Onion Analytics

IEOR 4650 - Business Analytics

Daniel Macgregor Oettler - dm3229@columbia.edu
Hassan Ahmed Mortagy - ham2149@columbia.edu
Kavana Rudresh - kr2682@columbia.edu
Ramon Rodriganez Riesco - rr3088@columbia.edu
Ramsey Hamadeh - rh2787@columbia.edu

Onion Analytics

Table of Contents

Acknowledgment

We would like to express our gratitude to **Mr. Ernesto Grossmann,** a Mexican onion producer for talking to us about the onion market in Mexico. His insights were of immense help to us over the course of this project.

We also thank **Prof. Adam Elmachtoub** for giving us an opportunity to do this project.

1. Introduction

a) Background

Small-scale farmers throughout the world are exposed to variability in the prices they get for their crops. Most farmers typically sell their produce at the spot price in the few weeks after they harvest their crop. This translates into two problems for many farmers: uncertainty of income and selling at a suboptimal price.

b) Problem Statement

What added value can small-scale farmers get by selectively picking the time of sale of their crops? Specifically, would the extra-income generated from selling at optimal price outweigh the cost of inventorying these crops?

c) Value

Our project aims to tackle the uncertainty in crop prices, and help producers achieve higher profit. We aim to build a model which considers various features such as time, weather and demand, to make a prediction of the spot price for a given crop for the next few days or weeks in order to make an optimal decision. This model could be used by small producers who rely on selling their produce at the spot price just after harvesting. While one could argue that the buyer has more power than a seller, the idea here is that the producer would not put his crops on the market until he or she is sure of getting a higher profit.

Developing a model to predict prices and solving this problem with high accuracy would enable farmers to decide when to sell their produce in the short term (local maximum). This point will need to be within the period during which the crop remains fresh. For various types of crops, the farmers could invest in some sort of warehouse (e.g.: a silo), which would allow farmers to hold the produce for a period before selling it. In most rural areas an d particularly in Mexico, the cost of storage is extremely low. If the potential increase in revenue is enough to pay the investment for storage infrastructure, there would be value for the farmers in holding their produce for a certain period and selling when the price is predicted to be higher.

2. Analysis

a) Approach

For data availability reasons and because of the durability of storage of different crops, we choose to analyze onions in Mexico. The project is broken down into three sections. We first web scrap daily crop prices and weather data from 2008 to 2016 using trusted websites. Second, we aggregate the datasets by matching dates and geo-locations and we apply functions to create the price predictors. Finally, we apply different models and select the one with best predictive accuracy.

In terms of models, we begin by trying different approaches seen in class, including Linear Regression, Trees and Lasso Regression. However, these models do not fare well. We therefore decided to implement Neural Networks and Support Vector Machines algorithms which tend to perform well in time series analysis. The performance of all these models is then evaluated and the prediction accuracies are compared.

b) Data Cleaning and Collection

Onion Prices: For the prices of the onions over the past few years, we used Mexico's National System of Information and Integration of Markets website (i.e. http://www.economia-sniim.gob.mx/nuevo/). We used Python to web scrape the data, and created a dataset with the following features: Date, Origin, Destination, MinPrice, MaxPrice, MeanPrice. The code used for Web scraping is included in **Appendix 1**. For all our analysis we kept only one Destination to maintain simplicity in our model, and chose "Aguascalientes" as this is one of the major markets.

Climate Data: We got the necessary climate data from the website of National Centers for Environmental Information (NCEI) (i.e. https://www.ncdc.noaa.gov/cdo-web/search). Our approach was to find the stations corresponding to the Onion producing areas of each state in Mexico, and request for the Climate Data captured at these particular stations. The columns of the dataset were Station Name, Date, Avg Precipitation, Min Temp, Avg Temp, Max Temp and State.

Features: The features were chosen based on interviews to real farmers in Mexico that enabled us to properly understand the farming of onions. We have 3 dummy variables for the season, 7 variables for the weather: Avg Temp, Max Temp, Min Temp, Cumulative Precipitation, Max Precipitation (since one day for highly excess rainfall can destroy crops), Days without Precipitation (to account for drought conditions) and if there was any humid period in the past month (since a humid period of more than 10 days can also affect crops by provoking diseases) and a 11 others for the price: mean, max and min 1 and 2 years ago and 28 days ago, the changes between 35 and 28 days and between 56 and 28 days ago. Overall, 22 features were chosen.

Timing of Features: We decided to explore two methods to make our dataset ready for modelling, namely Method 0 and Method 1. Method0 predicts the price of the next day using the last 4 days as predictors, while Method1 predicts the next 28 days (using predictors from 28 days ago and more). For our tool to be useful, we have to predict a few days ahead to be able to choose the best day of selling. We calculated that optimal price spread was 1.4 Pesos / kg if we were able to predict the next 28 days, and it was reduced to 0.75 Pesos if we were just able to predict 14 days ahead. When applied Method0 to do just 2 periods multi-step, the MSE increases to about 17 as shown in Figure 1. Therefore, it will never yield good results if we target 14 or 28 days ahead. This low accuracy and increased MSE was our motivation to use Method1 over Method0.

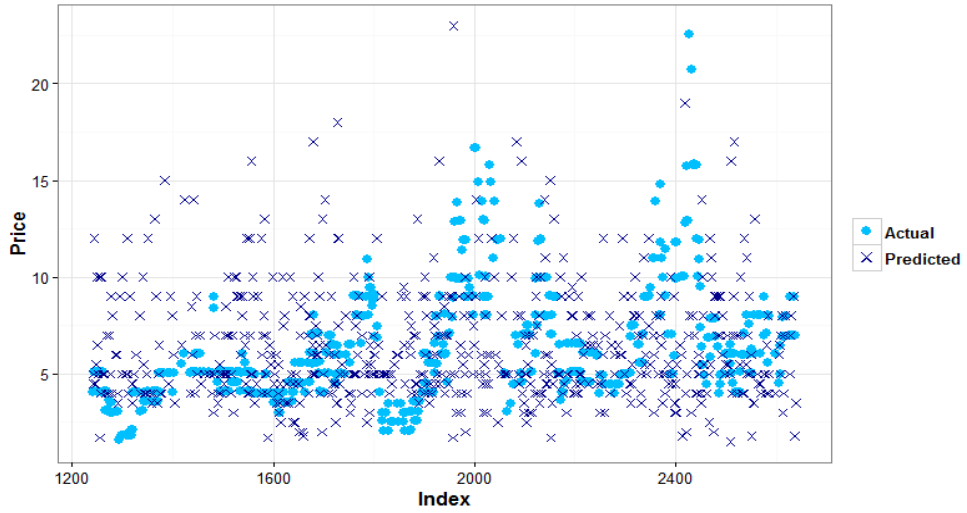

Figure 1. 2 periods, multi-step prediction with SVM (Method 0)

Origin selection: We also wanted our model to also predict if a particular origin caters to the market on a particular day, since in real life one does not know that in advance. For each date, our approach was to check the proportion of days in the prior years each origin was active and we forecasted a 1 if this proportion was over 0.5.

c) Modeling

We conducted our analysis using Lasso, Support Vector Machines and Neural networks.

I) Lasso

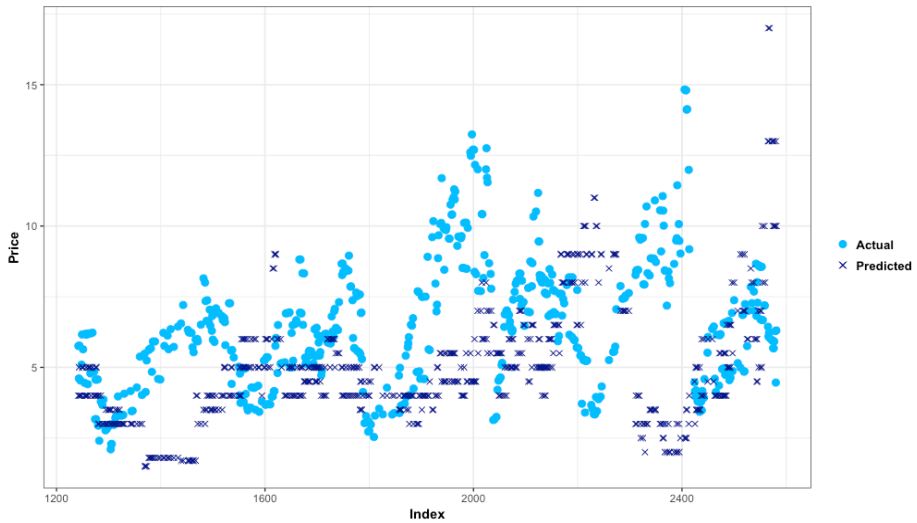

Figure 2. Lasso Prediction (out of sample)

After cross validation, Lambda_optimal = 0.01. The Coefficients of Lasso for Model1 are attached in **Appendix2**. **MSE_Lasso = 3.98 (out of sample)**

II) Neural Networks

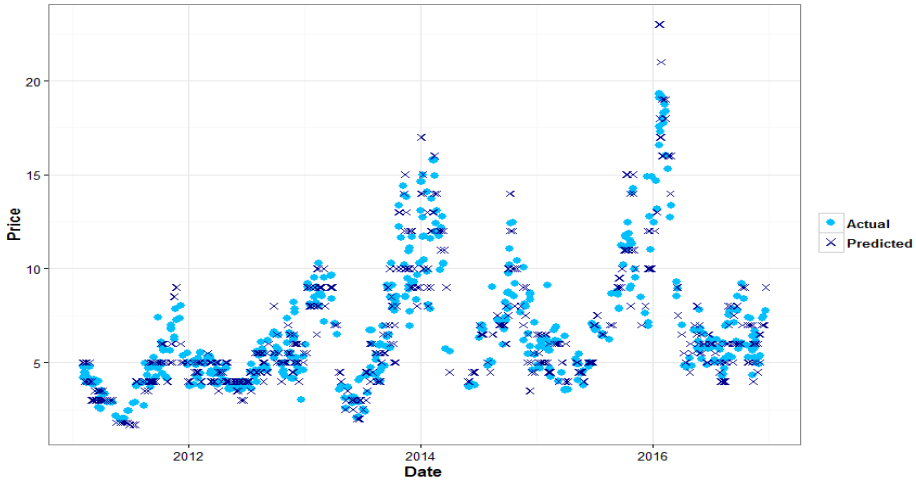

Figure 3. Neural Net prediction (out of sample)

After cross validation, we found that the optimal parameters for the neural network are: Hidden layers = 2, Nodes_layer_1 = 13, Nodes_layer_2 = 7. The Neural Net is shown in **Appendix 3.** **MSE_NN = 1.2 (out of sample)**

III) Support Vector Machines (SVM)

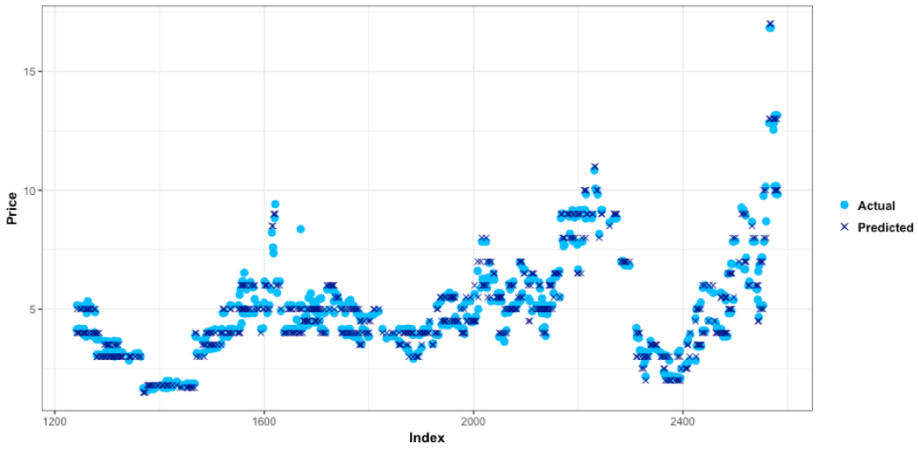

Figure 4. SVM prediction (out of sample)

After cross validation, we found that the optimal parameters are: Best kernel: Radial, Gamma for kernel = 0.2, Cost = 10, Epsilon = 0.05. **MSE_SVM = 0.395 (out of sample).**

Table 1 summarizes these analyses:

	Method 0 (next day)	**Method 1 (28 days ahead)**
Lasso	0.429	3.98
Neural Networks	0.49	1.20

Support Vector Machine	0.43	0.395

Table 1: MSE comparisons on out of sample sets

The predictive performance on the out of sample set shows that the SVM model yields the best prediction accuracy with an MSE of 0.395 on method 1. For that reason, we use SVM for our final model which we explain in further details below.

d) Final Methodology:

Now that we have determined that SVM produces the best prediction accuracy for the chosen methodology, we create a final model that is more representative of reality. To that end, we train the data on dates from Jan 01, 2011 to Jan 01, 2014. We then test the model on the 28 days following Jan 01, 2014 and store the predictions. After that, we incorporate these 28 days, and retrain the model on everything up until Jan 28, 2014. We then predict the 28 days between Jan 29, 2014 to Feb 26, 2014 and store those predictions. We keep on iterating until we use the whole dataset. The overall MSE over the period Jan 01 2015 and Dec 31 2016 is: **MSE = 7.75.**

As expected, the MSE is a lot higher in this real setting. When we sampled at random data in between 2011 and 2016, the model was much better because it had "learnt" from all prices ranges, whereas now we only train on prices that have occurred up to date. For instance, the price peak in January 2016 is almost impossible to forecast with data available at the end of 2015. The MSE increases at the expense of a model that can be used in real life.

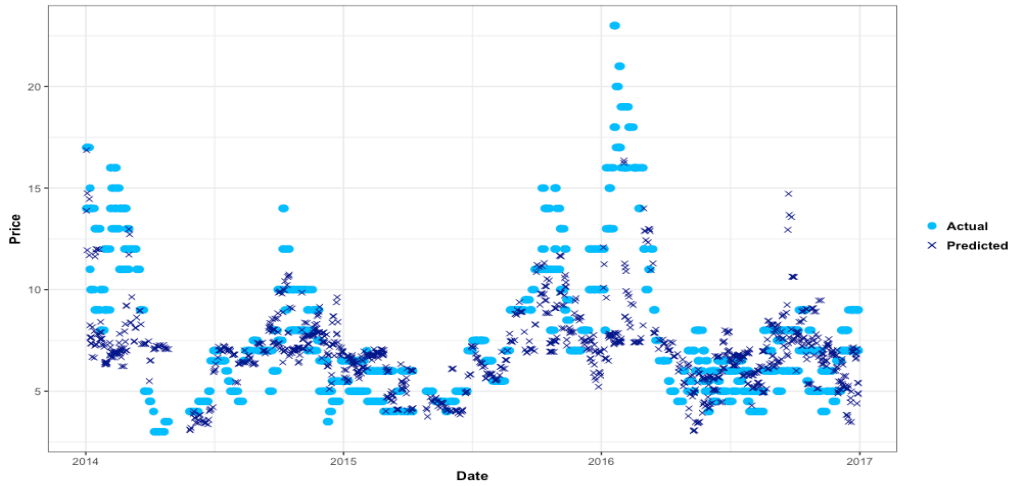

Figure 5. SVM final model prediction, 2014 – 2016

3. Conclusion

a) Value creation using SVM

Our analysis has shown that Support Vector Machines (SVM) yields the best performance for this time-series analysis. To quantify the business value of our results, we estimate the net additional

profit that a medium sized farmer would make. To do this, we use the forecast to know when the price is going to reach its maximum in the next 28 days. Then, we let the new price of sale ("optimal seller") be equal to the actual price of the crop on the day we forecasted would have the highest price. Finally, we compute the price difference between the "optimal seller" and a farmer that sells his produce randomly in the next 28 days at the market. This difference can be positive (good) or negative (bad). Using our final methodology, we find that 0.30 MXN/kg can be generated per year if the farmer follows our model's predictions. This is a great number given that the optimal price spread (the extra amount one could get if predicted the optimal day to sell among the next 28 with 100% accuracy) is $1.43. This means we are capturing 21% of potential value.

Consider a farm that harvests 35 tons of onions per hectare, with an average surface area of 100 hectares. Given our model, this particular farm can bring in an additional revenue of **USD 31,000** per year! The assumption that is made here is that the silo and holding costs are negligible (reasonable in rural Mexico). In terms of the market opportunity of this idea to develop a business, foresee a **Total Addressable Market of $1.4 million** for the market of onions in Mexico. We use the yearly 1.3 million tones production and the optimal price spread and the hypothesis selling this tool to small farmers accounting for 5% of the overall production will not affect market prices as they are price takers. If we charge farmers a 30% of the value generated, we could reach sales of:

$$\boldsymbol{TAM} = \%5 \cdot \mathit{Total\ Market} \cdot \mathit{Optimal\ Price\ Spread} \cdot 30\% = \$\mathbf{1.4\ million}$$

The potential value of this project, using our price spread (0.3 MXN/kg) instead of the optimal, is:

$$\boldsymbol{Onion\ Analytics\ Value} = \%5 \cdot \mathit{Total\ Market} \cdot \mathit{OA\ Price\ Spread} \cdot 30\% = \$\mathbf{310k}$$

b) Limitations

- We are assuming that the prices listed on the market site are accurate. Based on our research, we found that this is the surveyed price from the market, and we believe that the government would not have a reason to list the wrong prices.
- There could be sudden epidemics, long periods of drought, macroeconomic changes, and other factors that might influence the prices.
- If all producers started doing this, then the model would no longer be profitable since this would affect macro demand. That is why we capper our value to 5%, a hypothesis that should be further tested
- We have done our analysis on only one destination market, and the same model might not work for other markets.
- Our origin prediction to create the test dataset is just an average of the usage on the previous years. This is of course not optimal, and could be further improved.

Appendix:

Appendix 1: Web scrapping code

```
import requests
from bs4 import BeautifulSoup
import csv

#read website with all rows (almost 5000)
response                              =                              requests.get("http://www.economia-
sniim.gob.mx/NUEVO/Consultas/MercadosNacionales/PreciosDeMercado/Agricolas/ResultadosConsultaFech
aFrutasYHortalizas.aspx?fechaInicio=01/01/2012&fechaFinal=31/12/2016&ProductoId=183&OrigenId=-
1&Origen=Todos&DestinoId=-1&Destino=Todos&PreciosPorId=2&RegistrosPorPagina=50000")

#parse with beautiful soup
soup = BeautifulSoup(response.content, 'html.parser')

#include in a list (first row to take:22)
#can also be done i a dictionary or dataframe
l = []
count = 0
for tr in soup.findAll('tr'):
    count += 1
    if (count > 21):
        for td in tr.findAll('td'):
            l.append(td.get_text())

#transform to a proper list of lists
dataset = []
for i in range(1,len(l),8):
    k = [l[i],l[i+2],l[i+3],l[i+4],l[i+5],l[i+6]]
    dataset.append(k)

#titles
titles = ['Date','Origin','Destiny','MinPrice','MaxPrice','MeanPrice']

#write to a csv
with open('DataBAOnions.csv','w') as fp:
    datawriter = csv.writer(fp, delimiter = ',')
    datawriter.writerow(titles)
    for i in range(1,len(dataset)):
        datawriter.writerow(dataset[i])

#import pandas
#my_df = pd.DataFrame(columns=['Date','Origin','Destiny','MinPrice','MeanPrice','MaxPrice'])
#for i in range(1,len(l),8):
#    my_df.loc[i]=[l[i],l[i+2],l[i+3],l[i+4],l[i+5],l[i+6]]
```

Appendix 2: Lasso Model coefficients

```
29 x 1 sparse Matrix of class "dgCMatrix"
                             1
(Intercept)      -1.310119960
TAVG              0.146349945
TMAX              .
TMIN              .
PRCPCUM           0.003176396
PRCPMAX           0.007567378
MAXDAYSNOPRCP     .
MAXDAYSHUMID10    .
WINTER            1.167178561
SPRING           -0.547416971
SUMMER            .
PAVG1Y            .
PMAX1Y            .
PMIN1Y            .
PAVG2Y            .
PMAX2Y            .
PMIN2Y            0.050622754
PAVG1M            0.491113940
PMAX1M            0.078778039
PMIN1M            .
PCH1M1W          -0.045545985
```

Appendix 3: Neural network results