
    {
        if (y[a] == 0)
        {
            y[a] = min / 100;
        }
    }

    #endregion

    double[] NewX = new double[size];
    double[] NewY = new double[size];

    for (int a = 0; a < size; a++)
    {
        NewY[a] = Math.Log(y[a]);
        NewX[a] = Math.Log(x[a]);
    }


    for (int i = 0; i < NewY.Length; i++)
    {
        sumX += NewX[i];
        SumY += NewY[i];
```

```
        sumX_2 += Math.Pow(NewX[i], 2);
        sumX_Y += NewX[i] * NewY[i];
    }

    double temp = (sumX_2 * size) - (sumX * sumX);
    double A = (sumX_Y * size - sumX * SumY) / temp;
    double B = Math.Exp((sumX_2 * SumY - sumX * sumX_Y) / temp);

    ANALIZ.Power_X_Ussu = A;
    ANALIZ.Power_X_Katsayi = B;
    ANALIZ.Power_M = A * B * Math.Pow(18, A - 1);
    ANALIZ.Power_Equ = "y = " + Math.Round(B, 2) + "* x^" + Math.Round(A, 2);
    return ANALIZ;
}



public ANALIZ TESTPower(ANALIZ ANALIZ)
{
    // power trendline  y = a * x^b.
    double sumX = 0;
    double SumY = 0;
    double sumX_2 = 0;
    double sumX_Y = 0;
    int size = 11;

    double[] tempX = new double[11] { 11, 15, 18, 23, 26, 31, 39, 44, 54, 64, 74 };
    double[] tempY = new double[11] { 0.00476, 0.0105, 0.0207, 0.0619, 0.337, 0.74, 1.7, 2.45, 3.5, 4.5, 5.09 };


    //double[] tempX = new double[4] { 1, 2, 3, 4 };
    //double[] tempY = new double[4] { 1, 4, 8, 15 };

    double[] NewX = new double[11];
    double[] NewY = new double[11];



    for (int a = 0; a < size; a++)
    {
        NewY[a] = Math.Log(tempY[a]);
        NewX[a] = Math.Log(tempX[a]);
    }


    for (int i = 0; i < NewY.Length; i++)
    {
        sumX += NewX[i];
        SumY += NewY[i];
        sumX_2 += Math.Pow(NewX[i], 2);
        sumX_Y += NewX[i] * NewY[i];
    }

    double temp = (sumX_2 * size) - (sumX * sumX);
    double A = (sumX_Y * size - sumX * SumY) / temp;
    double B = Math.Exp((sumX_2 * SumY - sumX * sumX_Y) / temp);

    ANALIZ.Power_X_Ussu = A;
    ANALIZ.Power_X_Katsayi = B;
    ANALIZ.Power_Equ = "y = " + Math.Round(B, 2) + "* x^" + Math.Round(A, 2);
    double[] Pow = new double[size];
    for (int i = 0; i < size; i++)
    {
        Pow[i] = Convert.ToDouble((ANALIZ.Power_X_Katsayi * Math.Pow(tempX[i], Convert.ToDouble(ANALIZ.Power_X_Ussu))));
    }
    errors er = new errors();

    er = FoundError(tempY, Pow); ANALIZ.Power_MaxError = er.max; ANALIZ.Power_AvgError = er.avg;
    ANALIZ.Power_RMS = er.Rms; ANALIZ.Power_R = er.R;
```

```
        return ANALIZ;
    }



    public ANALIZ Equations(ANALIZ analiz, double[] OrjData)
    {
        int size = 18;
        double[] tst_Reg = new double[10] { 40, 35, 30, 32, 19, 26, 24, 22, 18, 6 };
        double[] tst_Orj = new double[10] { 40.88, 34.3, 33.36, 29.6, 23.02, 23.02, 22.08, 18.38, 18.38, 10.8 };
        double[] LS_1 = new double[18];
        double[] LS_2 = new double[18];
        double[] LS_3 = new double[18];
        double[] Pow = new double[18];
        double[] Log = new double[18];
        double[] Exp = new double[18];
        for (int i = 0; i < size; i++)
        {
            LS_1[i] = Convert.ToDouble((i + 1) * analiz.LeastSquare_1_X + Convert.ToDouble(analiz.LeastSquare_1_C));
            LS_2[i] = Convert.ToDouble((i + 1) * (i * 1) * analiz.LeastSquare_2_X_sqr + (i + 1) * analiz.LeastSquare_2_X +
Convert.ToDouble(analiz.LeastSquare_2_C));
            LS_3[i] = Convert.ToDouble(Math.Pow((i + 1), 3) * analiz.LeastSquare_3_X_kup + (i + 1) * (i * 1) * analiz.LeastSquare_3_X_sqr + (i
+ 1) * analiz.LeastSquare_3_X + Convert.ToDouble(analiz.LeastSquare_3_C));
            Pow[i] = Convert.ToDouble((analiz.Power_X_Katsayi * Math.Pow((i + 1), Convert.ToDouble(analiz.Power_X_Ussu))));
            Log[i] = Convert.ToDouble(analiz.Logarithmic_X_ln * Math.Log((i + 1)) + analiz.Logarithmic_C);
            Exp[i] = Convert.ToDouble(analiz.Exponential_eKatsayi * Math.Pow(Math.E, ((i + 1) *
Convert.ToDouble(analiz.Exponential_X_eUzeri))));

        }
        errors er = new errors();
        er = FoundError(OrjData, Pow); analiz.Power_MaxError = er.max; analiz.Power_AvgError = er.avg;
        analiz.Power_RMS = er.Rms; analiz.Power_R = er.R; analiz.Power_R_error = er.R_error;
        er = FoundError(OrjData, LS_1); analiz.LeastSquare_1_MaxError = er.max; analiz.LeastSquare_1_AvgError = er.avg;
        analiz.LeastSquare_1_Rms = er.Rms; analiz.LeastSquare_1_R = er.R; analiz.LeastSquare_1_R_error = er.R_error;
        er = FoundError(OrjData, LS_2); analiz.LeastSquare_2_MaxError = er.max; analiz.LeastSquare_2_AvgError = er.avg;
        analiz.LeastSquare_2_Rms = er.Rms; analiz.LeastSquare_2_R = er.R; analiz.LeastSquare_2_R_error = er.R_error;
        er = FoundError(OrjData, LS_3); analiz.LeastSquare_3_MaxError = er.max; analiz.LeastSquare_3_AvgError = er.avg;
        analiz.LeastSquare_3_Rms = er.Rms; analiz.LeastSquare_3_R = er.R; analiz.LeastSquare_3_R_error = er.R_error;

        er = FoundError(OrjData, Log); analiz.Logarithmic_MaxError = er.max; analiz.Logarithmic_AvgError = er.avg;
        analiz.Logarithmic_Rms = er.Rms; analiz.Logarithmic_R = er.R; analiz.Logarithmic_R_error = er.R_error;

        er = FoundError(OrjData, Exp); analiz.Exponential_MaxError = er.max; analiz.Exponential_AvgError = er.avg;
        analiz.Exponential_Rms = er.Rms; analiz.Exponential_R = er.R; analiz.Exponential_R_error = er.R_error;

        return analiz;

    }


}
}
```

(Figure 4. Least Square Degree 1 Method Code) – All codes for the regression and relative error

Figure 9.1 Regression Graphs for Lark Blue
Figure 10.1 Forecasting for Lark Blue