Abstract

Food sales prediction is concerned with estimating future sales of companies in the food industry, such as supermarkets, groceries, restaurants, bakeries and patisseries.

Accurate short-term sales prediction allows companies to minimize stocked and expired products inside stores and at the same time avoid missing sales. This paper reviews existing machine learning approaches for food sales prediction. It discusses important design decisions

of a data analyst working on food sales prediction, such as the temporal granularity of sales

data, the input variables to use for predicting sales and the representation of the sales output

variable. In addition, it reviews machine learning algorithms that have been applied to food

sales prediction and appropriate measures for evaluating their accuracy. Finally, it discusses

the main challenges and opportunities for applied machine learning in the domain of food

sales prediction.

Keywords Food · Sales prediction · Demand forecasting · Machine learning · Regression · Timeseries forecasting

**Business problem:**

In today’s highly competitive and constantly changing business environment, the accurate and timely estimation of future sales, also known as sales prediction or sales forecasting, can offer critical knowledge to companies involved in the manufacturing, wholesale or retail of products. Short-term predictions mainly help in production planning and stock management, while long-term predictions can help in business development decision making. In our specific case we have a fast-food chain in Brazil with 400 stores with difficulty in forecasting its production.

Sales prediction is particularly important in the food industry due to the short shelf-life of many of the products, which leads to loss of income in both shortage and surplus situations. Ordering too many leads to waste of products, while ordering too few leads to opportunity loss.Therefore we have a situation where predicting correctly how much you have to produce of each item each day is important.

Moreover, food consumer demand is constantly fluctuating due to factors such as price, promotions, changing consumer preferences or weather changes.Sales prediction is typically done arbitrarily by managers. However, skilled managers are hard to find and they are not always available. In our specific case this forecast is done based on experience but is far from accurate. Average loss (too much or too little) spins around 10%.

Here it is important to put the management perspective: It is their view that today they relay too much in the people experience and would like to have some tool which would free them from this human dependence. In addition of that they believe that the level of error is high and could be reduced.

(e.g. theymay get sick or take a leave). Therefore,sales prediction should be supported by computer systems that can play the role of a skilled manager when she is not there and/or help her take the right decision by providing estimates of future sales. One way to build such a system would be to try and model the expert knowledge of skilled managers within a computer system. Alternatively, one can exploit the wealth of sales data and related information to automatically construct accurate sales prediction models via machine learning techniques. The latter is a much simpler process, it is not biased from the particularities of a specific sales manager and it is dynamic, meaning it can adapt to changes in the data. Furthermore, it has the potential to outweigh the prediction accuracy of a human expert, who typically is imperfect.

Therefore, from the management perspective a system capable of predicting the sales would be worth having even if it doesn´t perform better than the current process. Equal performance would be acceptable. In addition of that there was an understanding that the system would be able to improve its performance overtime as more and more historical data is added to its reference database. (Machine learning effect)

**Analytical problem**

The problem is how to build a model which effectively predicts the demand with a level of assertiveness equal or superior to the current one and improves overtime. Listening the thoughts of the people currently in charge of making this forecast we were informed that they believe that there is correlations between the following factors:

1. Day of the month (payment days usually have bigger demand)
2. Day of the week (Fridays, Saturdays, Sundays and holidays usually have a big demand)
3. Month (Holidays months usually have bigger sales – In Brazil Dez-Jan-jun-Jul)
4. Weather (temperature, rain and sun have an impact in what people eat)

This particular company sales its food through several channels: 1) directly from its stores, 2) through a web deliver service and 3) through a call-center. In our study we are not going to differentiate these channels just counting the total volume sold of each item each day.

Here it is worth mentioning that the insigns provided by the people in charge of the process today should be seen with a grain of salt given the fact that they know that a system like that would be built to replace them. Therefore all these assumptions must be checked against the hard data.

**Datasets**

To be able to identify the patterns we select the city of São Paulo which alone responds for almost 40% of the sales. This is a simplifying strategy given the fact that if the process works for this city we easily deploy it in the others.

**Getting the data**

Initially we managed to get the weather stations measurements in São Paulo for the whole year of 2018 and January of 2019. It is public information available at the website:

<http://www.inmet.gov.br/projetos/rede/pesquisa/>

Secondly we managed to get from the company the sales by type by day. In sequence we prepared this data crossing these two files unifying them by date. Our database format was:

**Evaluating the data**

Accuracy and completeness: A visual inspection showed that the sales data per day was correct, however the weather measurements had some problems of completeness. There were several days without the insolation, temperature and humidity recorded. In addition of that there are several days where the level of rain is zero, this is a problem because we don´t know it happens because it wasn´t recorded or because in fact didn´t rain in these days.

**Cleaning the data**

To deal with the inconsistences we defined three strategies:

1. Regards the missing values in precipitation we managed to see the average rain in each month (public information) and check it against the sum of each day/month in the database. Through this process we managed to identify that in fact the zero represented days without rain (the data was right).
2. The registers with temperature min, max or med and humidity equals zero were filled with the mean of these parameters (just two samples fall into this scenario).
3. In the case of the sun intensity we had a situation where 196 out of 396 samples were equal zero. Considering that there is no possibility that the sun didn´t appear for so many days it was assumed we had a problem with the data. We managed to check the average sun intensity per month in the city of São Paulo (public information) and fill the gaps manually. Subsequently we identify that the mean of the registers with measurement represent the average therefore it was possible to implement a code to correct it automatically.

The evaluation was important because allowed us to create a cleaning layer in the R code where we check these factors (2 and 3) and adjust it automatically – It is important because we assume that a new samples will be added to the training data as time goes by.

**Selecting the training data and the test data**

The idea is to use 80% of the measurements as our training data and 20% as our test data.

**The meaning of the columns (data dictionary)**

Data – Day of the weather measurements and sales

Precipitac – Volume of rain in millimetres per square meter during the day

Tempmax – Max temperature during the day

Tempmin – Minimum temperature during the day

Tempmed – Average temperature during the day

Umidade – Level of humidity in the air

Insolacao – Level of sun

Diasemana – day of the week abreviation

Diasem – weight of the day of the week considering the average volumes sold

Mes – Weight of the month considering the average sales volumes

Desserts – number of desserts sold in the day

Pizzas – Number of pizzas sold in the day

Beverage – Number of Beverage sold in the day

Cbmaker – Number of a special dish sold in the day

Combos – Number of combos sold in the day

Sfiha – Number of sfihas sold in the day

kit1 – Number of kits 01 sold in the day (Is a dish with a gift)

kit2 - Number of kits 02 sold in the day (Is a dish with a gift)

snack – Number of snacks sold in the day

pastas – Number of pastas sold in the day

dishes – Number of lunches sold in the day

promotion – Number of promotions sold in the day (This is episodic and may not be counted)

savory – Number of savory sold in the day

salads – Number of salads sold in the day

total – Total number of items sold in the day

The meaning of the numbers in the field diasem is:

1. Monday, Tuesday, Thursday
2. Friday
3. Saturday and Sunday
4. Holidays

The number represents the weight of the day regards sales. This weight reflects the view of the current planners regards sales and we need to check if the assumption holds. Analyzing the sales per day of the week we have the following graphic:



The current assumption of separating the days of week into four categories seems to be a bit wrong , the graphic shows three ranges as follows:

1. Monday – Tuesday – Wednesday-Thursday – Range 11% - 13% of the sales
2. Friday and Saturday – Range 17% and 19% of the sales
3. Sunday – Range 14% a 15% of the sales

The holidays match the volumes of FRI and SAT falling into the Range 2. The distribution by type of item goes as shown:



This analysis suggests that we change the days classification in the following manner:

1. Monday – Tuesday – Wednesday-Thursday – Range 11% - 13% of the sales
2. Sunday – Range 14% a 15% of the sales
3. Friday and Saturday – Range 17% and 19% of the sales

The current meaning of the numbers in the field mes is:

1. March, April, May, August, September, October, November
2. February and July
3. January ,June and December

The number represents the weight of the month regards sales, this weight reflects the view of the current planners regards sales and we need to check if the assumption holds. Analyzing the actual sales monthly by month classification we saw the following:



That suggests that there is a differentiation among the months as follows:

1. 1.500.000 items sold monthly (8,61% above the baseline)
2. 1.380.000 items sold monthly (Baseline)
3. 2.079.000 items sold (50% above the baseline)

That suggests the change 1 with 2 as a classification for the mes to keep it aligned with the sales volumes:

1. 1.380.000 items sold monthly (Baseline)
2. 1.500.000 items sold monthly (8,61% above the baseline)
3. 2.079.000 items sold (50% above the baseline)

Note that there is a mix between the climatic factors and what we could call seasonal effect however the inclusion of these factors were necessary given the fact that although it is assumed that the weather do influence the choice of the food the volumes are mostly influenced by the day of the week and the season.

Note that the objective is first to identify if the listed factors are in fact defining the demand and identify which ones are the most relevant.

Stablishing a correlation using the Kendall method between the demand for specific items. Here we are going to limit ourselves to the following items:

Desserts – number of desserts sold in the day

Pizzas – Number of pizzas sold in the day

Beverage – Number of Beverage sold in the day

Sfiha – Number of sfihas sold in the day

snack – Number of snacks sold in the day

pastas – Number of pastas sold in the day

dishes – Number of lunches sold in the day

savory – Number of savory sold in the day

salads – Number of salads sold in the day

The reason for that is the fact that the others are not regular items but some sort of promotion only made available for limited span of time. Items expurgated from the analysis:

Cbmaker – Number of a special dish sold in the day

Combos – Number of combos sold in the day

kit1 – Number of kits 01 sold in the day (includes a gift)

kit2 - Number of kits 02 sold in the day (Includes a gift)

promotion – Number of promotions sold in the day (This is episodic and may not be counted)

Seasonal parameters:

Diasem – weight of the day of the week considering the average volumes sold (1,2 or 3)

Mes – Weight of the month considering the average sales volumes (1,2 or 3)

Using the following code in R:

“setwd("/Curso-ML/Assignment-1/")

variables <- read.csv("data-06.csv")

summary(variables)

variables1 <- variables[17:32]

summary(variables1)

library(corrplot)

cor.test(variables1$diasem, variables1$pizzas)

cor.test(variables1$diasem, variables1$pizzas)

forcorrplot <-cor(variables1)

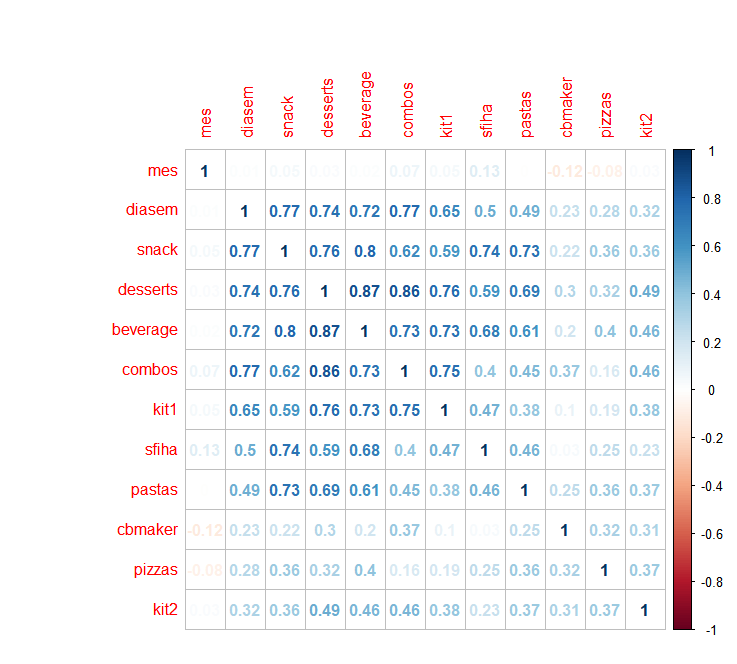
corrplot(forcorrplot)

corrplot(forcorrplot, upper="number", lower="color", order="hclust")

corrplot(forcorrplot, method="number",order="hclust" )

cormat(variables)”

We managed to see the correlation between the day of the week and month and the consumption of the several items:



As we can see the consumption of some items vary more them others as the days of the week change.

Now we have to see if there is correlation between the weather factors and the consumption by item:

Precipitac – Volume of rain in millimetres per square meter during the day

Tempmax – Max temperature during the day

Tempmin – Minimum temperature during the day

Tempmed – Average temperature during the day

Umidade – Level of humidity in the air

Insolacao – Level of sun

Remembering that doing the correlation we are going to identify the variable R which can varies from +1 to -1 indicating that a relationship exists between the two variables from absolute direct correlation (+1), no correlation (0) to inverse correlation (-1). Using R studio and the command “corrplot” we managed to see the following:

“setwd("/Curso-ML/Assignment-1/")

variables <- read.csv("data-06.csv")

summary(variables)

variables$diasemana <- NULL

variables1 <- variables[10:31]

summary(variables1)

library(corrplot)

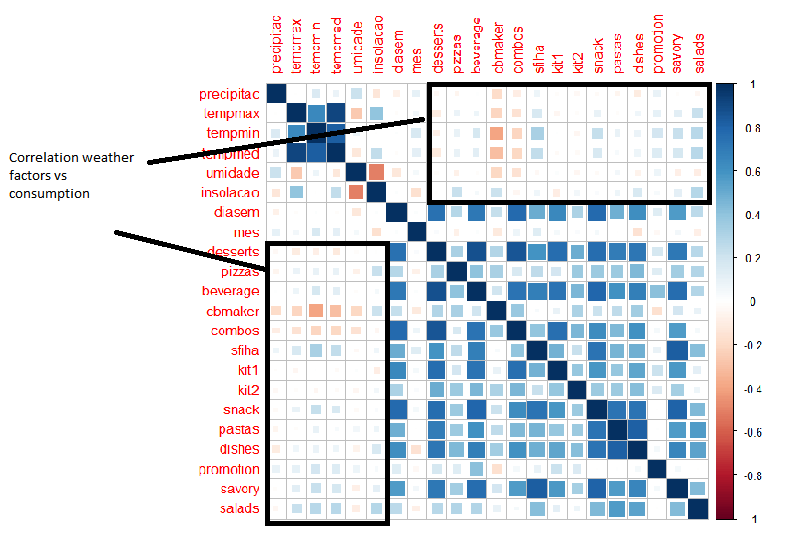
cor.test(variables1$diasem, variables1$pizzas)

cor.test(variables1$diasem, variables1$pizzas)

forcorrplot <-cor(variables1)

corrplot(forcorrplot)

corrplot(forcorrplot, method = "square")”



As we can see in the graphics there are some correlation between weather parameters and the consumption. However the weather conditions are not so determining as the day of the week and the month of the year.

Therefore building a model to guess the demand will imply the definition of the average demand of the day of the week in a given month and in sequence addjust this demand by the weather conditions.

To archive that we grouped the demand by the combination of type of day vs type of month with nine categories:



This analysis grouped the data as follows:



This would be the demand if only these two factors were influencing the actual sales, therefore the challenge is to identify how the weather conditions make the average demand deviate (up and down) from these values.

**Identifying the demand**

Once we sliced the dataset by type of week and type of month we have to identify which one of the six weather parameters is more influential in adjusting the demand for each item (nine types).

To be able to do that we have to identify the correlation factor R² between each weather parameter and the volume sold. The factor with bigger R² is the ones to be used as “predictor” of the demand. We are going to use linear regression to predict the demand, identifying the parameters A and B in the formula Ax + B = Y where x is the chosen weather parameter and Y de expected demand for the product. This calculation will be done for each one of the nine types of products.

The results

Errors

Considerations

Here we have to analyze each item separately:

Desserts:

Analyzing the graphic x we can see that the consumption of desserts is inversely proportional to the temperature. That means the colder the weather more desserts people consume.

That said we have to identify the curve associating the consumption of desserts with temperature for each one of the nine combinations of (day of week – month):

Day of week 1 – month 1

Sfihas

Coefficients:

Coefficients:

Estimate Std. Error t value Pr(>|t|)

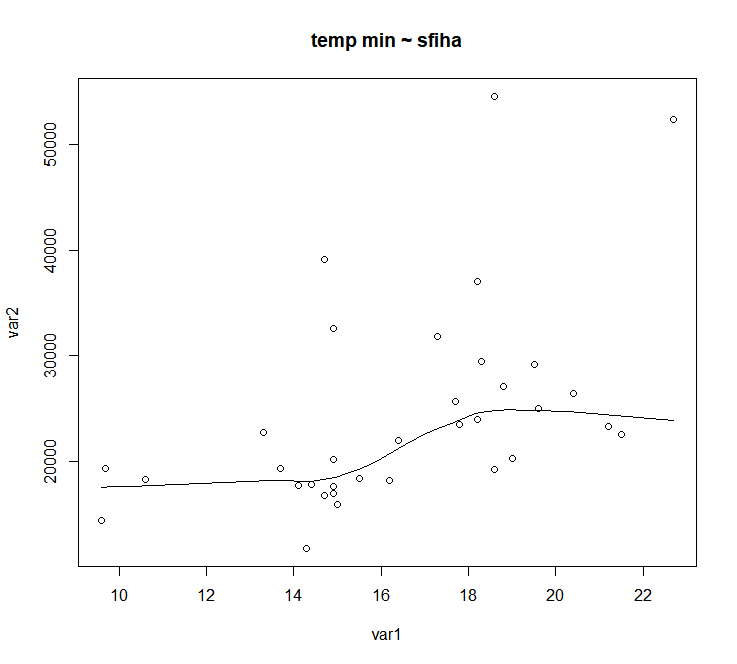
(Intercept) -664.1 7771.5 -0.085 0.93244

var1 1524.0 464.2 3.283 0.00249 \*\*

---

Here we have an linear equation = 1524 x temp med -664.1 = Volume of sfihas sold

That generates a scenario where when we apply the formula and compare the results with the real sales we have that 7 out of 34 samples show results with an error below 10% (Better than the current process). Therefore we have a model that got it right in 20% of the guesses (at least as right as today).



Day of week 2 – month 1

Sfihas

Coefficients:

Estimate Std. Error t value Pr(>|t|)

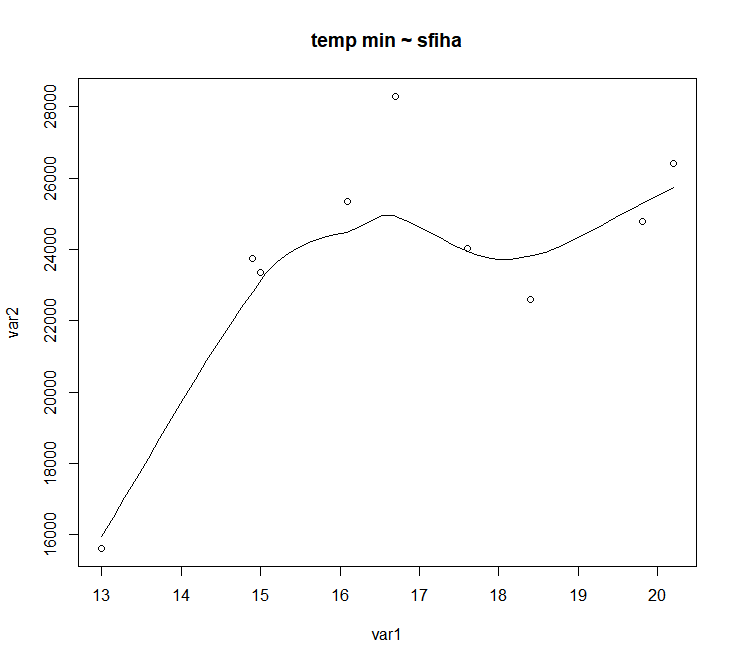
(Intercept) 8780.8 7531.3 1.166 0.2818

var1 890.9 442.9 2.012 0.0842 .

---

Here we have an linear equation = 890.9 x temp min + 8780,8 = Volume of sfihas sold

That generates a scenario where when we apply the formula and compare the results with the real sales we have that 6 out of 9 samples show results with an error below 10% (Better than the current process). Therefore we have a model that got it right in 67% of the guesses (at least as right as today).



Day of week 3 – month 1

Sfihas

Coefficients:

Estimate Std. Error t value Pr(>|t|)

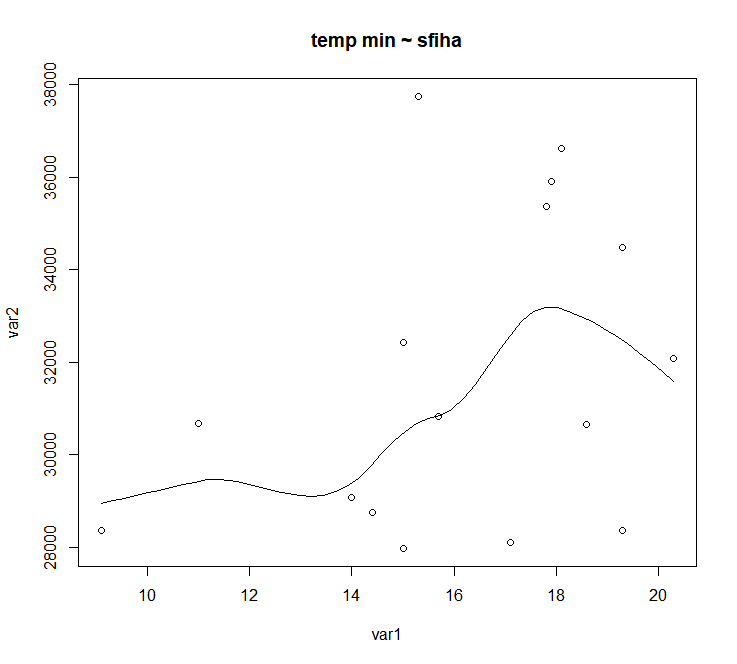
(Intercept) 25417.8 4455.2 5.705 5.44e-05 \*\*\*

var1 390.8 271.8 1.438 0.173

---

Here we have an linear equation = 390.8 x temp min + 25417.8 = Volume of sfihas sold

That generates a scenario where when we apply the formula and compare the results with the real sales we have that 10 out of 16 samples show results with an error below 10% (Better than the current process). Therefore we have a model that got it right in 62% of the guesses (at least as right as today).



Week 1 – month 2

Sfihas

Coefficients:

Estimate Std. Error t value Pr(>|t|)

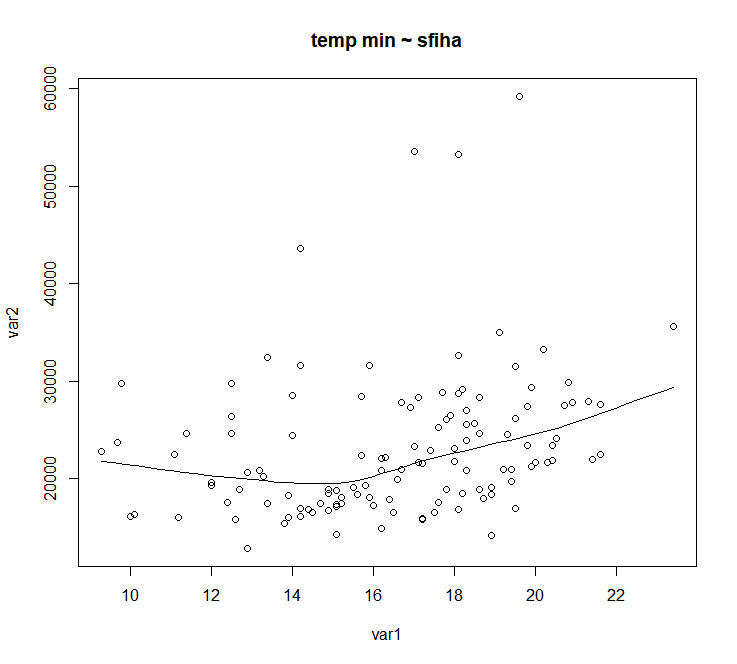
(Intercept) 12182.0 3665.8 3.323 0.00118 \*\*

var1 664.0 218.2 3.043 0.00287 \*\*

---

Here we have an linear equation = 664 x temp min + 12182 = Volume of sfihas sold

That generates a scenario where when we apply the formula and compare the results with the real sales we have that 30 out of 123 samples show results with an error below 10% (Better than the current process). Therefore we have a model that got it right in 25% of the guesses (at least as right as today).



Week 2 – month 2

Sfihas

Coefficients:

Estimate Std. Error t value Pr(>|t|)

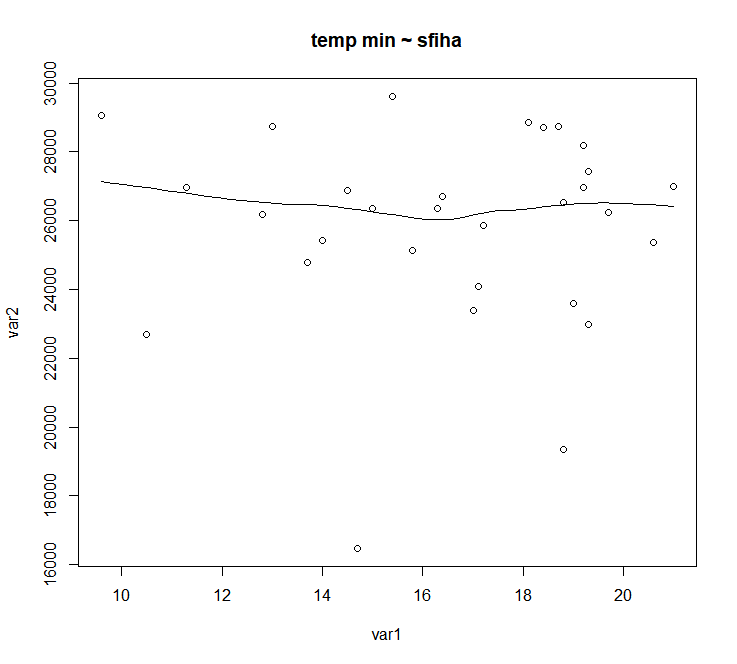
(Intercept) 25865.055 2978.734 8.683 1.97e-09 \*\*\*

var1 -2.627 177.843 -0.015 0.988

---

Here we have an linear equation = -2.627 x temp min + 25865 = Volume of sfihas sold

That generates a scenario where when we apply the formula and compare the results with the real sales we have that 19 out of 30 samples show results with an error below 10% (Better than the current process). Therefore we have a model that got it right in 55% of the guesses (at least as right as today).



Week 3 – month 2

Sfihas

Coefficients:

Estimate Std. Error t value Pr(>|t|)

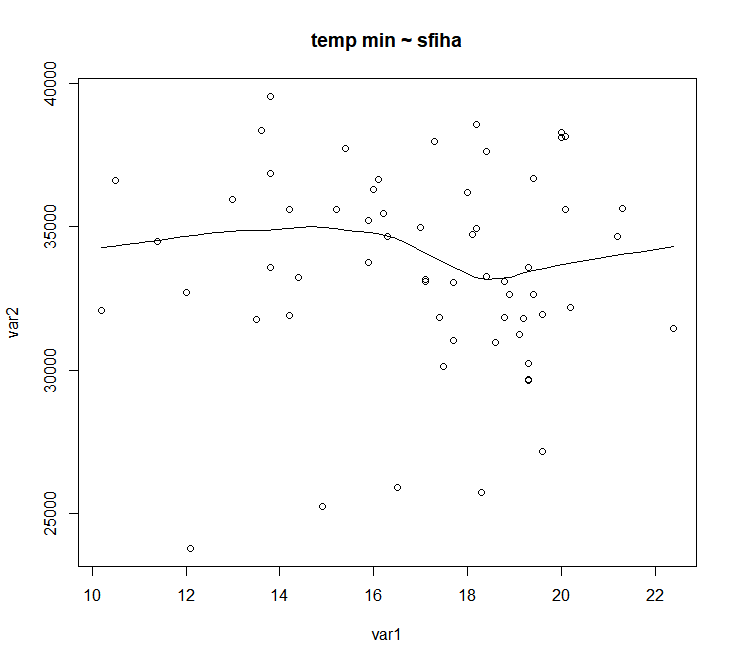
(Intercept) 34328.14 2776.22 12.365 <2e-16 \*\*\*

var1 -45.79 160.84 -0.285 0.777

---

Here we have an linear equation = -45.79 x temp min + 34328 = Volume of sfihas sold

That generates a scenario where when we apply the formula and compare the results with the real sales we have that 43 out of 61 samples show results with an error below 10% (Better than the current process). Therefore we have a model that got it right in 70% of the guesses (at least as right as today).



Week 1 – month 3

Sfihas

Coefficients:

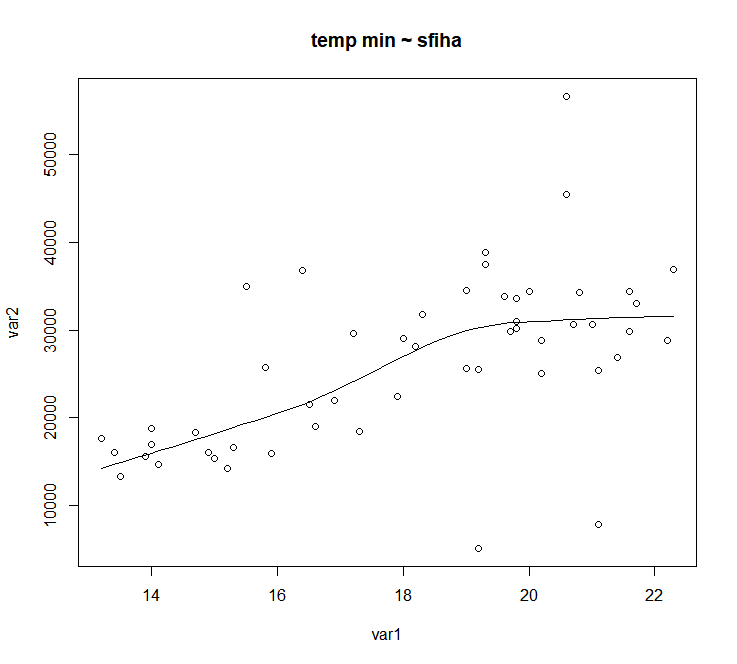
Estimate Std. Error t value Pr(>|t|)

(Intercept) -11381.9 7489.5 -1.520 0.135

var1 2075.5 408.7 5.078 5.67e-06 \*\*\*

Here we have an linear equation = 2075 x temp min – 11.381,9 = Volume of sfihas sold

That generates a scenario where when we apply the formula and compare the results with the real sales we have that 18 out of 52 samples show results with an error below 10% (Better than the current process). Therefore we have a model that got it right in 34% of the guesses (at least as right as today).



“scatter.smooth(x=var1, y=var2, main="temp min ~ sfiha") # scatterplot”

Week 2 – month 3

Sfihas

Coefficients:

Estimate Std. Error t value Pr(>|t|)

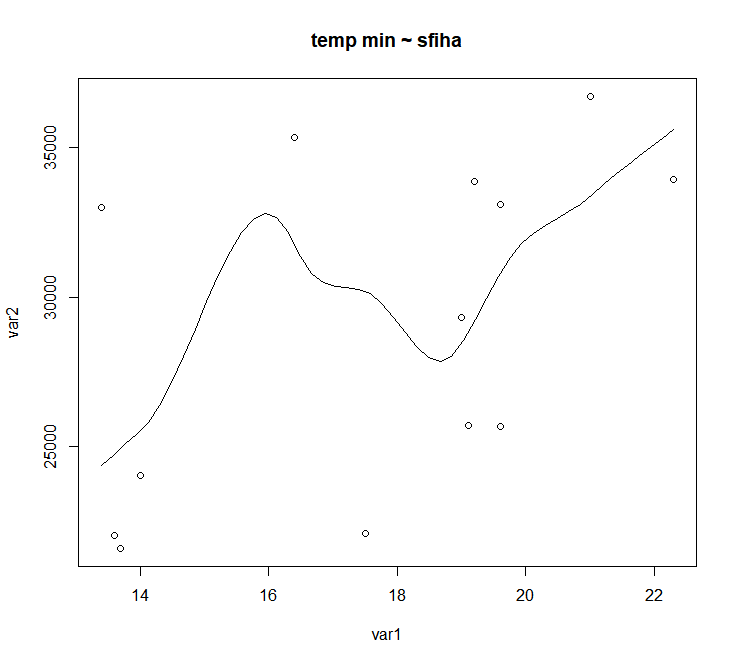
(Intercept) 12413.3 8488.7 1.462 0.1716

var1 941.6 476.6 1.976 0.0738 .

---

Here we have an linear equation = 941.6 x temp min +12413.3 = Volume of sfihas sold

That generates a scenario where when we apply the formula and compare the results with the real sales we have that 5 out of 13 samples show results with an error below 10% (Better than the current process). Therefore we have a model that got it right in 38% of the guesses (at least as right as today).



Week 3 – month 3

Sfihas

Coefficients:

Estimate Std. Error t value Pr(>|t|)

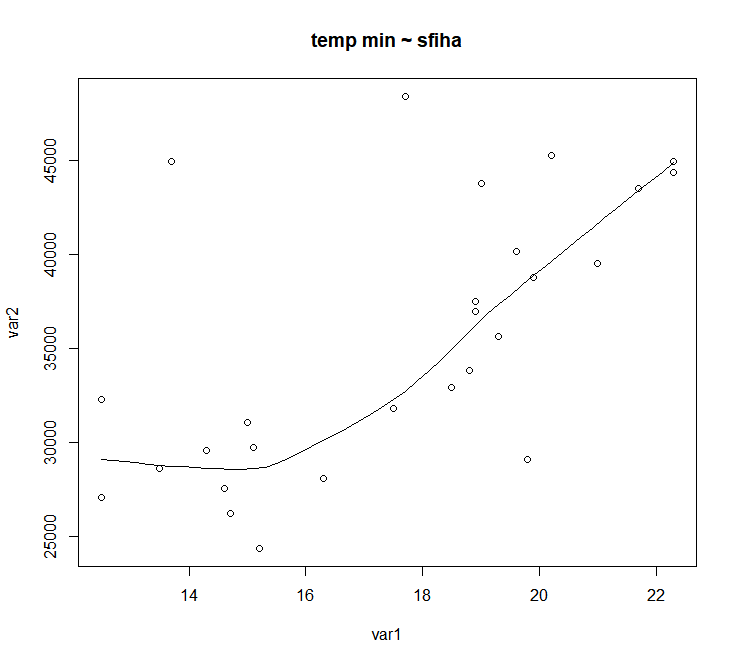
(Intercept) 8939.2 6418.1 1.393 0.175944

var1 1511.7 361.4 4.183 0.000309 \*\*\*

---

Here we have an linear equation = 1511.7 x temp min + 8939.2 = Volume of sfihas sold

That generates a scenario where when we apply the formula and compare the results with the real sales we have that 5 out of 27 samples show results with an error below 10% (Better than the current process). Therefore we have a model that got it right in 18% of the guesses (at least as right as today).





**Annex 1**

Using this code in R we manage to get the average volumes of sales per type of product per range:

“setwd("/Curso-ML/Assignment-1/")

data <- read.csv("data-06.csv")

data1 <- read.csv("dia-sem-me.csv")

summary(data)

summary(data1)

print(data1)

len3=nrow(data1)

print(len3)

print(data1[1,4])

print(data[1,17])

library(stringr)

len2=nrow(data)

print(len2)

contador2 = 1

while (contador2<=len2){

print(contador2)

print(contador2)

print(contador2,17)

len3=nrow(data1)

print(len3)

contador1 = 1

while (contador1<=len3){

print(contador1)

print(data1[contador1,2])

print(data[contador2,17])

if (data1[contador1,2] == data[contador2,17] && data1[contador1,3] == data[contador2,18]){

print(data1[contador1,1])

data1[contador1,4]<-data1[contador1,4]+data[contador2,19]

data1[contador1,5]<-data1[contador1,5]+data[contador2,20]

data1[contador1,6]<-data1[contador1,6]+data[contador2,21]

data1[contador1,7]<-data1[contador1,7]+data[contador2,22]

data1[contador1,8]<-data1[contador1,8]+data[contador2,23]

data1[contador1,9]<-data1[contador1,9]+data[contador2,24]

data1[contador1,10]<-data1[contador1,10]+data[contador2,25]

data1[contador1,11]<-data1[contador1,11]+data[contador2,26]

data1[contador1,12]<-data1[contador1,12]+data[contador2,27]

data1[contador1,13]<-data1[contador1,13]+data[contador2,28]

data1[contador1,14]<-data1[contador1,14]+data[contador2,29]

data1[contador1,15]<-data1[contador1,15]+data[contador2,30]

data1[contador1,16]<-data1[contador1,16]+data[contador2,31]

data1[contador1,17]<-data1[contador1,17]+data[contador2,32]

data1[contador1,19]<-data1[contador1,19]+1

}

contador1 = contador1 + 1

}

contador2 = contador2 + 1

}

len4 = nrow(data1)

contador3 = 1

while (contador3<=len4){

print(contador3)

print(data1[contador3,2])

print(data1[contador3,1])

data1[contador3,4]<-data1[contador3,4]/data1[contador3,19]

data1[contador3,5]<-data1[contador3,5]/data1[contador3,19]

data1[contador3,6]<-data1[contador3,6]/data1[contador3,19]

data1[contador3,7]<-data1[contador3,7]/data1[contador3,19]

data1[contador3,8]<-data1[contador3,8]/data1[contador3,19]

data1[contador3,9]<-data1[contador3,9]/data1[contador3,19]

data1[contador3,10]<-data1[contador3,10]/data1[contador3,19]

data1[contador3,11]<-data1[contador3,11]/data1[contador3,19]

data1[contador3,12]<-data1[contador3,12]/data1[contador3,19]

data1[contador3,13]<-data1[contador3,13]/data1[contador3,19]

data1[contador3,14]<-data1[contador3,14]/data1[contador3,19]

data1[contador3,15]<-data1[contador3,15]/data1[contador3,19]

data1[contador3,16]<-data1[contador3,16]/data1[contador3,19]

data1[contador3,17]<-data1[contador3,17]/data1[contador3,19]

contador3 = contador3 + 1

}

print(data1)

##write.csv(data1,'/Curso-ML/Assignment-1/dia-sem-me.csv')

class(data$diasem)

print(nrow(data))”