Supervised Learning on ”IPMA” and INE data

1st Beatriz Gonçalves 115367
*DETI**
University of Aveiro
Data Science Foundations
Prof. Sérgio Matos

2nd Tiago Nazário 89980
*DETI**
University of Aveiro
Data Science Foundations
Prof. Sérgio Matos

*DETI: Department of Electronics, Telecommunications and Informatics

***Abstract*—We combined agricultural data from the INE website [9] with meteorological data from the ”IPMA” website [?]. The goal was to test our ability to understand what is the most relevant information we can extract from these datasets by first doing data cleaning in order to organize the data so that we can extract important information, and then doing a visualization of what we think is most important to extract information from this data for a better understanding and analysis; finally, we used machine learning models in order to predict the production of a series of product-region combinations, and also predict the precipitation in the North of the year 2018 (exclusively with ”IPMA”” data). For this last part, we solved a supervised regression problem, comparing four algorithms: linear regression (Lr), ridge regression (Rr), lasso regression (Lsr), and decision tree regression (Dtr).**

***Index Terms*—data cleaning, visualization, machine learning, linear regression, ridge regression, lasso regression, decision tree regression**

I. Introduction

Portugal produces a wide variety of products. It is the largest world producer of both cork and carob, as well as the third largest exporter of chestnut and the third largest European producer of pulp. [4] Portugal is among the top ten largest olive oil producers in the world and is the fourth biggest exporter. [2] The country is also one of the world's largest exporters of wine, being reputed for its fine wines. [1] The land area of slightly more than 9.2 million hectares was classified as follows (in thousands of hectares): 2,755 arable land and permanent crops (including 710 in permanent crops), 530 permanent pasture, 3,640 forest and woodland, and 2,270 other lands. [4]

We used five different datasets and tried to work with them in a way that we could extract relevant information. The datasets used were:

- Productivity of the main agricultural crops (kg/ ha) by Geographic Location (NUTS - 2013) and Species; Annual; [5]
- Area of main agricultural crops (ha) by Geographic Location (NUTS - 2013) and Species; Annual; [5]
- Production of main agricultural crops (t) by Geographic Location (NUTS - 2013) and Species; Annual; [5]
- Electricity consumption (kWh) by Geographic localization (NUTS - 2013) and Type of consumption; Annual; [3]
- ”IPMA” data on temperature (ºC) and precipitation (mm); Annual. [**?**]

In this project, we intend to prepare our data in order to be able to relate different datasets so that we can extract the information we find most relevant from them, through visualization. After we get to know the data we are dealing with, through data pre-processing and visualization, we feel comfortable choosing the best way to apply machine learning models to it. This way, the models will be applied in order to predict the production of products that make sense in a given region, from the maximum and minimum temperatures and precipitation over the years, that is, from the ””IPMA”” data. We chose to predict production through these models since it was the variable that made the most sense to us; it would not make sense to predict the electricity consumption, from the data we have or even to predict the area of main agricultural crop and productivity, since productivity is a quantity that depends on production and area. Thus, we predicted banana production in Madeira, oranges production in Algarve, olives in Alentejo, and grapes in the North of the country, this being a problem of supervised regression and not of classification. Furthermore, based exclusively on ”IPMA” data, we forecast the precipitation for the year 2018 in the North, since it is the last year of which we have information, becoming a supervised regression problem too.

II. Data description and Pre-processing

A. Data from INE

We start by treating the data from INE. We notice that the datasets for production, productivity, area, and electricity consumption have exactly the same shape. In general, the dataset came with multi-index columns, where at level 0 we had the years and a series of unnamed entries and at level 1 we had the products. The first row had all the entries the same where it just tells us the unit of measurement of the variable in question, i.e. for productivity, kilogram per hectare (kg/ha), production, tons (t), for the area, hectare (ha), and, for energy, kilowatt-hour (kWh). The first column gave the name of each region and the respective region code. Finally, the remaining entries are the measure of the variable for the respective year, region, and product.

As we said, the four datasets have exactly the same shape, and therefore we can create a function that does the data cleaning of them, in a much more practical way, without repeating the same steps over and over again.

In this function we must then input the file name as a string, that is, between ' ', and the number of rows of the dataset. We then make the function read the .csv file given as input, setting the delimiter ";" and the number of rows. To begin the data cleaning, we start by dropping the first row because, as we have already mentioned, it only gives us information about the units of measure, which is not at all important for the interpretation of our data, as well as it may bring future complications having this row only with strings. In addition, we search for null values, and we can see that the last column is full of *NaNs*, so we will drop it. We can also see that the row with the region "Região Autónoma dos Açores" in the datasets for productivity, production, and the area is duplicated, so we should drop one of them. We will drop the 8th column. In the dataset about electricity consumption, row 8 is not repeated. However, keeping in mind that we want to join the datasets in order to be able to relate the information between them, we should stick to just the regions that we have in common in the four datasets. This row, in the electricity consumption dataset, is about a region that is not listed on the other datasets. If we didn't drop it now, we would have to drop it after, together with the other regions that are not common to the other datasets. So, there is no problem in removing now one of those regions that we were going to remove anyway.

Next, we want to shape the dataset into the desired format. We're going to clean up column names by filling in the missing ones. To do this, we will transform the levels where the products are numbered (level one) and where the years are numbered (level zero) into series, and replace all the names at level zero (years) that start with "Unnamed" with null values. Now, we can replace the columns of the dataset with the variables that we have defined as series, renaming the name of the first column as "Region", and renaming the first two levels of the dataset as "products", for the level where the products are numbered (level 1), and "years" for the level where the years are numbered (level 0). Finally, we set "Region" as an index. The advantage is that the series store the data in sequential order. They can take any type of data, but it should be consistent throughout the series.

To simplify future observations, we want to identify the regions only by their code. For example, "PT" corresponds to "Portugal", "1" corresponds to "Continente", "11" to "Norte", etc. To do that, we use a *lambda* function that splits on the ":", in order to create a new column where we have only the region codes. After that, we just have to define that column as an index, change its name to "Region" and drop the old column where we had the code with the name of the region. To finish the shaping we only apply the function `.stack()` to get the desired format for our dataset.

If we check the description of the dataset [5], we verify that the "- -" means that there was no production, which means value "0". It also says that "x" means no record of that specific product. The description says nothing about the "x x" case, so we assume that it means the same as "x". Basically, the "x" and "x x" means null value (*NaN*). We apply these substitutions to our datasets.

We can also see that some entries end with a special character, such as " *" and " &". We must remove these. For this, we apply, again, a *lambda* function and replace these special characters with an empty string. This last function transforms all the inputs into strings, so we have to convert back to integer, if possible, or float. Then, we get our desired dataset, with the level values as the index and with the columns sorted alphabetically.

Now, we are comfortable calling this function, first for productivity, production, and area. We decided to apply again the function `.stack()` to the previous datasets to show the three common indexes: "Region", "years" and all products listed. After that, we `.concat` them, that is, we put them together taking into account the common indexes. We are left with a dataset with 19313 rows, but this did not bring us any problems at the exploration level. We called this final dataset "agriculture".

We call the electricity consumption dataset separately using, of course, the same function and doing `,` selecting only the regions in common with the "agriculture" dataset.

B. Data from "IPMA"

Regarding the "IPMA" data, it comes in different Excel files for various regions. We just want to get information about the maximum and minimum temperatures and precipitation for each of the regions we have from the INE data. To do this, we will create a function that reads the excel files we want to import. The arguments to this function will be the file name, the sheet name, and the region number. We then create a function that joins the files that belong to the same regions into a single data set, doing this for all regions. To complete this, we also add columns with the average precipitation and the average maximum and minimum, for more information and future exploration.

We now can see the information that is given to us in a much more clear way, using the "agriculture" dataset and the "IPMA" dataset.

C. Preparation of the data for future supervised learning

Still preparing our data, we can make sub-datasets so that we can later apply them to the machine learning models we intend to apply.

As mentioned in the Introduction, we will select products and regions that make sense to be related. So, we start by studying the production of bananas in Madeira. For this, we start by selecting the region we want to study (Madeira), as well as the product (Banana) and the variable (Production (t)). We intend to join this data with the "IPMA" data, selecting therefore, information that we have in common in both. However, for Madeira, the "IPMA" dataset only has information up to the year 2018. So, for our sub-dataset, we will only select the data from the "agriculture" dataset until this year. In addition, we also don't have information in the range of years 1986 to 1999 in the "agriculture" dataset, which we have in the "IPMA" dataset. So, we will also drop these years in the latter dataset. Furthermore, we drop the columns

with the average precipitation and the average maximum and minimum temperatures, because we will not need them for our prediction.

Now, we can `.concat()` these two, thus forming the desired sub-dataset, with all the information about the precipitation and maximum and minimum temperatures of the common years for Madeira and with the corresponding production for the product we want (Banana).

We do the same for the remaining combinations we find desirable: orange in Algarve, olive in Alentejo, and wine grape in the North. We can't create a function that does the same transformations for this sub-dataset, which relates the data from the "IPMA" dataset with the "agriculture" data, for all the product-region combinations because there are cases where the years in common are different, so when we apply the `.contact()` function we drop different years, depending on the product and the region we are analyzing.

Since our features are measured at different scales we have to normalize our data. This is because scales with higher numbers end up having more importance in our models, while smaller numbers end up having less, and this could end up misleading our models. For the normalization, we used the maximum absolute scaling. It re-scales each feature between -1 and 1 by dividing every observation by its maximum absolute value.

We call X a sub-dataset that we want to use for prediction (the maximum and minimum temperatures and the precipitation) and y the column we want to predict, that is, the production.

In addition to predicting the output of these product-region combinations, we also intend to forecast precipitation in 2018. For this prediction, we start by defining a region. In this case, we will choose the North region, with region code "11". We will also select a year that we want to predict. We chose to choose the year 2018, because it is the last year that we have a record. As before, we will drop the average precipitation and the average minimum and maximum temperatures, since they won't be useful to us in this case. Thus, we can, at a later stage, make the comparison of the predicted value with the actual value, turning this problem into a supervised regression problem, as we explained in Introduction.

In this case, we must also normalize our data. We have quite different scales, for precipitation and for temperature, and we don't want one of them to end up having more weight than it should in our models. Again, we call X a sub-dataset that we want to use for prediction (the precipitation up until 2018) and y the column we want to predict, that is, the 2018's precipitation.

We now have our data 100% ready for visualization and for implementing the machine learning algorithms.

III. Data Visualization

In this section, we provide some different types of graphics and plots done before the normalization of our features, for a user-friendly comprehension of the data to study before applying any model for prediction. To make a good prediction we must first be familiar with the data we are dealing with, and there is nothing better for this than visualization. Visual and automatic exploration models should always work together.

A. Correlation Matrix between Productivity (Kg/ha), Production (t) and Area (ha)

We made a function where the user can explore by region, choosing it's code. After this, the correlation matrix for productivity, production and area is returned. In this case, we show the correlation matrix for the North region (region code "11"):

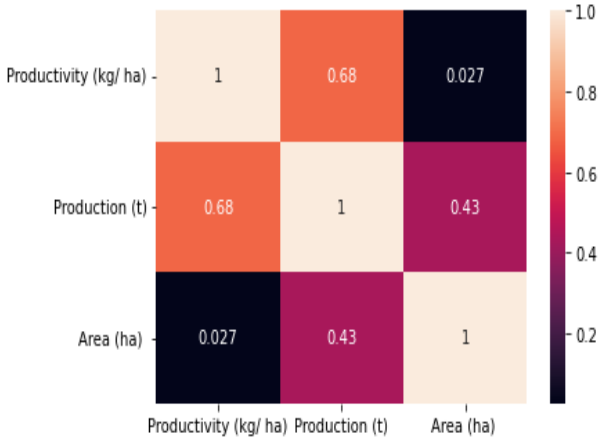


Fig. 1: Correlation Matrix: Productivity vs. Production vs. Area.

As we know, the correlation matrix is a symmetric matrix. Since the units of measurement for productivity are Kg/ha, we defined productivity as the amount of production per unit area. We expected productivity to have a positive correlation with production (the higher the production, the higher the productivity), which is what happens, we can see a correlation of 68% in figure 1. A negative correlation was expected between production and area (the larger the area, the lower the productivity). We obtained a value of 2.7% for this correlation, which is a fairly low positive correlation, which is not exactly what we expected, but it still makes sense given the definition of productivity. Regarding the correlation between area and production, we got a value of 43%, which is a reasonable positive correlation.

B. Correlation Matrix between Electricity Consumption on Agriculture (kWh), Productivity (Kg/ha) and Production (t)

We got the mean of the electricity consumption (kWh) on agriculture by region. The mean of productivity and production by region was also discovered. After this, the correlation matrix between electricity spent on agriculture, productivity (Kg/ha), production (t) is returned. The electricity spent on agriculture is given, on the heatmap, by the name "Agricultura":

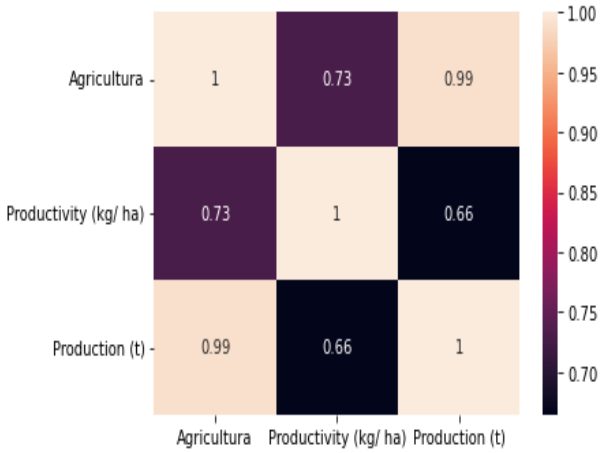


Fig. 2: Correlation Matrix: Electricity spent in Agriculture (kWh) vs. Productivity (Kg/ha) vs. Production (t).

The main objective of this visualization is to test whether the electricity spent on agriculture is related to productivity and production, and how much. Of course, we expected these would be highly correlated, which is what we can confirm by the figure 2. The electricity spent on agriculture and production have a correlation of almost 100%. We also have a fairly strong correlation between electricity consumption in agriculture and productivity, 73%.

C. Correlation Matrix between Production, Productivity, Minimum Temperature (Tmin (average)), Maximum Temperature (Tmax (average)) and Precipitation (average) by year

Here, we will make use of the annual averages we computed in the section Data Description and Pre-Processing for maximum and minimum temperatures, as well as precipitation. We also computed the mean of production and productivity for every year. After this, we obtained the correlation matrix between production (average), productivity (average), tmin (average), tmax (average), and precipitation (average) by year:

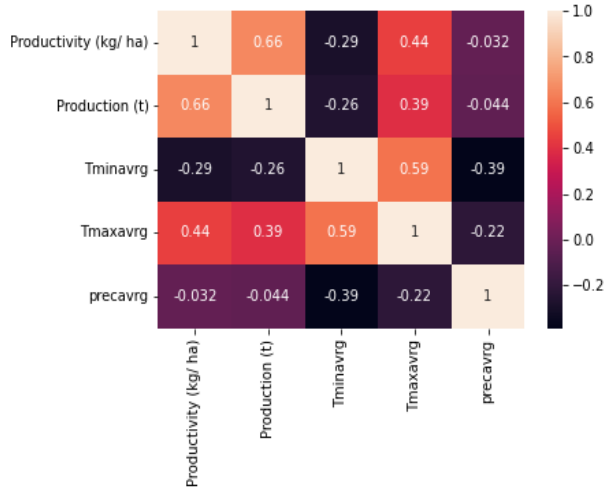


Fig. 3: Correlation Matrix: Productivity (Kg/ha) vs. Production (t) vs. tmin (average) vs. tmax (average) vs. precipitation (average)

What we want to see here is mainly the correlation between tmin, tmax, and precipitation with production and productivity. The average minimum temperature has a negative correlation with production and productivity, of 26% and 29%, respectively. The average maximum temperature has a relatively strong correlation with productivity, 44%, and 39% with production. Contrary to what we expected, the correlation of precipitation with productivity and production is negligible. We got a negative correlation of 3.2% and 4.4%.

D. TOP Product with more Production by Region

We decided to create a function that allows the user to see how the most produced product from a certain region behaved over time. We give the user the chance to explore the region he is interested in, by selecting the region code in this function.

We created a variable ”top” that returns a dataframe with the most produced products for each region. We got that the top products for each region were:

Region	products	Production (t)
1	Principais culturas forrageiras	49999109.0
11	Principais culturas forrageiras	22659587.0
15	Citrinos	7607623.0
16	Batata	13856393.0
17	Principais culturas para indústria	6952432.0
18	Principais culturas para indústria	34919019.0
20	Milho forrageiro	3987451.0
3	Batata	1323408.0
PT	Principais culturas forrageiras	54012370.0

Fig. 4: Top Products and respective Production (t), per Region

Where the meaning of the region codes are given by:

Region Code	Region
1	Continente
11	Norte
15	Algarve
16	Centro
17	Área Metropolitana de Lisboa
18	Alentejo
20	Região Autónoma dos Açores
3	Região Autónoma da Madeira
PT	Portugal

TABLE I: Region code atribution.

We also created another function that returns the production for each year of the chosen product in the desired region. Using the two previous functions, we show the line chart with a time evolution of the top product, in this case for the region Portugal (region code ”PT”):

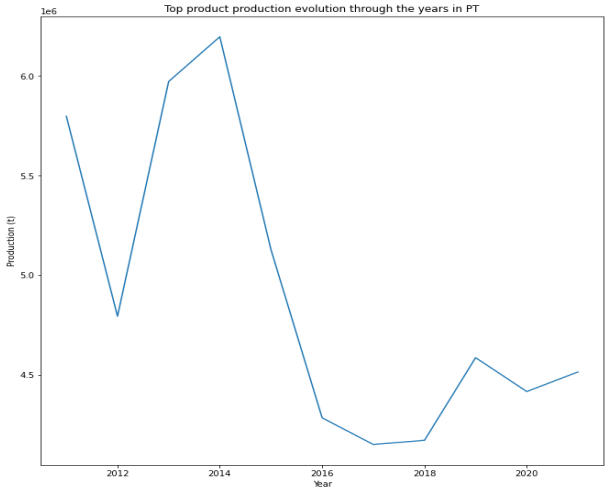


Fig. 5: Time evolution of the top product production in Portugal.

We know from table I that the top product in Portugal (region code PT) is ”Principais culturas forrageiras”, and we can see it’s time evolution by the chart of figure 5.

For the visualizations that follow we use as reference the agricultural statistics made by INE referring to the years 2021 [8], 2013 [7] and 2004 [6], so that we can understand certain phenomena that explain our visualizations.

E. Precipitation mean through the years - bar chart

We will explore the evolution of average annual precipitation from 1986 to 2020 since the ”IPMA” data only gives us information up to 2020. In blue we present the evolution for the regions north of the Tejo river and, in orange, the regions south of the Tejo river:

For the regions north of the Tejo river we consider: North, Center, and ”Área Metropolitana de Lisboa”.

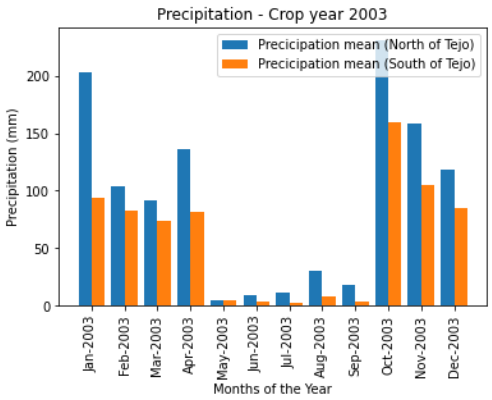
For the regions south of the Tejo we consider: ”Alentejo” and ”Algarve”.

Thus, in this visualization, the regions: ”Portugal”, ”Continente”, ”Região Autónoma dos Açores,” and ”Região Autónoma da Madeira” are not considered.

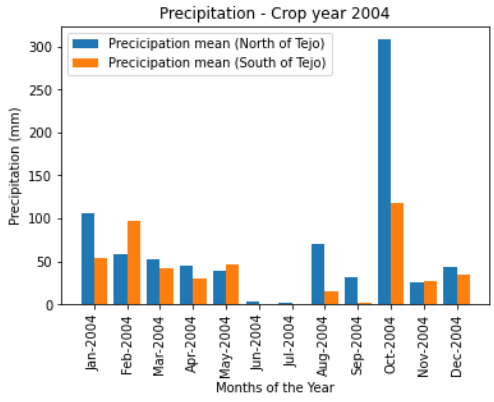
We then made a function where the user can choose the year he wants to get information about. Here, we will show some examples of how it can be applied, as well as conclusions that could be drawn from the exploration.

If we try to apply the visualization for years equal to or greater than 2019 we can see that we only get information about the average rainfall for the region north of the Tejo. This is because in the ”IPMA” dataset itself, we don’t have rainfall information for those regions in those years.

Take, for example, the years 2003/2004:



(a) Precipitation - Crop year 2003



(b) Precipitation - Crop year 2004

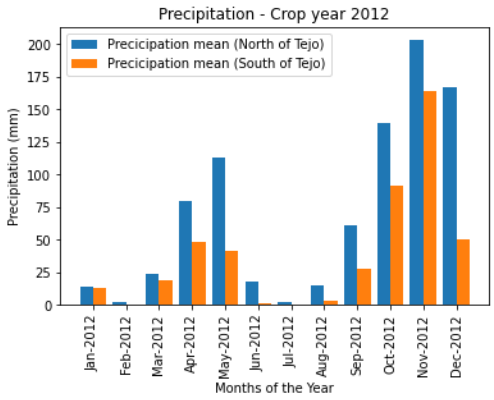
Fig. 6: Precipitation of agricultural year 2003/2004.

According to the agricultural statistics of the 2003/04 agricultural year from INE [6], the heavy rainfall at the end of November and beginning of December is confirmed. To this is added the heavy rainfall in the month of October. According to these, the occurrence led to the interruption of the autumn-winter sowing works. The improvement of weather conditions allowed the completion of the sowing of cereal crops (such as oats and wheat, for example).

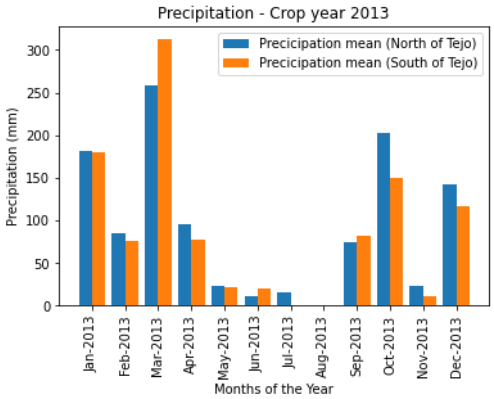
We also verified the beginning of a dry summer. However, unusually heavy rainfall was recorded in August for the

season, which did not, however, have negative repercussions on agriculture.

We also noticed some interesting aspects in the years 2012/2013:



(a) Precipitation - Crop year 2012



(b) Precipitation - Crop year 2013

Fig. 7: Precipitation of agricultural year 2012/2013.

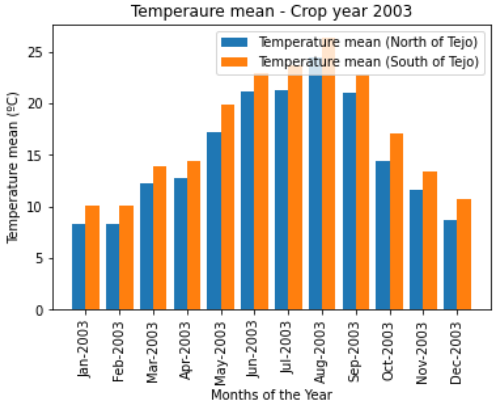
The crop year 2012/2013 was marked by tornadoes (Barlavento Algarvio, November 16, 2012) and storms with exceptionally heavy rainfall and winds (the whole territory, January 19, 2013), which caused high losses on many farms.

There were very high precipitation values, and the beginning of spring was very rainy. It was the second wettest March in mainland Portugal in the last 50 years, to date. This situation conditioned the soil preparation works and the sowing/planting of spring-summer crops, as well as the development of the crops installed. On the other hand, it allowed irrigation needs to be met.

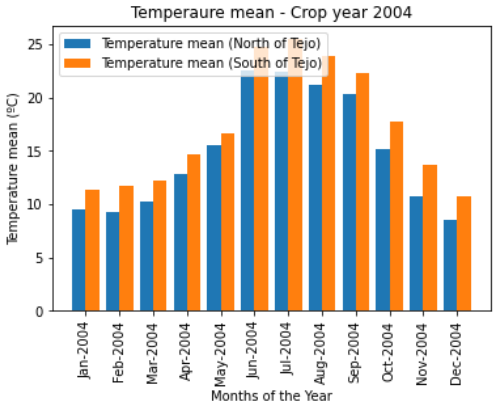
The summer was dry, but promoted the normal development of crops.

F. Temperature mean through the years - bar chart

Here we do the same thing we did for precipitation in the section Precipitation mean through the years - bar chart, but this time for temperature. That is, we explore the evolution of the annual average temperature (average of the maximum and minimum temperatures). Again the data for the visualization is taken from the "IPMA" dataset and the regions are distributed in the same way as in the previous visualization (north of Tejo and south of Tejo). Again, we made a function where the user can choose the year he wants to collect information about. Here we will show some examples for the years we explored in the section Precipitation mean through the years - bar chart. For the years 2003/2004 we got:



(a) Temperature mean - Crop year 2003

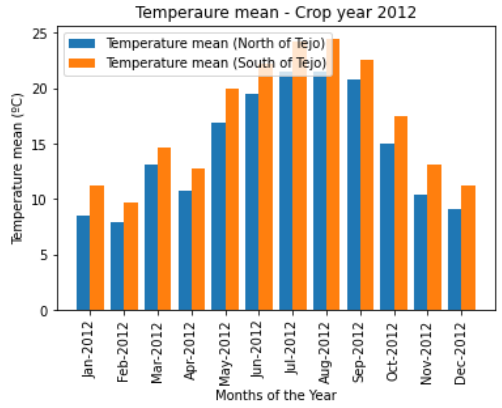


(b) Temperature mean - Crop year 2004

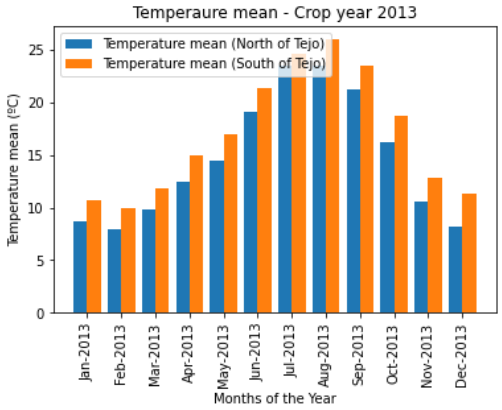
Fig. 8: Temperature mean of agricultural year 2003/2004.

Besides the drought for this agricultural year, there was a very hot summer, as can be seen in the months of June and July, mainly, where the average maximum and minimum temperature reaches around 25ºC.

For the years 2012/2013:



(a) Temperature mean - Crop year 2012



(b) Temperature mean - Crop year 2013

Fig. 9: Temperature mean of agricultural year 2012/2013.

Again, for this agricultural year, besides having a hot summer, we also had a very dry summer, as we can conclude by comparing figure 7 and figure 9. The heat waves recorded in June, July (more extensive in territorial terms) and August, occasionally damaged some crops, particularly those in full bloom / fruit-vegetation, according to agricultural statistics given by INE [7] and confirmed by our visualization.

G. Area Vs. Production Line charts

In this section, we will present a series of line charts where we compare the area and production of the products we find most interesting to explore in Portugal. This way, for this visualizations we are going to use the ”Area(ha)” and ”Prodution(t)” columns from the ”agriculture” dataset.

1) Autumn and Winter Cereals - Area vs Production line charts: The cereals present in the ”agriculture” dataset are: ”Arroz, Aveia, Aveia forrageira, Centeio, Cereais para grão, Cevada, Milho, Milho forrageiro, Milho regadio, Milho sequeiro , Trigo, Trigo duro, Trigo mole and Triticale”. As autumn and winter cereals, we will use: ”Aveia, Aveia forrageira, Centeio, Cevada, Cereais para grão, Trigo, Trigo duro, Trigo mole and Triticale”.

As we are going to make more comparisons of this kind, that is, between the area and production of a given product, we have created a function that allows the user to choose the region that he wants to analyze, in this example we chose Portugal (region code ”PT”):

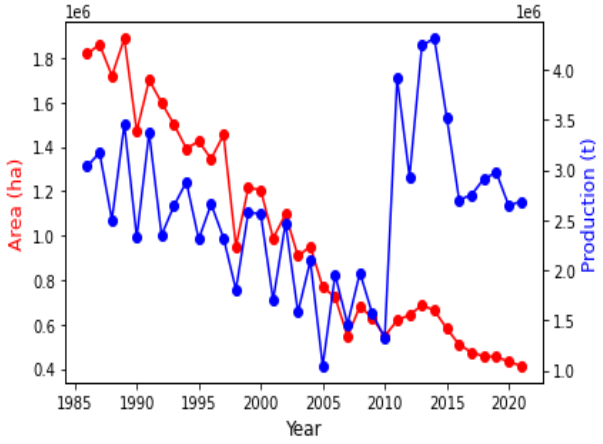


Fig. 10: Autumn and Winter Cereals PT - Area Vs. Production

The usual periods of rainfall at this time of year, hinder the sowing of autumn/winter cereals, with the interruption of these operations when conditions were not agronomically acceptable (waterlogged soils).

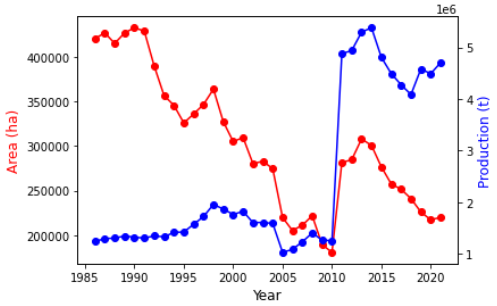
A large part of the winter cereal harvest was finished by the end of July, thus registering decreases in all winter cereals, with the exception of durum wheat, whose significant increase in area led to an increase in production.

We can also see a decrease in area and production, compared to previous years, until approximately 2010. However, starting in 2010 there is a sharp increase in production accompanied by a brief increase in area as well. In this case we see a greater difference between area and production.

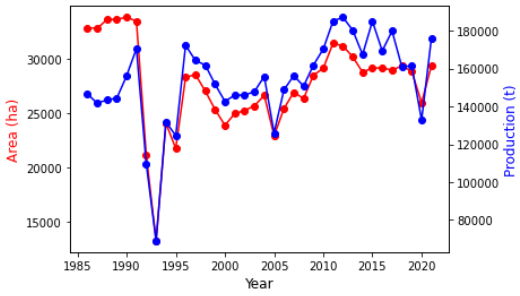
2) Spring and Summer Cereals - Area vs Production line charts: We do the same comparison as the previous visualization of figure 10 but, in this case, for spring/summer cereals. As spring/summer cereals we consider: ”Arroz, Milho, Milho forrageiro, Milho regadio, Milho sequeiro”.

We will do two separate visualizations, one for ”Arroz” and one for ”Milho”. In the case of ”Milho” we consider all types of it (we make a sum of them): ”Milho, Milho forrageiro, Milho regadio, Milho sequeiro”.

Again, we have created a function that allows the user to choose, the region that they want to analyze and will show here the visualization for the region ”PT”:



(a) Milho PT - Area Vs. Production



(b) Arroz PT - Crop year 2013

Fig. 11: Spring and Summer Cereals - Area Vs. Production.

For the corn ("Milho") visualization, we can see that until 1995 there's an "odd" relationship between area and production: area, in general, decreases, yet production increases. For the following years we verify an approximate evolution between area and production. Again, we see a very sharp increase in 2010, as we saw for autumn and winter cereals. Therefore, we can assume that from 2010 onwards there was a high production of cereals, in general.

For the rice ("Arroz"), we can see that area generally follows production. They are highly correlated for this product and this region. The decrease in production is due to the decrease in cultivation area and also due to the lack of light and heat during the summer, the uncontrolled emergence of pyriculariosis 1 and the high degree of infestation of the fields by millet, for example [7].

3) Tomatoes for industry - Area vs Production line charts: Let's now compare the area and production of tomatoes for processing. Again, we have created a function that allows the user to choose, the region that they want to analyze and here we will show for the region "PT":

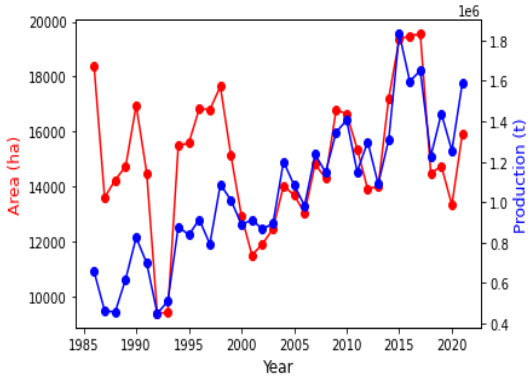


Fig. 12: Tomatoes for industry - Area Vs. Production

From 2000 on, approximately, we can see that the area follows, once again, the production, very closely. We have seen, from that year on, a constant evolution in production and, consequently, in the area under cultivation.

4) Potatoes - Area vs Production line charts: Let's now compare the area and production of potatoes for the region PT:

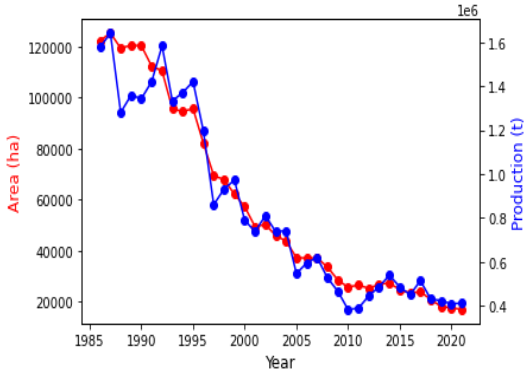


Fig. 13: Potatoes - Area Vs. Production

We can see a general decrease in both production and area from 1986 to 2021. Again, we see that production and area are highly correlated, for this product and this region.

5) Fresh fruit, subtropical fruits, and citrus - Area vs Production line charts: Now, we will look at the production and cultivation area for fresh fruit, subtropical fruits, and citrus. We do the exploration, individually, for apple, pear, peach, cherry, and orange, in figures 21 to 25 in Appendix.

Regarding the visualization for the apple production and area in the region "PT", we can see that there is practically no correlation between area and production for this product and this region. We can see a sharp decrease in the area of cultivation between 1997 and 2010, approximately. However, in general, there is an increase in apple production over the years.

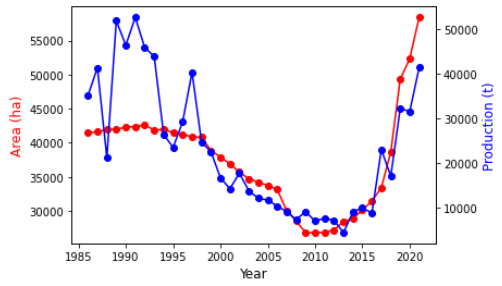
For the pear, again, we can see that there is practically no correlation between area and production for this product and this region. However, over the years we see, in general, an increase in production of pears and a decrease in the area under cultivation.

For the peach, we already see a certain relationship between the evolution of production and the area under cultivation. In general, there is a large decrease in peach production and area over the years.

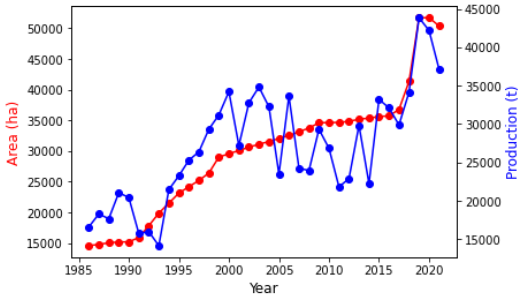
We can see that the cherry growing area grew tremendously from about 1990 on, and has been growing ever since, generally speaking. The production has had a slight increase over previous years.

As for the orange, we can verify that the evolution of the orange growing area had almost a negative quadratic form, we see an increase until around 2000 and then a sharp decrease until 2010, approximately. In general, we see an increase in production over the years, but there is a decrease between 2003 and 2008, approximately. The sustained upward trend in production since 2015 is confirmed.

6) Ripe Nuts - Area vs Production line charts: We will now explore the nuts by making, as before, the comparison between the area under cultivation and the production. To do this, we will explore almonds and chestnuts individually, for the region ”PT”:



(a) Almond PT - Area Vs. Production



(b) Chestnut PT - Crop year 2013

Fig. 14: Ripe Nuts - Area Vs. Production.

In the visualization for the almond’s production and area, we see a positive quadratic evolution of both production and area under cultivation. After a systematic reduction of the almond area between 1994 and 2009, interest in this crop resurged, with the installation of new intensive orchards. From 2010, the almond area began a continuous process of increase, with reflection, necessarily delayed, in production (intensive almond orchards begin production between the second and third year after planting and reach full production in about seven years.

As for the chestnut, the decreases in this product’s production can be justified by the occurrence of several periods of rainfall and not very high average temperatures, enhancing the appearance and development of septoria, a disease caused by the fungus Mycosphaerella maculiformis, which usually has marginal occurrence and little economic impact. The cultivation area has been growing over the years, in general, with a small decrease in the last 2 years.

H. Precipitation Vs. Production Line charts

With this visualization, we intend to explore the production of the wine grape. For this case, it makes sense to make a comparison between wine grape production and precipitation during the months that the grapes start to bloom, i.e. precipitation from April to May (1986-2021), since precipitation has a strong influence at this stage. From this, we get, for the region ”PT”:

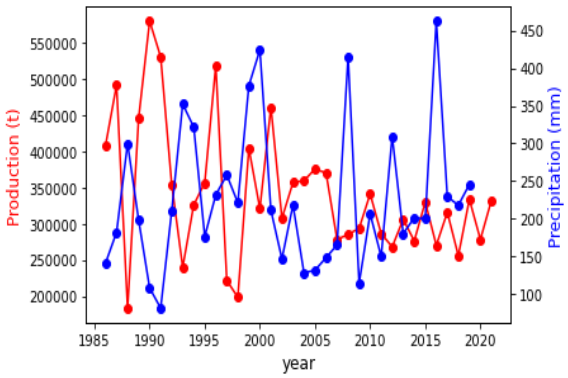


Fig. 15: Wine Grape - Precipitation Vs. Production.

Annual wine production in the last two decades has been stabilizing, in contrast to the large fluctuations in the early years, possibly related to the wet conditions in April and May (in years when the spring was dry, yields were generally higher).

I. Olive oil - Production line chart

As we saw in the Introduction, Portugal is among the top ten largest olive oil producers in the world and is the fourth biggest exporter. [2] This way, in this thread we will limit ourselves to exploring the evolution of olive oil production through the years:

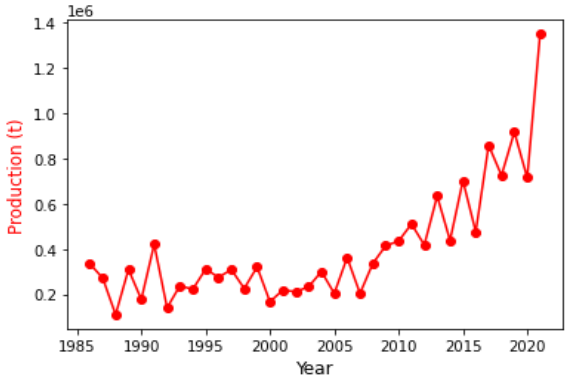


Fig. 16: Olive oil - Production.

We can see, in general, an increase in olive production for olive oil. We mainly see a sharp increase from 2019 to now.

IV. Implemented Machine Learning Models

After having pre-processed the data and after we have explored visualization of this data, we proceeded to the implementation of ML Models. For our solution, we implemented the four algorithms below.

The hyperparameters of our algorithms will be described below as well. To make sure we get the best performance from this algorithm, we will use GridSearchCV. [15]

In the context of grid search, the "CV" parameter refers to the number of folds used in the cross-validation process. For example, if "CV" is set to 5 (which is what we will be using), then the data will be split into 5 folds, and the model will be trained and evaluated 5 times, each time using a different fold as the test set and the remaining folds as the training set. The performance of the model will then be averaged across the 5 iterations. This is an important parameter to set, as it helps to ensure that the model is being thoroughly evaluated and that the hyperparameter combination chosen is robust and generalizable to unseen data.

A. Linear Regression

In linear regression, the goal is to find the line of best fit that can be used to make predictions about the dependent variable based on the independent variable(s). The line of best fit is represented by the equation:

$$y = mx + b \tag{1}$$

Where y is the dependent variable, m is the slope of the line, x is the independent variable, and b is the y-intercept (i.e., the point at which the line crosses the y-axis).

To find the line of best fit, we need to find the values of m and b that minimize the sum of the squared errors between the predicted values (i.e., the values on the line of best fit) and the actual values. This is done using the least squares method, which finds the values of m and b that minimize the following cost function:

$$J(m, b) = \sum_{i=1}^{n} (y_i' - y_i)^2 \tag{2}$$

where y_i' is the predicted value for the ith data point and y_i is the actual value.

Once the values of m and b have been determined, the line of best fit can be used to make predictions about the dependent variable based on new independent variable(s) using the following equation:

$$y' = mx + b \tag{3}$$

where y' is the predicted value for the dependent variable, m is the slope of the line, x is the independent variable, and b is the y-intercept.

There are several parameters that we can set when using a linear regression algorithm:

- **fit_intercept:** a boolean value that indicates whether or not an intercept term (also known as the y-intercept) should be added to the model. The default value is True, which means that an intercept term will be included;
- **normalize:** a boolean value that indicates whether or not the input variables should be normalized before fitting the model. The default value is False, which means that the input variables will not be normalized;
- **copy_X:** a boolean value that indicates whether or not the input data should be copied before fitting the model. The default value is True, which means that a copy of the input data will be made;
- **n_jobs:** an integer value that indicates the number of CPU cores that should be used when fitting the model. The default value is None, which means that all available cores will be used.

[12]

B. Lasso Regression

The Lasso is an optimization technique used to estimate sparse linear regression. It is a useful technique in situations where some of the input variables may not be informative or may not be needed to make accurate predictions.

Lasso regression is a type of linear regression that uses L1 regularization to shrink the coefficients of the model towards zero. L1 regularization is a technique that adds a penalty to the objective function that is being optimized, which encourages the coefficients to be small.

In lasso regression, the objective function is defined as the sum of the squared errors between the predicted values and the actual values, plus the regularization term. The regularization term is defined as the sum of the absolute values of the coefficients, multiplied by a constant alpha. The objective function can be written as follows:

$$Objective\ function = \sum_{i=1}^{n} (y_i - y'i)^2 + \alpha \sum_{i=1}^{n} |w_i| \tag{4}$$

where y is the vector of actual values, y' is the vector of predicted values, w is the vector of coefficients, and $alpha$ is the strength of the regularization.

The lasso regression algorithm aims to find the values of the coefficients (w) that minimize the objective function. To do this, it uses an optimization algorithm, such as gradient descent, to update the coefficients iteratively until the objective function is minimized.

As the $alpha$ value increases, the regularization term becomes more dominant and the coefficients are more heavily penalized. This results in a smaller number of features being selected and a simpler model. Conversely, as the $alpha$ value decreases, the regularization term becomes less dominant and the model is able to fit the data more closely, potentially leading to a more complex model.

There are several hyperparameters that we can set when using a lasso regression algorithm:

- **alpha:** Constant that multiplies the L1 term. Defaults to 1.0;
- **fit_intercept:** whether to calculate the intercept for this model. If set to False, no intercept will be used in calculations;
- **normalize:** This parameter is ignored when fit_intercept is set to False. If True, the regressors X will be normalized before regression by subtracting the mean and dividing by the l2-norm;
- **precompute:** whether to use a precomputed Gram matrix to speed up calculations;
- **copy_X:** If True, X will be copied; else, it may be overwritten;
- **max_iter:** The maximum number of iterations;
- **tol:** The tolerance for the optimization: if the updates are smaller than tol, the optimization code checks the dual gap for optimality and continues until it is smaller than tol;
- **warm_start:** When set to True, reuse the solution of the previous call to fit as initialization, otherwise, just erase the previous solution.

[11]

C. Ridge Regression

This class provides a similar interface to other linear models in scikit-learn, such as Linear Regression and Lasso Regression.

The Ridge Regression is an optimization technique used to estimate linear regression models with L2 regularization term. It tries to prevent overfitting by adding a penalty term to the cost function, this term is the sum of the squares of the coefficients, which is a L2-norm. It is a useful technique in situations where all the input variables are needed to make accurate predictions but the model may be overfitting, the regularization term helps to reduce the size of the coefficients and thus the complexity of the model.

There are several hyperparameters that we can set when using a ridge regression algorithm:

- **fit_intercept:** whether to calculate the intercept for this model. If set to False, no intercept will be used in calculations;
- **normalize:** This parameter is ignored when fit_intercept is set to False. If True, the regressors X will be normalized before regression by subtracting the mean and dividing by the l2-norm;
- **copy_X:** If True, X will be copied; else, it may be overwritten;
- solver: ‘auto’, ‘svd’, ‘cholesky’, ‘lsqr’, ‘sparse_cg’, ‘sag’, ‘saga’

The solver to use in the computational routines:

- ‘auto’ chooses the solver automatically based on the type of data;
- ‘svd’ uses a Singular Value Decomposition of X to compute the Ridge coefficients. More stable for singular matrices than ‘cholesky’;
- ‘cholesky’ uses the standard scipy.linalg.solve function to obtain a closed-form solution;
- ‘lsqr’ and ‘sparse_cg’ use iterative procedures. They are more appropriate for cases where n_samples ¿ n_features;
- ‘sag’ uses a Stochastic Average Gradient descent, and ‘saga’ uses its variant SAGA. ‘sag’ and ‘saga’ solvers also supports ‘warm_start’.

[13]

D. Decision Tree

A decision tree is a type of supervised learning algorithm that is often used for regression and classification tasks. In decision tree regression, the goal is to create a model that can predict a continuous target variable based on input features. The decision tree algorithm works by recursively partitioning the input space into smaller regions, called leaves, with the goal of creating regions that are homogeneous with respect to the target variable. Each internal node of the tree represents a test on one of the input features, and each leaf node represents a prediction.

The algorithm starts by selecting the feature and the value that best split the data into subsets that are more homogeneous (pure) with respect to the target variable, the feature with the highest information gain is chosen as the root of the tree. The process is then recursively repeated on each subset of the data, until a stopping criterion is met (e.g., a maximum tree depth is reached or a minimum number of samples is reached in a leaf).

There are several hyperparameters that we can set when using a decision tree regression algorithm:

- **max_depth:** the maximum depth of the tree. The default value is None, which means that the tree can grow until all the leaves contain less than min_samples_split samples;
- **min_samples_split:** the minimum number of samples required to split an internal node. This is used to control the complexity of the model and prevent overfitting. The default value is 2;
- **min_samples_leaf:** the minimum number of samples required in a leaf node. This is used to control the complexity of the model and prevent overfitting. The default value is 1;

- **max_leaf_nodes:** the maximum number of leaf nodes. This is used to control the complexity of the model and prevent overfitting. The default value is None;
- **min_weight_fraction_leaf :** the minimum weighted fraction of the sum total of weights (of all the input samples) required to be at a leaf node. Samples have equal weight when sample_weight is not provided;
- **max_features :** the number of features to consider when looking for the best split;
- **random_state :** a seed for the random number generator used when shuffling the data to allow reproducible results.

[16]

One thing we can do for this algorithm that we cannot do for the others we applied, is to look at the feature importance. In this case, we will plot the importance of each feature in the model.

V. Evaluation of the Implemented Models

To evaluate the models we used the metric MASE. Mean absolute squared error (MASE) is a measure of the difference between the predicted values of a model and the true values. It is similar to mean squared error (MSE), but instead of squaring the differences between the predicted and true values, it takes the absolute value of the differences. MASE is often used in time series analysis, as it is more robust to the presence of outliers and can handle data with different scales and units.

MASE is calculated by taking the sum of the absolute differences between the predicted values and the true values and then dividing them by the number of data points. Mathematically, MASE can be expressed as:

$$MASE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{5}$$

where n is the number of data points, y is the true value, and $\hat{y}$ is the predicted value. [14]

This way, the MASE will be returned by all the algorithms and used as and used as a comparison of them. To do this, we will return a table where we have the values of MASE for each algorithm for easier comparison.

VI. Application and Analysis of the ML Models

We decided to analyse:

- banana production in Madeira;
- orange production in Algarve;
- olive oil production in Alentejo;
- wine grapes in the North.

In addition to this, in relation exclusively to the "IPMA" dataset, we made the prediction of precipitation for the year 2018, as we explained in Introduction.

A. Banana production in Madeira

We can see the output of the models' MASE comparison (with the hyperparameters that got the best scores) for the banana production in Madeira in figure 26 in Appendix.

By analysing the table we can see that the decision tree regression has the best evaluation (it has the lowest MASE). The worst model was linear regression, with the highest MASE. However, overall, our models behaved well with our worst model having an MASE of about 19%, which is not bad at all.

For the feature importance of the Decision Tree Regression model for this product we got the output:

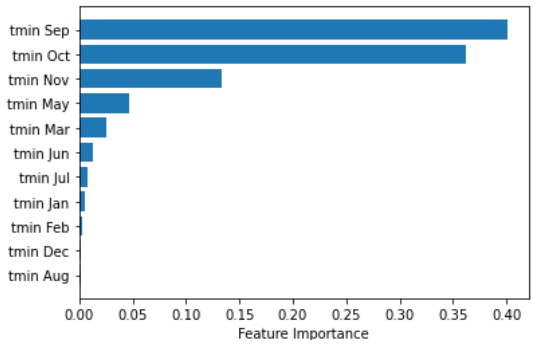


Fig. 17: Feature Importance for Decision Tree Regression - Banana production in Madeira.

In this graph only the features that actually had the most importance for the model appear. We can see that the minimum temperatures are great predictors of banana production in Madeira, being the minimum temperatures of September, October and November the features with more importance, in general.

B. Orange production in Algarve

Just like before, we can see the output of the models' MASE comparison (with the hyperparameters that got the best scores) for the orange production in Algarve in figure 27 in Appendix.

Using the previous results as a comparison, we can see that, in general, we have a higher MASE that the previous product-region combination. But, in this case, linear regression has the best result whereas the other three are worse. In this case lasso and ridge are very similar, as they were before.

For the feature importance of the Decision Tree Regression model for this product we got the output:

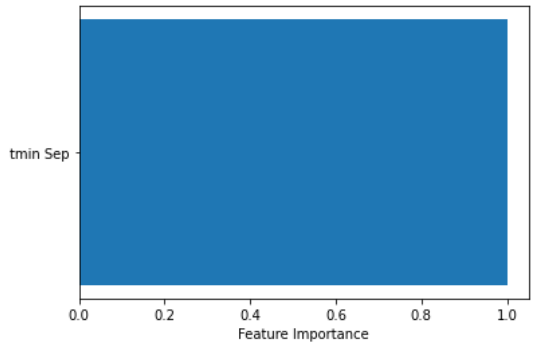


Fig. 18: Feature Importance for Decision Tree Regression - Orange production in Algarve.

In this case, we only got one feature in the feature importance graph, the September minimum temperature. It appears to have an importance of 100%.

Usually, if only one feature appears in the feature importance plot with a relative importance of 100%, this might indicate that this feature alone is able to predict the target variable well, and that the other features are not needed for this task. However, this could also be an indication of a problem with the model or the data.

A few possible reasons for this could be:

- The other features in the data set are not informative, they do not have a significant relationship with the target variable, they are not useful in making predictions.
- The model is overfitting to the training data, it's memorizing the data instead of generalizing it, in this case, the model is not generalizing well to unseen data, and the feature importance on the test set will not reflect the true importance of the feature.
- The other features have been pre-processed in a way that has removed their information, or they are highly correlated with the most important feature, they might have been combined or removed from the dataset, which could cause a single feature to have all the importance.

[17]

The orange season is usually November/December, so it makes sense to us that the September minimum temperature is quite important in the prediction of orange production.

C. Grape production in the North Region

We can see the output of the models' MASE comparison (with the hyperparameters that got the best scores) for the grape production in the North region in figure 28 in Appendix.

This is really the product-region combination where the models perform best (given the low MASE values). The MASE values of the four models are very close. Even so we can see that the model with the lowest MASE was the decision tree regression just as it was for Banana production in Madeira.

For the feature importance of the Decision Tree Regression model for this product we got the output:

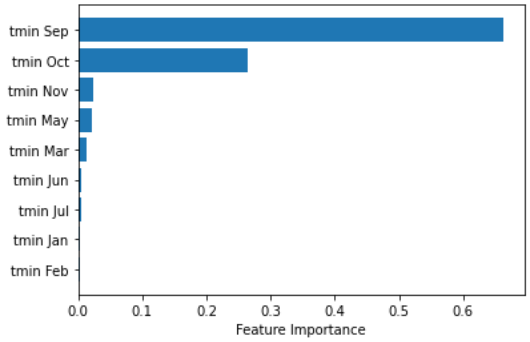


Fig. 19: Feature Importance for Decision Tree Regression - Grape production in the North region.

We can see that, again, the September minimum temperature has the greatest relative importance, at about 65%. Thus, the September and October minimum temperatures are the ones that contribute most to the grape production prediction.

D. Olive production in Alentejo

We can see the output of the models' MASE comparison (with the hyperparameters that got the best scores) for the olive production in Alentejo in figure 29 in Appendix.

Of all product-region pairs analyzed this one is, by far, the one which has the worst evaluation at all. However linear regression and decision tree have way better results. Comparing only these two, the linear regression model stands out for its positivity.

For the feature importance of the Decision Tree Regression model for this product we got the output:

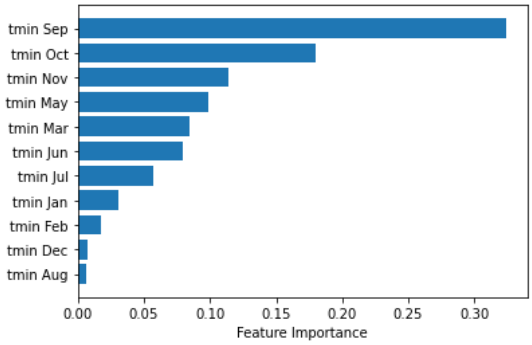


Fig. 20: Feature Importance for Decision Tree Regression - Olive production in Alentejo.

As we saw when we analyzed the feature importance for Banana production in Madeira, here too we only have minimum temperatures as features with more importance, and again the September minimum temperature has the highest relative importance. However, unlike what happened before, although the features with the highest relative importance are again September, October and November, we can see that the remaining minimum temperatures have more contribution to the model than what happened in the case of Banana production in Madeira.

E. Precipitation forecast for 2018

Another demonstration of the models chosen is presented in this following section. We predict the precipitation for the most recent year with data, in this case 2018.

For the prediction of the 2018's precipitation we have only used the linear regression model since it was the one with the best results for this particular section, even though we tested others. We got a MASE of 9.8%. We also compute the RMSE and get a value of, approximately, 25.9%.

In linear regression, the Root Mean Squared Error (RMSE) is a commonly used metric to evaluate the performance of the model. RMSE measures the average magnitude of the error of the model's predictions, it is defined as the square root of

the average of the squared difference between the predicted values and the actual values. The formula for the RMSE is:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2} \quad (6)$$

Where y_i is the true value of the i-th sample, and $\hat{y_i}$ is the predicted value, n is the number of samples.

The RMSE will always be non-negative, and the lower the RMSE, the better the model. A low RMSE value indicates that the model's predictions are close to the true values, and therefore, the model is considered to be performing well. The squared term makes larger differences more pronounced than smaller ones and thus makes it more sensitive to larger errors.

However, the RMSE is sensitive to outliers and the magnitude of the errors, that's why the Mean Absolute Error (MAE) is often used as an alternative. [10]

We then build a time series with the comparison between the predicted and the original data of that year, that we can see in figure 30 in Appendix.

Taking a look at the graph, we can see that, in general, our model predictions are very similar to the original ones. Also we can verify that summer and autumn are the seasons with the most accuracy.

Since we are dealing with a regression problem and not a classification problem, we cannot present, for instance, the ROC (Receiver Operating Characteristic) curve and the confusion matrix or classification report, since these only apply to classification problems.

VII. Conclusions and Future Work

In conclusion, this work has presented a comprehensive analysis of the datasets we had, including data cleaning and pre-processing, data visualization, and machine learning modeling. The data cleaning and pre-processing steps were essential in ensuring the quality and integrity of the data, and in making it suitable for analysis. This step helped to identify and remove any missing, duplicate, or irrelevant data.

The data visualization step helped to gain insights into the distribution and relationships between the features and the target variable. The various plots and charts used in this step helped to identify patterns and trends in the data, which are important for understanding the underlying structure of the data set.

Finally, several machine learning models were trained and tested on the pre-processed data. The results of these models have shown that Linear regression, Decision Tree Regression, Lasso Regression, and Ridge Regression have all been able to make accurate predictions with high degree of accuracy, in the majority of cases. The best model was selected based on the Mean Absolute Error (MAE) metric.

Regarding the Decision Tree Regression model, we also plotted feature importance. We found that the same feature has the most importance in different analyses, the September minimum temperature. It could be an indication that this feature is a strong predictor of the target variables. However, as future work, it would be a good idea to confirm this finding by performing other types of analyses, such as correlation analysis or hypothesis testing, to make sure that this feature is indeed the most informative.

It's also important to keep in mind that even if a feature has high importance, it does not mean that it alone can predict the target variable perfectly. There are other features that might also be important but are not as well correlated with the target variable as the feature in question. Additionally, it might also mean that other features are highly correlated with the feature that has high importance.

Finally, another thing to keep in mind is that feature importance can change when we run the model multiple times. This can be happening because we are using a sample of data to train the model, different samples can have different feature importances. Another justification for this may be hyperparameters tuning, using cross validation.

In summary, this work demonstrates the importance of data cleaning and pre-processing, data visualization, and machine learning modeling in analyzing and understanding complex datasets. It also highlights the potential of machine learning models to make accurate predictions and identify important relationships in the data.

References

[1] 2013 top 10 wine countries. http://www.transowine.com/news/68-general/132-2013-top-10-wine-countries.html. Accessed: 2023-01-05.

[2] Alentejo olive oil. https://azeitedoalentejo.pt/en/o-azeite-do-alentejo/. Accessed: 2023-01-05.

[3] Consumo de energia. https://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine_destaques&DESTAQUESdest_boui=512530049&DESTAQUESmodo=2. Accessed: 2022-10-01.

[4] De portugal para o mundo: produtos florestais líderes de mercado. https://florestas.pt/valorizar/de-portugal-para-o-mundo-produtos-florestais-lideres-de-mercado/. Accessed: 2023-01-05.

[5] Estatísticas agrícolas. https://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine_publicacoes&PUBLICACOESpub_boui=129211&PUBLICACOESmodo=2&xlang=pt. Accessed: 2022-10-01.

[6] Estatísticas agrícolas - 2004. https://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine_publicacoes&PUBLICACOESpub_boui=129211&PUBLICACOESmodo=2. Accessed: 2022-10-01.

[7] Estatísticas agrícolas - 2013. https://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine_publicacoes&PUBLICACOESpub_boui=210756829&PUBLICACOESmodo=2. Accessed: 2022-10-01.

[8] Estatísticas agrícolas - 2021. https://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine_publicacoes&PUBLICACOESpub_boui=31589846&PUBLICACOESmodo=2. Accessed: 2022-10-01.

[9] Instituto nacional de estatística. https://www.ine.pt/xportal/xmain?xpgid=ine_main&xpid=INE&xlang=pt. Accessed: 2022-10-01.

[10] Rmse: Root mean square error. https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/. Accessed: 2022-12-01.

[11] sklearn.linear_model.lasso. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html. Accessed: 2022-12-01.

[12] sklearn.linear_model.linearregression. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html. Accessed: 2022-12-01.

[13] sklearn.linear_model.ridge. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html. Accessed: 2022-12-01.

[14] sklearn.metrics.mean_absolute_error. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_error.html. Accessed: 2022-12-01.

[15] sklearn.model_selection.gridsearchcv. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html. Accessed: 2022-12-01.
[16] sklearn.tree.decisiontreeregressor. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html. Accessed: 2022-12-01.
[17] Understanding feature importance and how to implement it in python. https://towardsdatascience.com/understanding-feature-importance-and-how-to-implement-it-in-python-ff0287b20285. Accessed: 2022-12-01.

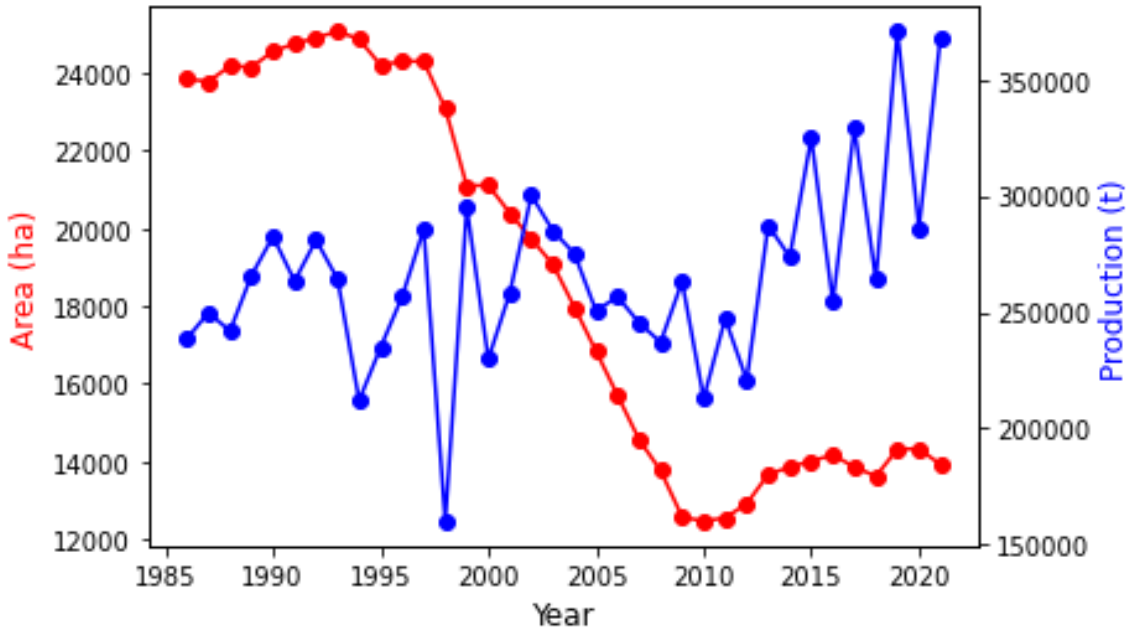


Fig. 21: Apple PT - Area Vs. Production.

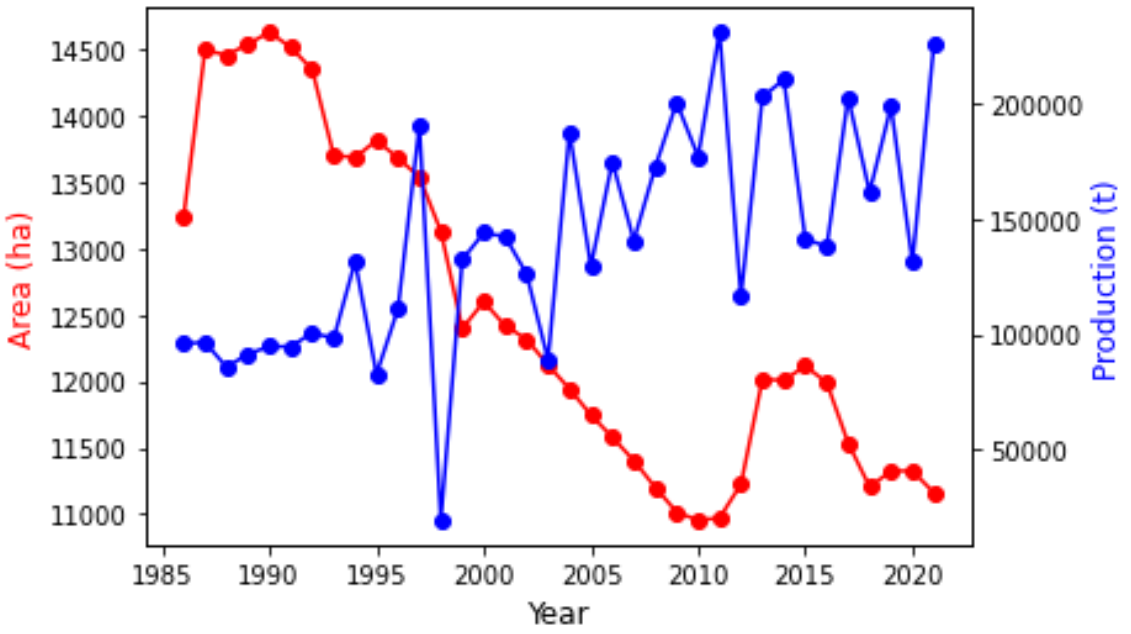


Fig. 22: Pear PT - Area Vs. Production.

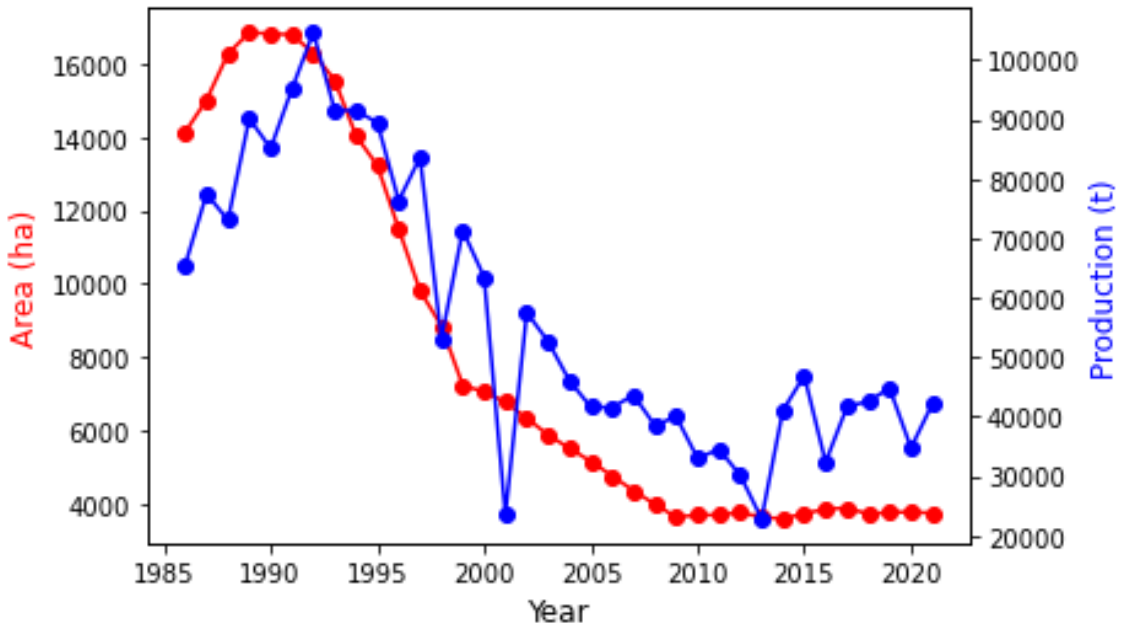


Fig. 23: Peach PT - Area Vs. Production.

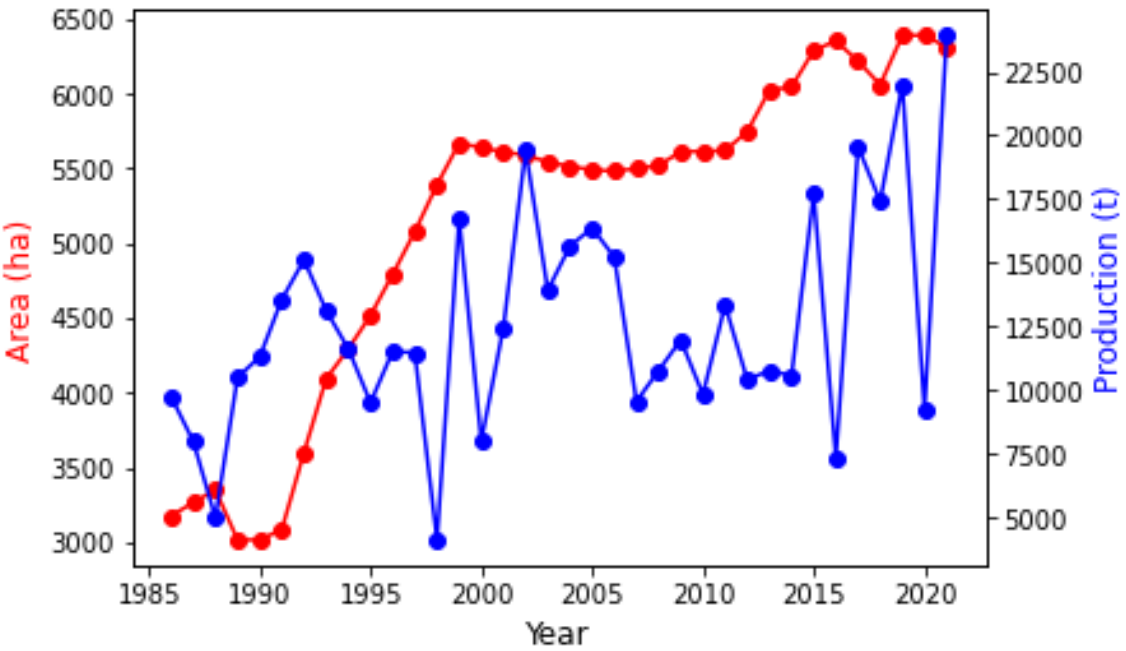


Fig. 24: Cherry PT - Area Vs. Production.

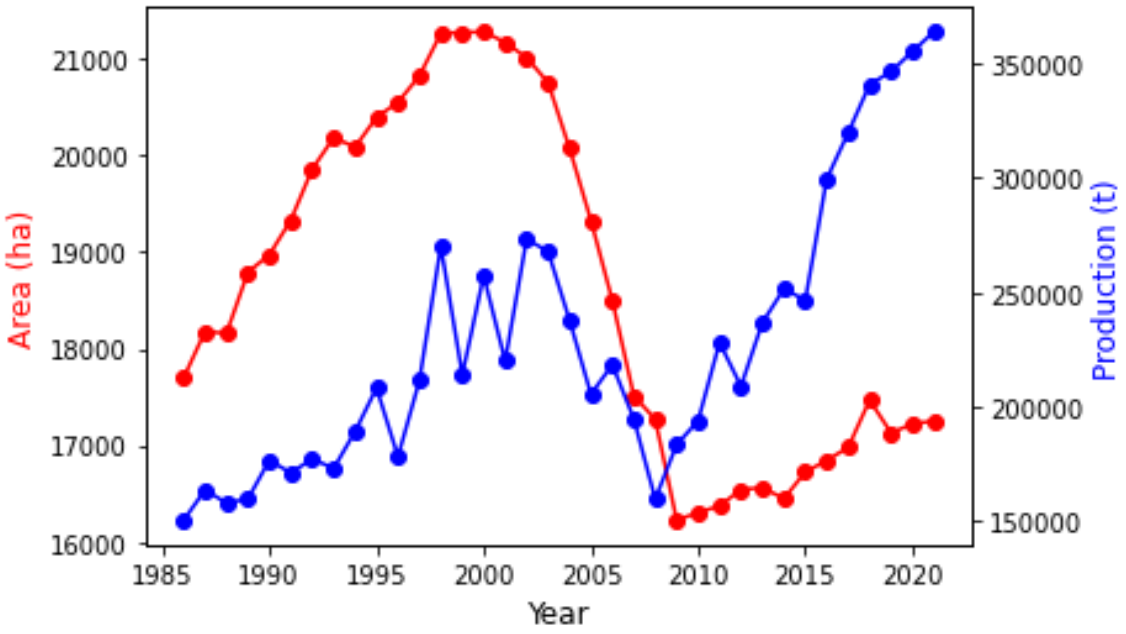


Fig. 25: Orange PT - Area Vs. Production.

Algorithm	Mean Absolute Squared Error
Linear Regression	0.18963133231672738
Decision Tree Regression	0.1378541671085377
Lasso Regression	0.16796664030356373
Ridge Regression	0.16010505224243088

Fig. 26: Banana production in Madeira - Comparison of the models' MASE.

Algorithm	Mean Absolute Squared Error
Linear Regression	0.1519435039615321
Decision Tree Regression	0.177204325975611
Lasso Regression	0.2017947230337752
Ridge Regression	0.19754365460296414

Fig. 27: Orange production in Algarve - Comparison of the models' MASE.

Algorithm	Mean Absolute Squared Error
Linear Regression	0.10776421230738617
Decision Tree Regression	0.10220344471867004
Lasso Regression	0.11480063739339989
Ridge Regression	0.11934036007204116

Fig. 28: Grape production in the North region - Comparison of the models' MASE.

Algorithm	Mean Absolute Squared Error
Linear Regression	0.3819626387092525
Decision Tree Regression	0.5076170401236678
Lasso Regression	0.8962930080243126
Ridge Regression	0.9338019327318211

Fig. 29: Olive production in Alentejo - Comparison of the models' MASE.

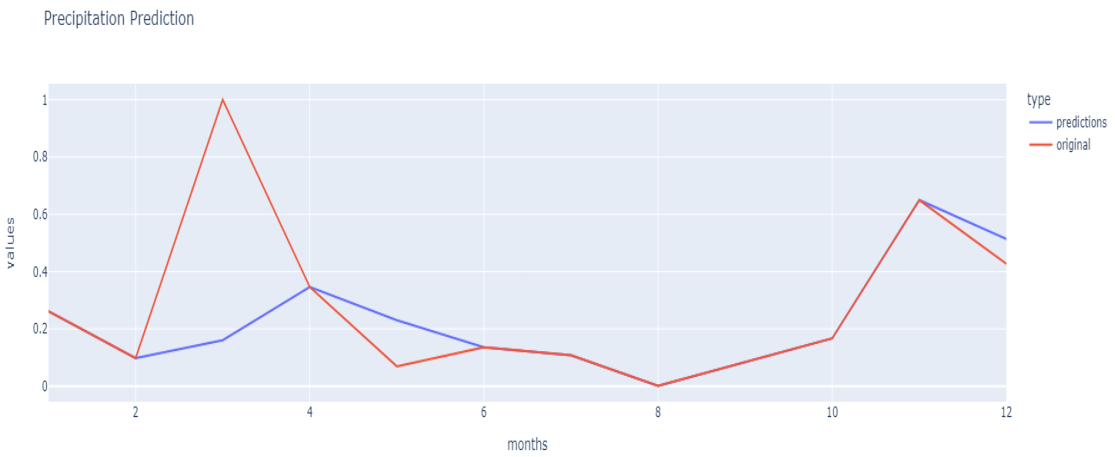


Fig. 30: Precipitation of 2018 - Prediction Vs. Real Values