Predial and ICA Tax Forecasting

Rionegro Mayor's Office

DS4A Colombia 5th Edition - Team 58

Carlos Cardona, David Cortés, José Parra, Julián Egas,

Laura Ocampo, Santiago Tellez, Vatsaid Molano

Triibu

Visit the Triibu app here: https://www.rionegrodatascience.com

Contents

1. Introduction

In Colombia, for most people, municipalities are the closest contact to any form of government, as they are one of the lowest levels of government in the country's political and administrative division. Colombian municipalities play a prominent role in the provision of public services including such as trash collection, water, sewage, health, roads construction and maintenance, and education, among others. Municipalities are also in charge of coordinating and regulating land use and, along with the police, promoting safety. Moreover, municipal governments usually act as implementer or connections between citizens and higher levels of governments (i.e. departmental and national).

Along with all of these responsibilities, the Colombian constitution grants municipalities with the ability to collect certain types of taxes and other fees. Among those, the most relevant are the property tax and the business tax (i.e. *Impuesto de Industria y Comercio* in Spanish. According to the *Comisión de Estudio del Sistema Tributario Terrotorial* both taxes represent around 69% of total tax collection in Colombian municipalities [6] (See Figure 1.1)

In this project we worked with The Mayor's Office of Rionegro, Antioquia. Rionegro is a municipality located in the department of Antioquia. It has 142,955 inhabitants, an area of 198 Km^2 and an annual budget of around $ 790.087 millions [10]. Rionegro is located next to Medellín (See figure 1.2, Colombia's second largest city, and hosts Medellín's international airport. Its main economic activity is the industrial sector, and it hosts a number of traditional Colombian manufacturers.

Specifically, we worked with Rionegro's Secretary of Finance to help them forecast revenues for its main sources of revenue: property and business. In 2019, those taxes amounted to 68% of the municipalities' tax revenues and 34% of its total revenue, which includes non-tax revenue and transfers from the National Government [12].

It is relevant to note that Rionegro is a municipality which has a healthy fiscal performance. In the most recent estimation of the Fiscal Performance Index by *Departamento Nacional de Planeación* (National Planning Department), which measures municipalities' performance related to revenue generation and expenditures, it ranked first among all of Antioquia's municipalities, with an index of 80,80 (out of 100) [1]. Moreover, historically, it has not been reliant on transfers from

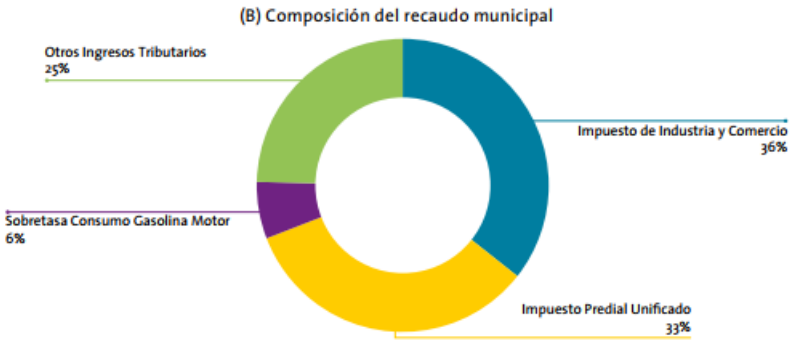


Figure 1.1: Composition of municipal tax collection. Source: Informe Final. Comisión de Estudio del Sistema Tributario Territorial. 2020. Accessed from: https://economia.uniandes.edu.co/sites/default/files/webproyectos/comisionstt/CESTT-Informe-web.pdf on September 3, 2021

Figure 1.2: Rionegro's location. Source: Accessed from: `https://es.wikipedia.org/wiki/Rionegro_(Antioquia)` on September 3, 2021

the National Government. As figure 1.3 below shows, in most of the years between 2012 and 2016, around 60% of the municipality's investments were funded by the municipality's own resources.

In Colombia, important efforts have been made to strengthen the income of municipal administrations, as an essential element of local autonomy, since it enables the fulfillment of the competences which they are in charge of. The amount of money that comes to the municipalities for taxes is limited, so the efficient administration of all those tributes is quite important and allows municipalities to manage and redirect resources towards the planned, sustainable and equitable development of the municipality.

Although, Rionegro has had an excellent fiscal performance in recent years, precise predictions about future revenue will allow authorities to better plan for service provision for its citizens. Forecasts for two of the municipality's most important sources of revenue allows it to reduce uncertainty related to future income to better plan the policies, programs, and services in education, health, infrastructure and transportation, among others, to be implemented in the future, which

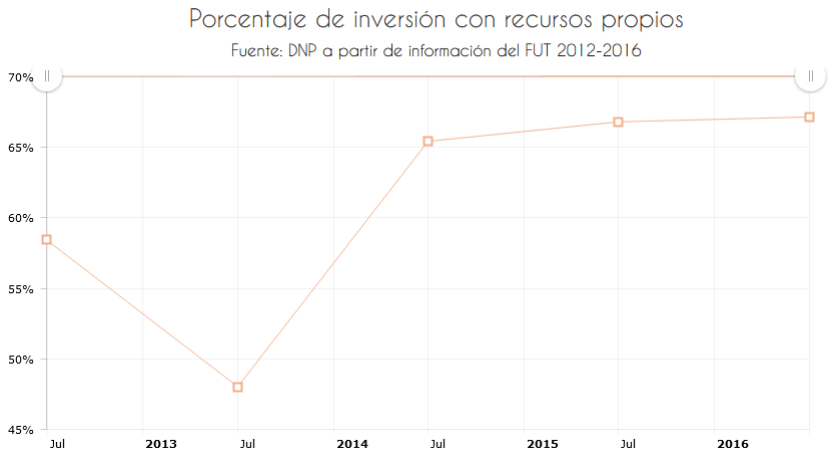


Figure 1.3: Percentage of investments made with municipality's own resources. Source: Terridata.Accessed from: `https://terridata.dnp.gov.co/` on September 3, 2021

ultimately benefits all of Rionegro's citizens. The extent to which those policies can be carried out is conditional on its budget, and therefore an accurate estimation of future revenue is essential for expenditure prioritization. Moreover, the methods that result as a solution of this problem can be generalized and applied to other municipalities so that the same issues can be addressed across the country.

The goal of this report is to describe the data analysis and models that we estimated to forecast future payments of the property and business taxes in Rionegro. It also describes the application we built for the use of the municipality to explore the results of our estimations.

Summary: Problem Statement & Background. In this section, we have provided the context and the background within which the problem we are dealing with is framed. Some references to the consulted sources are cited within the section as well as along the remaining of the report and in the Bibliographic section.

2. Data Description

2.1 Data description

The main source of our datasets is Alcaldía de Rionegro (Rionegro Mayor's office) [1].
For this project, the organization initially provided us with 4 types of datasets for each of the years from 2016 to 2020. However, we soon realized that the among of data was not enough to provide a sensitive statistical analysis. We get in contact with the entity to request more data, if possible, and after some delay in communication they were able to provide datasets for the time period between years 2000 to 2020. Table 1 below provides a brief explanation of the datasets:

Dataset type	Description	No. of variables	No. of observations	Unit of analysis
Property tax billing/ Facturación predial	Includes information for all properties in the municipality, including, the tax rate, the socioeconomic stratum for the property, its neighborhood, the lot and construction areas, and their valuations, the owner's name and its ownership share (in case there is more than one owner), and the tax expected to be collect (billed). There does not seem to be missing values (at least formally declared in the data).	22 for 2020 and 2017, and 21 for remaining years (Variable matricula missing for those years).	Ranges from 76859 in 2000 to 99262 in 2021.	Owner-property.

[1] We have complemented the datasets provided by the Mayor's Office, with the following data: Data of monthly arrivals to Rionegro's airport, gathered from the Aerocivil database [2], data of monthly unemployment rate in the department of Antioquia, gathered from the Colombian office of statistics, Dane [7], as well as yearly commercial license registrations, gathered from the commercial chamber of Antioquia [4]

Property tax payments/ Pagos predial	Includes information about the actual revenue collected for this tax, including the billed value, date of last payment, and the owners' and property's ID.	20 for all years	Total is 1443754. Maximum is 109347 in 2016, and minimum is 254 in 1999.	Payment.
Business tax declaration / Declara. industria y comercio	Includes information about businesses' revenues which is used to estimate the value of the tax for each period, including the business name and identification, whether the business is a company or an individual, ID and description of the economic activity, the business's revenue for the period, the tax rate and value, and an indication of whether there is a need to pay for other concepts and their value.	11 for all years	Ranges from 6351 in 2016 to 7509 in 2020.	Business (ID)
Business tax payments / Pagos industria y comercio	Includes information about the actual payments, including bill number, billed value, business ID and name, and date of last payment.	9 for all years	Ranges from 6037 in 2000 to 37523 in 2020.	Bill number.

2.1.1 Data sets links

There is no relationship between the property taxes and the business taxes datasets. Therefore, we are going to treat them independently and for the most part we will perform independent and unrelated analysis between. For the property tax data sets, there is a clear relationship. The billing dataset includes the bills issued by the mayoralty. The payments' dataset includes the payments made by individuals and companies towards those bills. In this case, the primary key(s) to join those two datasets are the individual or business identification. In the billings data set the variable is called "**nro_doc_id_catastral**", while in the payments' dataset the variable's name is "**CC O NIT**. In both cases, we have the names of the individual/business, which we will use to verify and improve the datasets merging process. The same occurs for the business tax. In that case, the primary keys are the business identification. The figure 2.1 shows the attributes of the raw data, and the current relationships between them.

In both cases, property and business, we have multiple years. As of now, we consider the best strategy to aggregate the data is to append all years together. Doing so will guarantee that the data is tidy, and allows us to group by year to perform some analysis in case we need it.

2.1.2 Planed data infrastructure

Planned data infrastructure: The database design will be based on the following E-R Model. The primary source of information include the same initial group of datasets, with 4 tables, property taxes billings and payments (**'tb_pagos_predial','tb_factura_predial'**), and business

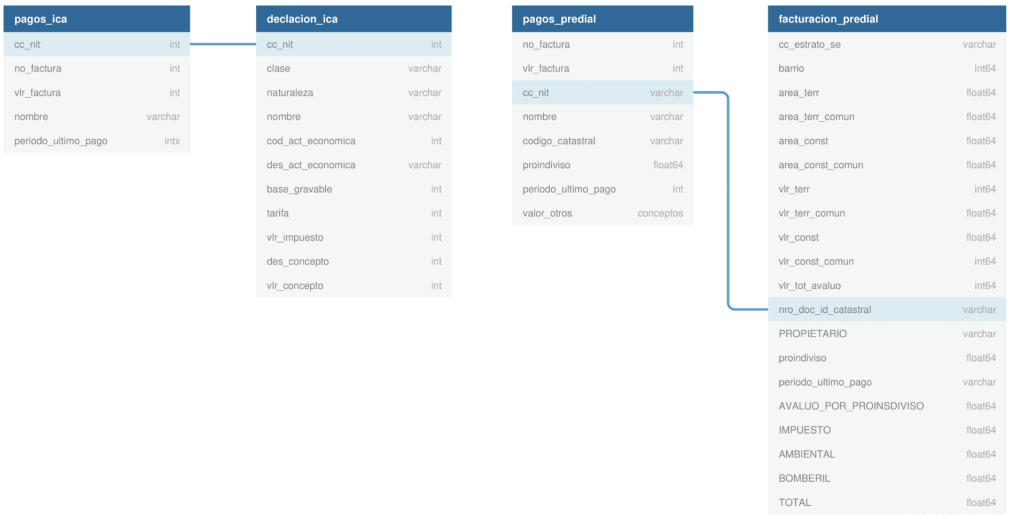


Figure 2.1: Data sets links

taxes billings and payments (**'tb_pagos_ica','tb_declara_ica'**). The two groups of datasets will be related between each other through the table **'dim_table'** with the field **'last payment date'** in the tables **'tb_pagos_ica'** and **'tb_pagos_predial'**. The primary key of the data table is a column with the date in format "YYYYMM" for the years between 2000 and 2020. In order to normalize the database and avoid duplicated information, for each group of taxes tables, a **'dim_tax_cc_nit'** table will be created with the unique identification number ('CC' or 'NIT') for a person or company, its name, and additional related features. Finally, having in mind a clear analysis of the information and a proper visualization on the dashboard, two new tables will be added. For the property tax, the table **'dim_barrio'** has the name of the neighborhood associated with the code provided in the original data **'cod_barrio'** and connected to the property table billings. Similarly, for the business tax, **'dim_act_economica'** relates the business activity code **'cod_act_economica'** with its description to easily classify the different kind of business existing in the municipality. The data structure is shown in figure 2.2.

2.2 Data wrangling

2.2.1 Property taxes data sets

Property taxes data are divided in two major categories: Property tax billing "FACTURACION PREDIAL_{Year}", and Property tax payments "PAGOS PREDIAL {Year}" where year goes from 1999 to 2020.

Data cleaning

Property tax billing or "FACTURACION PREDIAL_{Year}": On the features '**VLR_TERR', 'VLR_TERR_COMUN', 'VLR_CONST', 'VLR_CONST_COMUN'. 'VLR_TOT_AVALUO', 'AVALUO POR PROINDIVISO', 'IMPUESTO', '% AMBIENTAL', '% BOMBERIL'** and **'TOTAL'** undesired dollar signs'$', commas',' and hyphens '-' were removed. These are the main numerical variables, so they were converted to float. Missing values were defined as NaN. From all the files, two records were ignored since the value in the main numerical variables had an undetermined string.

Figure 2.2: Data structure

Property tax payments or "PAGOS PREDIAL {Year}": An empty column was deleted. Two of the column names '**VALOR FACTURA'** and'**VALOR OTROS CONCEPTOS'** have indentation at the beginning, so the columns were renamed, removing these spaces. '**VALOR FACTURA'** also contained non-numeric characters, indicating type of currency and hyphens for empty strings; It was converted to integer, the empty strings placeholders were replaced with zero and the periods were removed.

'PROINDIVISO' was given with comma as decimal separator, so they were replaced with periods. Columns **'NOMBRE', 'CODIGO CATASTRAL', 'PROINDIVISO'** and **'CC O NIT'** were converted to string variables.

Missing values

Concerning the Property tax billing data from 2016 to 2020, there are 1443754records. There are missing values in the following columns:

- 18613 records without data of "cc_estrato_se". This is a categorical variable that indicates the socioeconomic stratum of the property. For instance, we assign a new category to these records.
- 79275 records without data of "per_ult_pag". 76860 of these records correspond to 2018's database that does not provide this variable. Probably, we should drop it due to the missing values (pending a meeting with the Municipality).
- There are around 15 records suspicious of being corrupt that have missing information in the variables: "area_terr", "vlr_terr_comun", "vlr_const", "proindiviso", "AVALUO POR PROINDIVISO", "IMPUESTO", "% AMBIENAL", "% BOMBERIL", "TOTAL". We are evaluating the option of removing them.

Concerning the Property tax payments data from 1999 to 2020, there are 1443754 records. In this dataset, there are just 6 records with no data of "ult_pago" (last payment). The taxpayer ID has missing data for 10517 records, and the payment period is missing for 92241.

2.2.2 Business taxes data sets

Business taxes data are divided in two major categories: Business tax declaration "DECLARA AÑO GRAVABLE_{Year}", and Business tax payments "PAGOS ICA {Year}" where year goes from 2016 to 2020.

Data Cleaning

Business tax declaration or "DECLARA AÑO GRAVABLE_{Year}": Variables contained in the features '**BASE GRAVABLE'**, **'VALOR IMPUESTO',** and **'VALOR CONCEPTO,** are numerical variables that were given as string type, and we have converted them to float. Also, they had undesired dollar signs'$' that were removed, as well as numbers separated by commas. Finally, those also contain missing values in the form of '$-', so we decided to put these values as NaN.

Business tax payments or "PAGOS ICA {Year}". '**VALOR FACTURA'** also contained non-numeric characters, indicating the type of currency and a placeholder for empty strings; It was converted to float and the empty strings placeholders were replaced by NaN values. **'LAST PAYMENT'** was given as a String Data type, but it should be a DATE data type, so we converted it to date type. It also has a character placeholder to indicate possible missing values which we replaced with NaN values. Columns **'NUMERO DE LA FACTURA'** and **'CEDULA O NIT'** were converted to integers.

All features names in all the previous datasets contained white spaces that were removed.

Data transformation

The table "**DECLARA AÑO GRAVABLE_{Year}** " contained two types of categorical variables: **'CODIGO ACTIVIDAD ECONOMICA'** that reflects the economic sector of the business according to the International Standard Industrial Classification (CIIU) and **'CONCEPTO'** that gives information on the type of tax that was paid. For the first column, a reduced number of working categories for exploration and modeling was desirable, since the original variable had 410 different values. Using the structure of the CIIU all these values were aggregated to 20 categories representing a more general classification of the economic categories of these businesses. For the second column, '**CONCEPTO'**, only three categories were present, so no additional transformation was needed.

Missing values

Business tax declaration or "DECLARA AÑO GRAVABLE_{Year}" contains 35045 records. The column 'BASE GRAVABLE' contains 3216 missing values, while the column 'VALOR IMPUESTO' contains 8721 missing values, and 'CONCEPTO' and 'VALOR CONCEPTO' contain 9762 and 14453 missing values, respectively.

Business tax payments or "PAGOS ICA {Year}" contains 515135 records. There are 3 missing values in the column ULTIMO PAGO.

Summary: Data Wrangling & Cleaning. In this section we have presented the collection of data, additional data gathering, description of datasets variables, cleaning and wrangling, as well as provided de links between tables and the data structure we are going to use. Some additional prepossessing on the data will be done during the modelling in later sections.

3. Exploratory Data Analysis

Our main task is to predict future payments for two sources of revenue of Rionegro, i.e. business and property taxes. Therefore, our exploratory analysis focuses on understanding the evolution of tax payments related to those sources overtime. For the both taxes, in addition to information regarding the payments, we also had a data set that included the bills or tax declarations issued by the municipality along the characteristics of the properties. Although, we were not able to properly link both data sets to identify exactly what payments were made for what bill and property, o declaration, we still used the information on properties for our exploratory analysis and to show some insights in our application.

3.1 Business tax

Figure 3.1 below shows the time series of business tax payments for the years 2001 to 2020. Total tax payments have increased gradually in the last 20 years, with a steeper growth in starting around 2013. Moreover, the graph shows a peak occurring in the last month of every year, which is consistent with the rules set out by the municipality to incentivize early payments, a trend that is clearer in the years since 2018.

Beyond looking at total payments, it is worth looking at the median tax payment by month. Figure 3.2 below shows a slight increase from 2000 to 2015, and then a more noticeable increase in the years after that. The graph also shows some peaks in April, May and December of some of the latest years. Moreover, there is a sharp decrease in the median payments for the last two months available.,

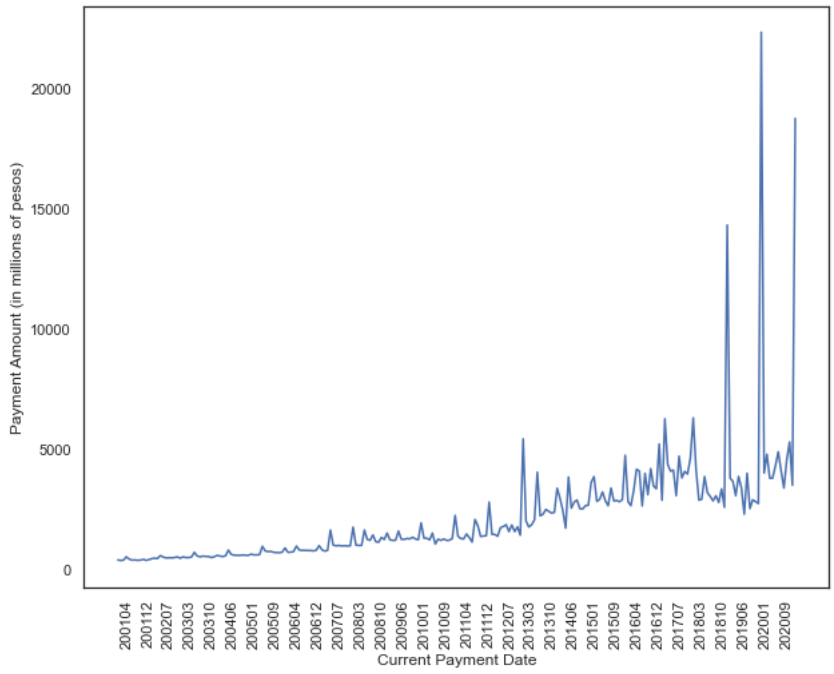


Figure 3.1: Business - tax total payments per month (2001 - 2020)

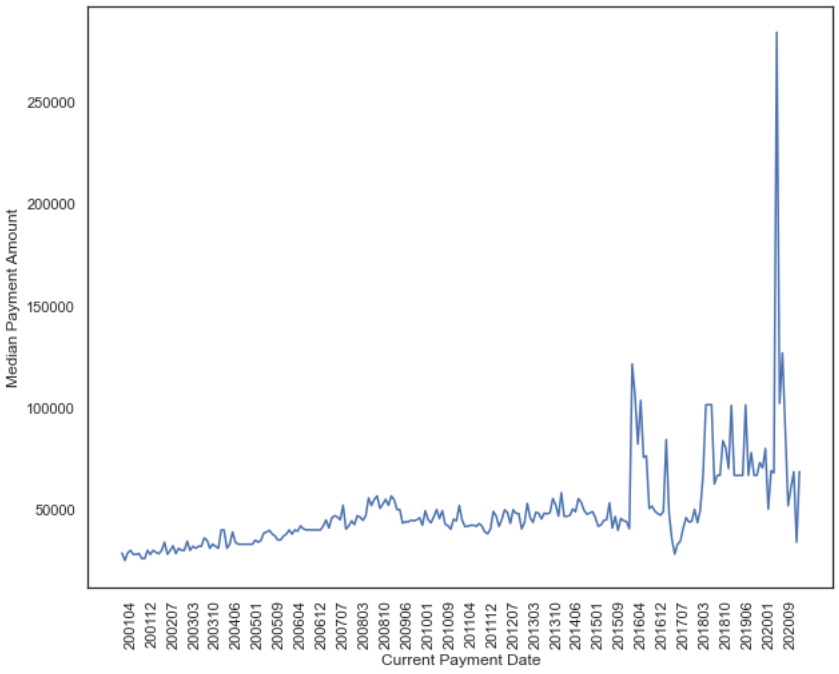


Figure 3.2: Business - median payment per year (2001 - 2020)

3.2 Property tax

In the case of the property tax, figure 3.3 below shows two clear patterns. First, there is an upward trend of property tax payments in the last 20 years, peaking in the fourth quarter of 2020. Second, there is a seasonal behavior, with the maximum total payments occurring in the last quarter of each year, when the municipality has set out discounts to promote early payments. We take into account

this seasonality in the the construction of our prediction models in later sections.

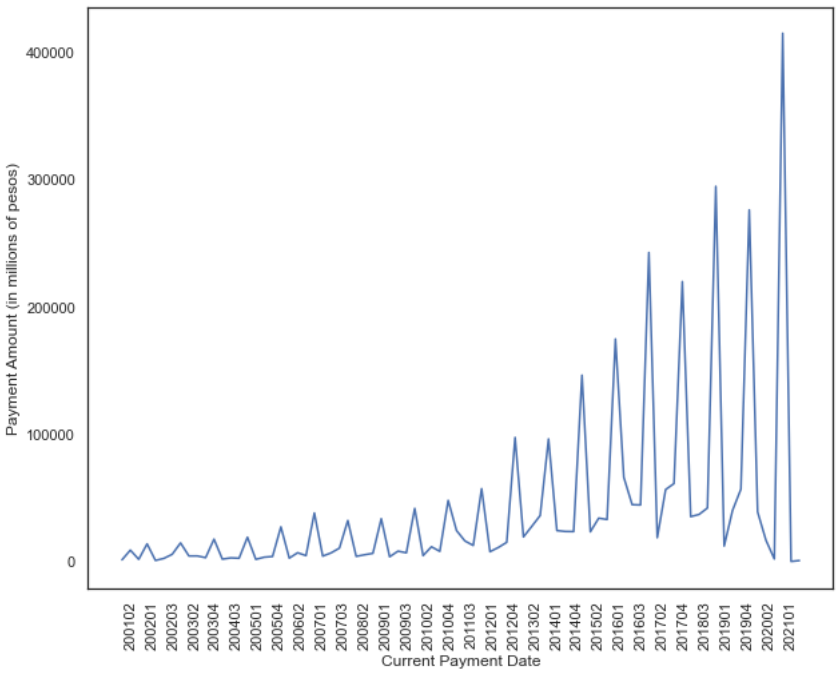


Figure 3.3: Property tax - Total payments per quarter (2001 - 2020)

The trend for the median payment per quarter (see figure 3.4) is very, very similar to that of the total payments. There is an upward trend, with a stark decrease in 2020, and the series shows clear signs of seasonality that we take into consideration for the predictive models that we present in later sections.

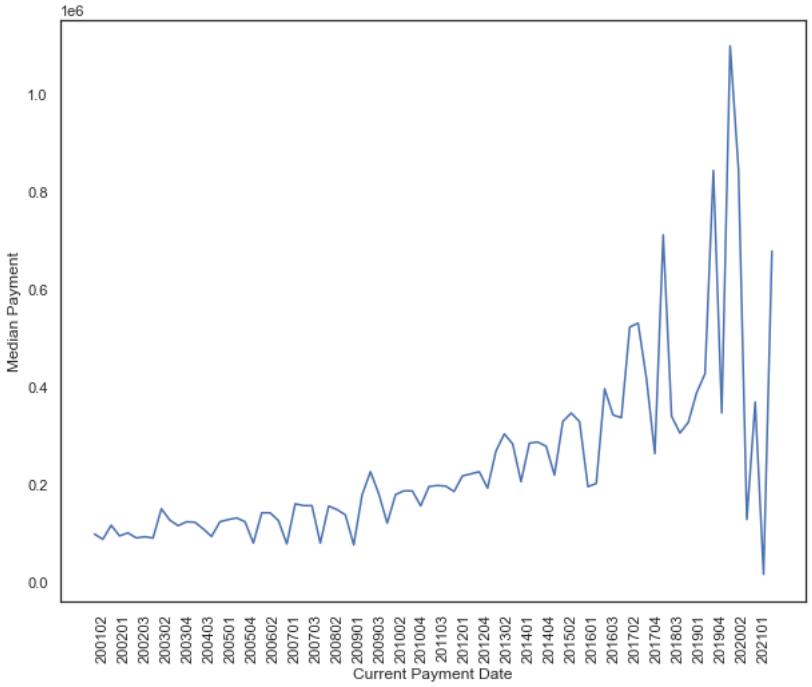


Figure 3.4: Property tax - median payment per quarter (2001 - 2020)

Something we need to keep in mind to predict revenues is that a good proportion of revenues might come from a small set of taxpayers. Figure 3.5 shows the top ten property taxpayers in the years included in the analysis:

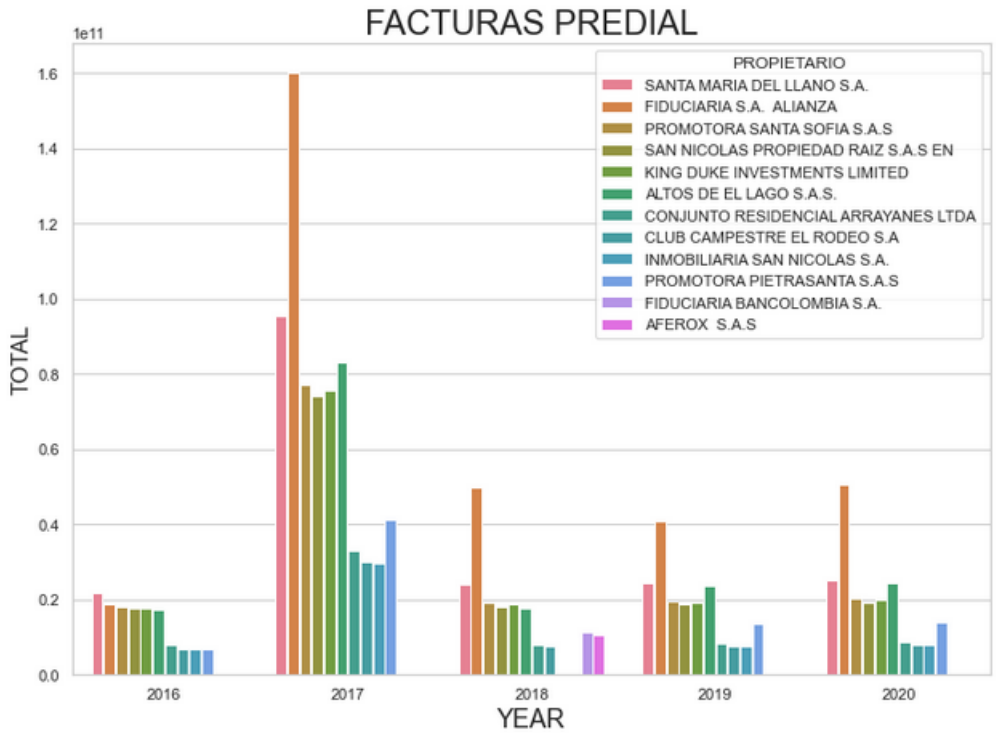


Figure 3.5: Property tax bills - top ten of taxpayers per year (2016 - 2020)

Moreover, figure 3.6 shows that the largest property tax bills are located in what the data identifies as stratum 0, a behavior that is reflected in the overview section of our application. Among the remaining strata, those properties classified as stratum 4 and 3 are the ones with a larger total billed.

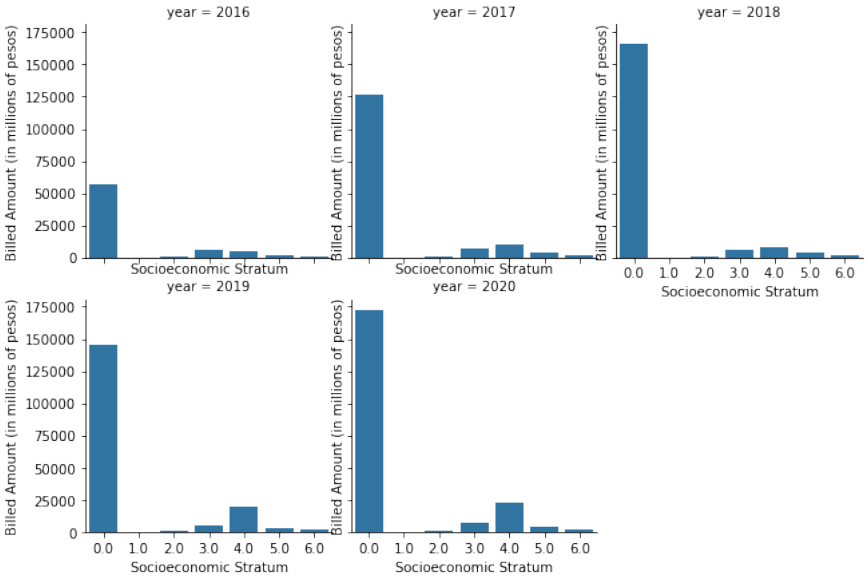


Figure 3.6: Property tax bills - Total per socio-economic stratum (2016 - 2020)

Summary: Descriptive Analysis In this section we have presented the exploratory data analysis alongside with the corresponding descriptive analysis. We have provided informative visualizations that allow us to get important insight of the data, that will lately result useful during the modeling building.

4. Models

4.1 Linear regression with regularization

The idea is to generate a linear regression to predict annual property tax collection. For this purpose, the data taken into account were the total appraisal, land area and stratum, as follows: All $0, 0.1, \ldots, 0.9$ percentiles of land area and land value and the number of lots in strata 1 to 6 per year.

Having already the matrix X and the vector y representing the collection, it's possible to run the regression. However, the data must first be scaled, which was done with the function 'Standard Scaler'. After this, the data were split into test and training (20% and 80% correspondingly), and the regression was run with the latter. Nevertheless, since X contains more columns than rows (as there is only 20 years of data), the solution of the problem is exact, which means that the linear equation $Xz = y$ has solution and is the same z that reaches the optimal value in $Min|Xz - y|$ which will be zero.

For this reason, the number of columns is reduced with recursive feature elimination (RFE) and then linear regression is run. The mean squared error of the training and test data is also obtained to verify the model efficiency, however for the years of training the result has an error close to 0 (a score close to 1) but for the test years the error is very large, Training RMSE is 0.0000 and Test RMSE is 0.3602. And it was also evidenced from the model plot that the model is over-fitted (see Figure 4.1).

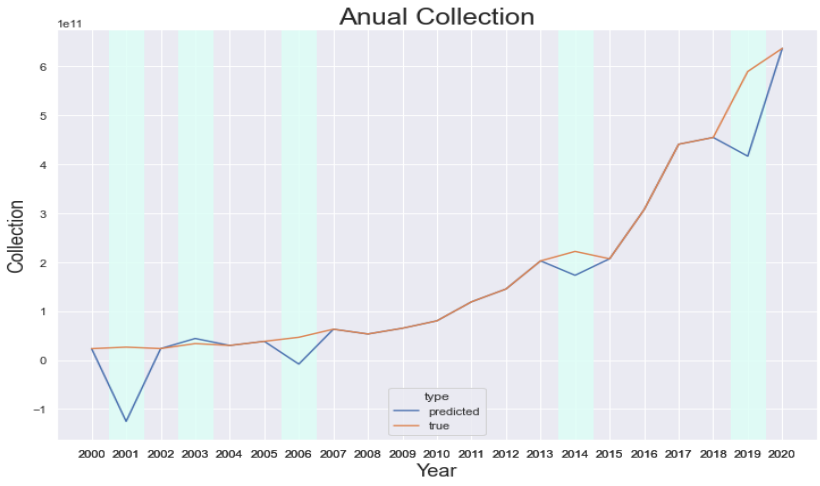


Figure 4.1: Annual property collection

To solve this, regularization (Ridge + Lasso) was used. The results improved a lot, and the following mean squared error was obtained: Training RMSE is 0.0146 and Test RMSE is 0.0172. Figure 4.2 shows these improvements.

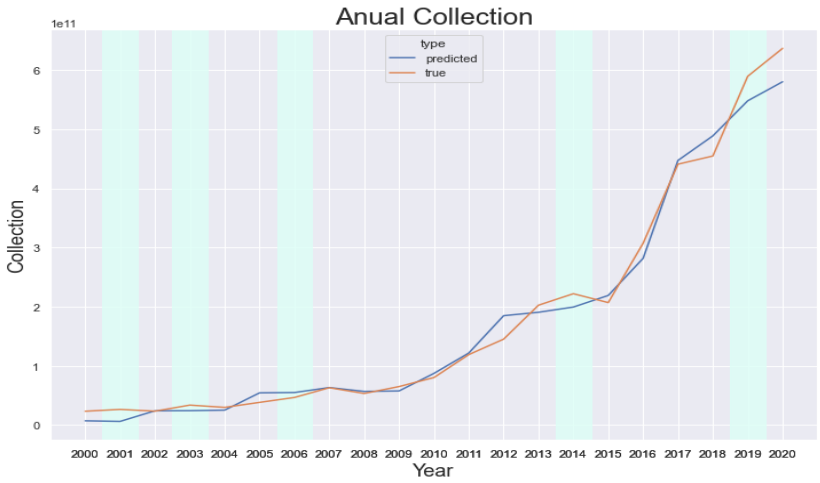
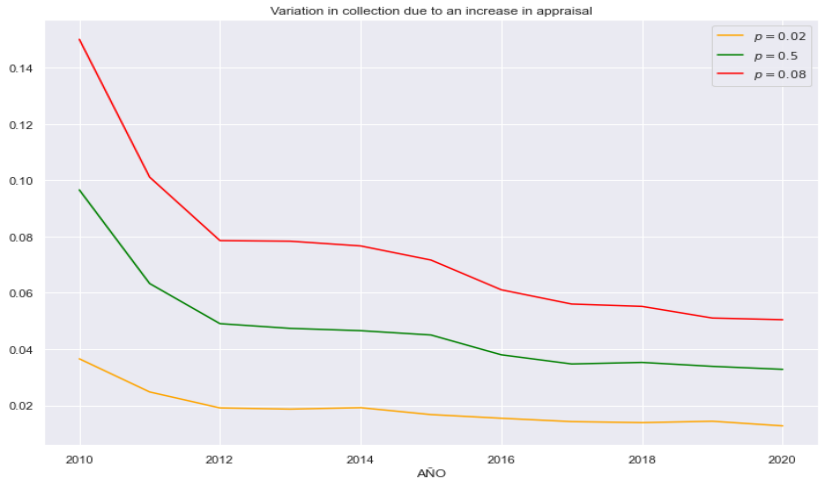


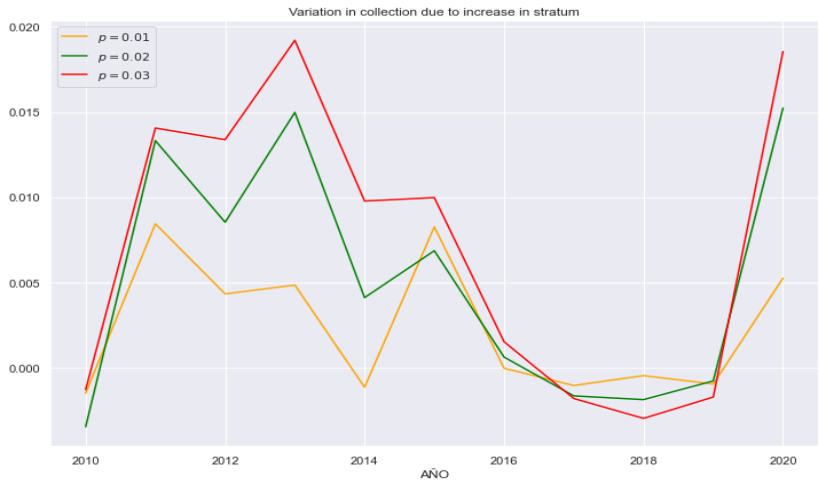
Figure 4.2: Annual property collection with regularization

The highlighted regions (in light green) are the years of the test set. It can be seen with the regularization that even though the error is greater than in the training ones, the results are adequate.

4.1.1 Sensitivity analysis

By having a model that predicts collection, it's now possible to alter some variables, and discover how this affect the result of total collection. The following images show the change in percentage of the total collection of the property, when the variables of appraisal, stratum and area are increased by a certain percentage.





It can be seen that collection is much more sensitive to changes in the stratum variable than to changes in the other two variables

4.2 Baseline models for time series

In order to set a baseline for comparison with more sophisticated models, we would like to present some very basic approaches that are easy to understand and interpret. The goal for these very basic models is to play as a sanity check for more complex and therefore less intuitive models (such as

Neuronal Networks and the likes). For the first part of the modeling, we are going to consider the time series nature of our datasets as the main component, and as such, the 'output' of these models will be primarily forecasting of some variables, as ' **tax_revenue**' and '**tax_declarations**'.

Simple Moving Average and Exponential Moving Average

One of the simplest and more commonly used methods in time series analysis is the Simple Moving Average (SMA) and Exponential Moving Average (EMA). They are used to smoothing out short term variability in order to highlight other time-like components such as trends, seasons and cycles. We can then use these methods to perform a rough forecasting that is solely driven by the 'average' behavior of the data.

SMA at time t, is computed simply by taking the mean of the T (window) consecutive observations previous to t. In mathematical terms:

$$SMA_t = \frac{x_t + x_{t-1} + x_{t-2} + \cdots + x_T}{T} \tag{4.1}$$

We start by exemplifying this method and the following only on the data sets corresponding to "Pagos Industria y Comercio", but of course the same analysis can be carried over the other datasets. In order to apply SMA to our data, we need to aggregate data by taking the mean value of all payments done per month. In other words, we have one data point ' **Valor_factura**' per month from 2000 to 2020. Applying (SMA) with a window T= 6 months gave us the following plot:

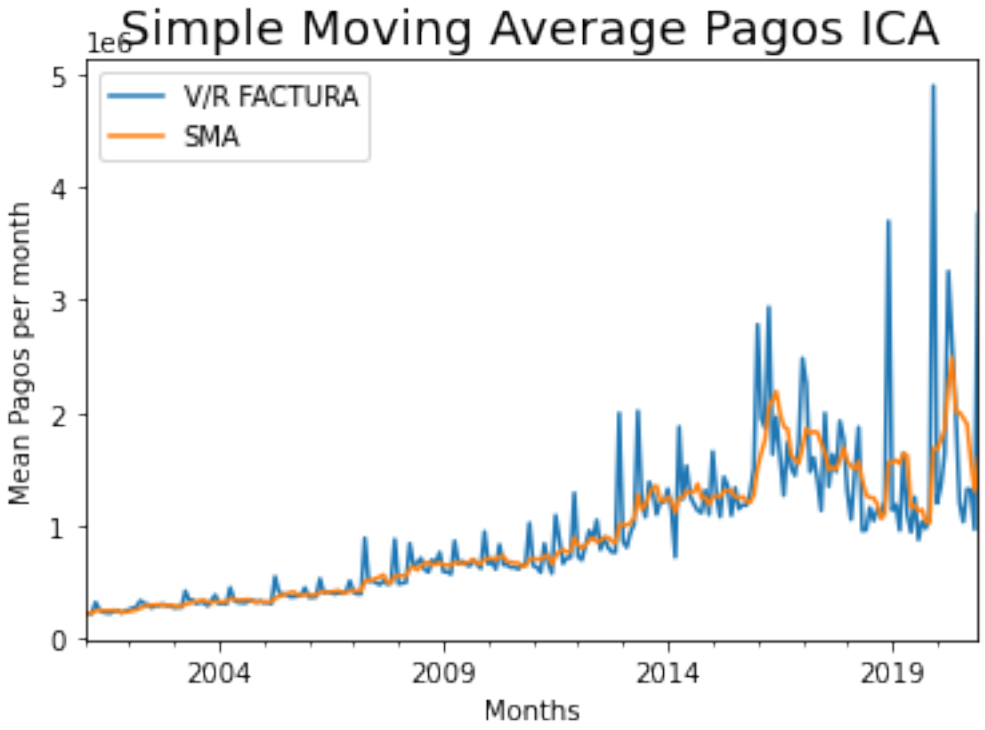


Figure 4.3: Simple Moving Average

This plot allows us to estimate that mean tax collection has been slightly increasing yearly since 2017. A more natural averaging for time series might be one in which we can give more weight to the most recent events than to the oldest occurrences. This is allowed by Exponential Smoothing (ES), where a weight alpha is given to the most recent observation and a weight α is given to the average over all previous observations. More precisely, we can define it in mathematical terms as a

recursion,

$$(ES)_{t+1} = \alpha x_t + (1-\alpha)(ES)_t \,. \tag{4.2}$$

We can easily apply this method in python by using the package 'ExponentialSmoothing' from the library 'statsmodels'[13]. This package allows us to play with several different options. One of the options is to ask for the package to choose some optimal value for alpha. Figure 4.4, we show the results for $\alpha = 0.2, 0.6$ and $\alpha = optimal$ which is computed by the 'ExponentialSmoothing' package,

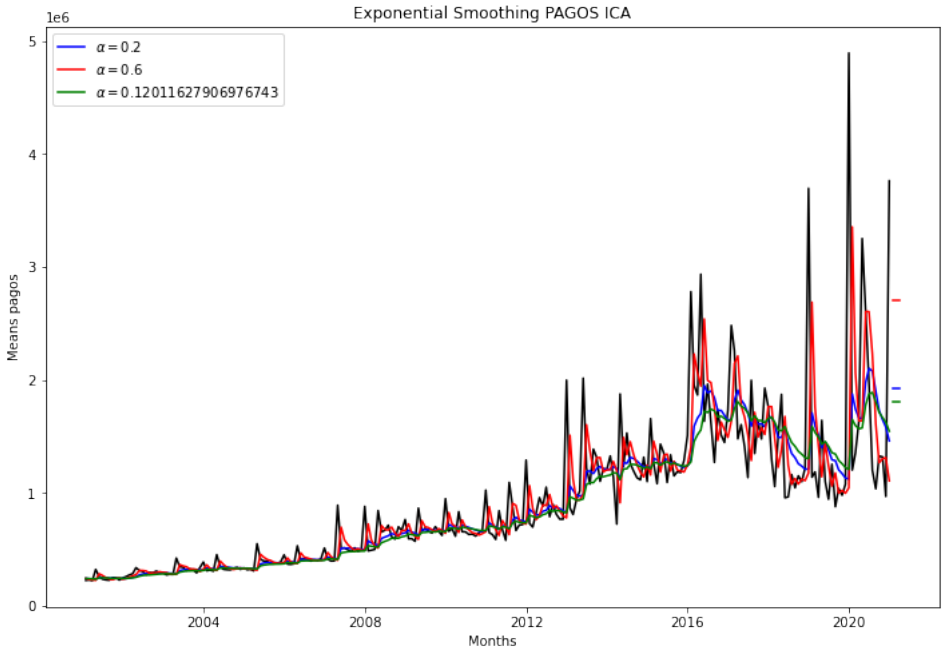


Figure 4.4: Exponential Smoothing

Now we can make predictions using the "ExponentialSmoothing" package very easily. The way to do it, follow pretty much the same lines as our best old friend Linear Regression (OLS). By splitting the data in training and test, we train the model and make a 36-month prediction that is to be compared with the test data. The results from the ES-forecast are summarized in figure 4.5. This plot shows the monthly means for 'Pagos' in black, the results of the model on the training data in blue, and the results on the unobserved data, or in other words, the predicted values, in red (SES stands for Simple Exponential Smoothing). The root square of the Mean Square Error equals, $(MSE)^{1/2} = 920771.02$, which means that the prediction is, on average, away by less than one million pesos from the real value. As we said at the beginning of this section, the goal of the simpler methods is to set a baseline for more sophisticated methods, in other words, any other more complex method should beat the MSE score of this simpler method, in order to be considered.

4.3 Deep and Recurrent Neural Networks

In order to provide a very brief explanation of Deep and Recurrent Neural Networks, we need to introduce a few concepts. Some terms used in Neural Networks literature, takes inspiration from Neurobiology: A neuron is loosely speaking a node that performs a non-linear transformation on the input data. Generally, this transformation is parametrized by some quantities known as weights and bias. The goal of a neuron is to adjust these parameters in order to approximate the non-linear

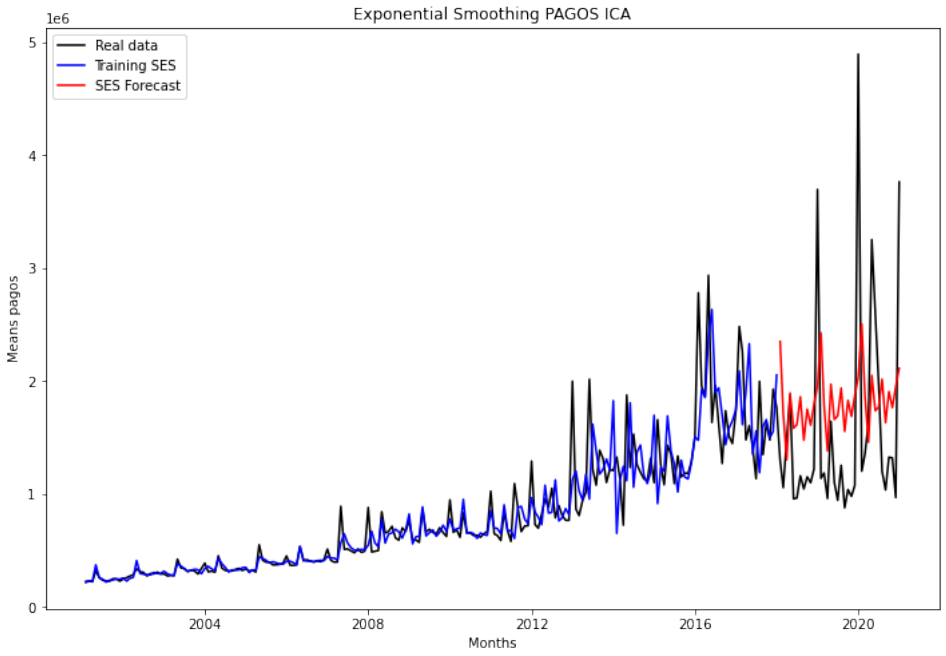


Figure 4.5: Exponential Smoothing Forecast 36 months

function as much as possible to the data provided, such as it reproduces the output expect on both, observed data (data we give the neuron to learn from) and unobserved data (data that we don't allow the neuron to look, but we use as a test of the learned function). A layer, is a collection of neurons, grouped in such a way that each of all the individual learned functions add up together to an 'averaged' function, that in practice will be better than a single function from a single neuron. Finally, a Deep Neural Networks (DNN for short) is, in few words, a collection of layers. Each layer takes as input the output of the previous layer and give its output to the next layer. For this reason, DNNs are known as feed forward networks.

A Recurrent Neural Network (RNN for short) has a little twist with respect to its sibling, the DNN. It can take the output of a deeper layer as the input of a previous layer. This simple exercise, provides the network with a sort of memory, and that reason make this architecture attractive to deal with time dependent data, such as time series [5]. We now go into explaining the details of the particular set up used in our modeling [1].

We start by aggregating data of tax payments by month, but this time we also keep a variable called "**Clase_comercial**" which is associated to the commercial nature of a business and is composed by five categories: **Financial, Commercial, Industrial, Services, Others** . Now our data consists of a data point per month per category from 2000 to 2020. We started by building a Dense Deep Neural Network (dense here only means that all the neurons are connected to both, all the neurons from the previous and next layer). The architecture we used for this DNN is very simple. We used a Sequential architecture from Keras with TensorFlow backend. We have an input layer connected to a layer of 32 neurons with 2% dropout to fight overfitting. This layer is then connected to the output layer, which only consists of one neuron and stores the value of the single variable "**Valor_factura**" per output, which corresponds to the tax revenue in a given month in the future

[1] A couple of papers we read in order to get familiar with what has been done in tax forecasting using neural networks are [3, 16]

(in training we often choose the future to be one year ahead, see below for more details.).

The Recurrent Neural Network is very similar to the previous DNN. We have an Input layer connected to a Gated Recurrent Unit (GRU) layer of 32 neurons, with 0.2 dropout to fight overfitting. This layer is then connected to the output layer. The model is compiled using a 'RMSprop' optimizer, and we measure the loss with the Mean Absolute Error function (MAE). Both architectures (DNN and RNN) suffered from over-fitting even after increasing the drop-out. In order to fight it, we applied a kernel l_2 regulator to the hidden layer in both cases. [2]

In non-time-dependent data, train test randomization is easily implemented simply by training the model with different batches of the same size whose elements are chosen arbitrarily from some random distribution. In the case of time series data, the previous procedure is not possible due to the fact that the time order has to be preserved. In order to induce some type of randomization for the neural networks, we have chosen the following procedure: Pick a random point from the data set, and from that position make a time window consisting of "lookback" months in the past of that point (usually 4*12 months in training), and a "lookforward" months in the future of that point (usually 12*1 months in training). Feed the network with that smaller window of time series. Pick the random temporal position in the data set a number of "batchsize" times (usually 12*1 months in training). Train the network with all those subsets. In sum, the DNN and RNN will learn to predict "lookforward" months in the future by looking into "lookback" months in the past.

We display the loss function (MAE) for both, the DNN and RNN with respect to the training data as well as the test data for different epochs of training in figure 4.6 below (it is worth clarifying that the plot in figure 4.6 is for normalized data), In these plots we can see that the DNN does not

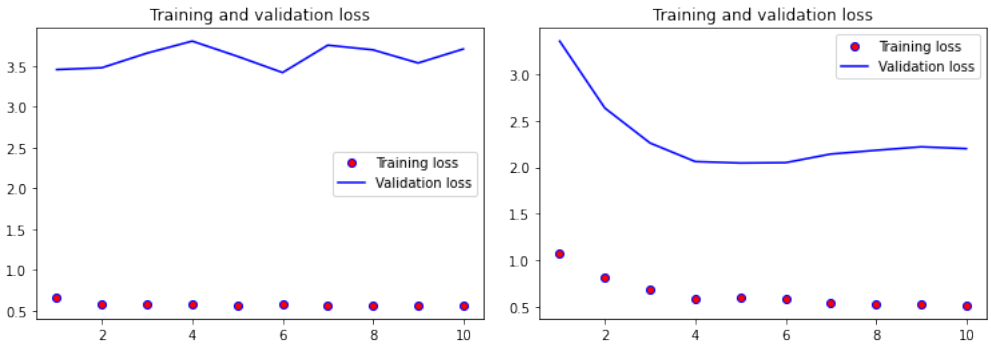


Figure 4.6: DNN and RNN loss per epoch

improve much with the number of epochs, and therefore we could have stopped the training at just the second epoch without loosing (or gaining) anything. The RNN improves at early epochs, but then it flattens after epoch number four, and therefore we can early stop the training for the RNN at around the fifth epoch.

[2] In practice, we tried several combinations of regularization, such bias and kernel regulators, and different combinations of l_1, l_2 and l_{12} measures. The one we have mentioned in the paragraph corresponds to the one that works the best.

4.4 (S)ARIMA

The classic univariate time series model in econometrics is the Autoregressive Integrated Moving Average (ARIMA) model. It is a forecasting model, but has inference capabilities. In this model, the time series y_t is assumed to be explained by its past values (autoregressive component or p) and contemporaneous and past values of the error term (moving average component or q); y_t is assumed to be weakly stationary, and a parameter for the order of integration of the series is incorporated into the model (d). Alternative specifications can include a multiplicative seasonal effect. A general ARIMA(p,d,q) model has the form:

$$y_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i y_{t-i} + \sum_{j=0}^{q} \beta_j \varepsilon_{t-j}$$

The ARIMA modelling process starts by identifying the stationarity and presence of seasonality of the series under analysis. This can be done either visually or using formal tests like the Augmented Dickey-Fuller (ADF) test or the Hylleberg-Engle-Granger-Yoo (HEGY) test. Sometimes both approaches complement each other. The next step is identifying the parameters q and p which is done by using the autocorrelation and partial autocorrelation function [9].

On the business tax, the ADF test indicated that the series was integrated of first order, which means that the first difference of the series was stationary. Using the HEGY test, no seasonal pattern was found. The value of the autoregressive and moving average parameters was found to be 2 and 12, and the final specification of the model was ARIMA(2,1,12). Figure 4.7 shows in dotted lines the forecast for 2016-2020 using data from 2001 to 2015, the confidence interval is represented in red.

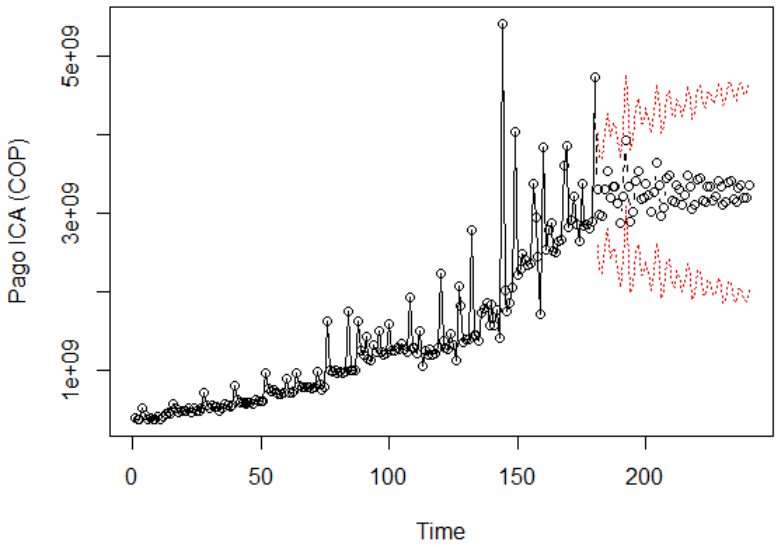


Figure 4.7: ARIMA forecast for business tax 2016-2020

For the property tax, the ADF test indicated again that the series was integrated of first order, and the HEGY test indicated a biannual seasonal pattern. The value of the autoregressive parameter

was found to be 2 and the moving average parameter was found to be 5, so the final specification of the model was SARIMA(2,1,5)(0,2,1), where the second parenthesis indicates the seasonal component. Figure 4.8 shows in dotted lines the forecast for 2016-2020 using data from 2001 to 2015, the confidence interval is represented in red.

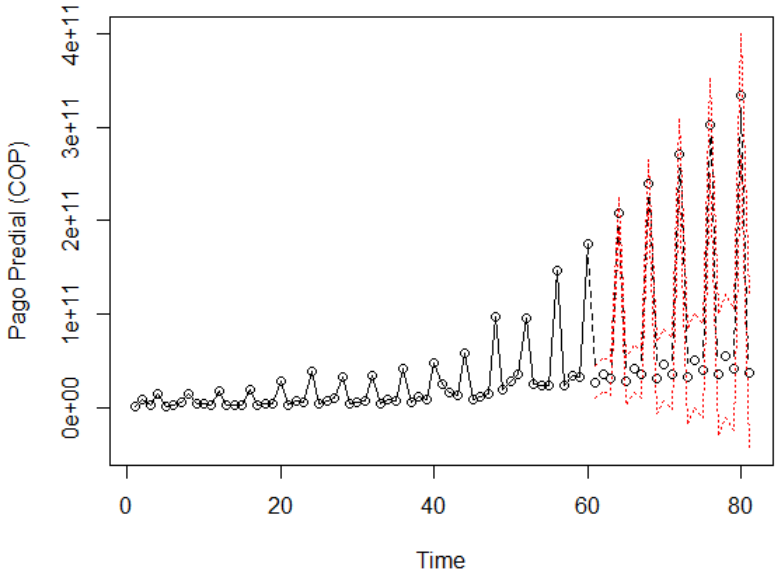


Figure 4.8: ARIMA forecast for property tax 2016-2020

Other models with the same specifications were trained on the full sample to obtain the 8-year forecast out of sample.

4.5 ARDL

The Autoregressive Distributed Lag (ARDL) model is the linear regression approach to multivariate time series analysis, where the conditional mean of one of the time series y_t is believed to be explained by its lags and other time series x_i and their lags. The general form of an ARDL model is [15]:

$$y_t = a_0 + \sum_{i=1}^{p} a_i y_{t-i} + \sum_{i=0}^{n}$$

The ARDL modelling process starts by identifying key time series that improve the explanatory power of the model. Then the integration order of all time series must be checked. Finally, the lag structure can be deduced using a general to specific methodology, where recurrent elimination of a large number of lags by their statistical significance reduces the structure to the most parsimonious one.

For the business tax time series, other variables such as the inflation rate in Medellín (the closest largest city), the unemployment in Medellín and passenger arrivals to the airport were incorporated,

as well as a lagged value for each one of them. For the property tax time series, a two period lag to capture seasonality was introduced in the model, and the same variables and their lags as the business tax were also introduced.

Table 4.1 shows a comparison of the performance of the models on the period 2016-2020.

Table 4.1: Model comparison

Database	**Business Tax**		**Property Tax**	
Model	ARIMA	ARDL	SARIMA	ARDL
MSE	3367.204	405557.8	27811.57	30727.55

Summary: Model and Statistical Analysis. In this section, we have built the models used to forecast tax revenue and do appraisal and revenue sensitivity analysis. We have made used of several machine learning models including classic econometric models, in particular we have used linear models and have built models based in methods not covered during the course, such Recurrent Neural Networks, ARIMA and SARIMA.

5. Front and Back End

5.1 Front End

Here we will provide a description of the different components of the front end, but the main goal of this section is to display the web application within the report and not to give too many details into the technicalities used in the construction.

The home page is displayed in figure 5.1. It gives a brief description of what is included in the rest of the web application. Also shows a map of Rionegro as well as its location withing the national territory. To help the user to gain even more familiarity with Rionegro city, we display some pictures of iconic places withing the city. This aim to create a connection feeling in the user.

The Overview page is presented in figure 5.2 and contains a brief descriptive data analysis of some of the more relevant features within the provided data sets, as well as some forecast analyses from the models.

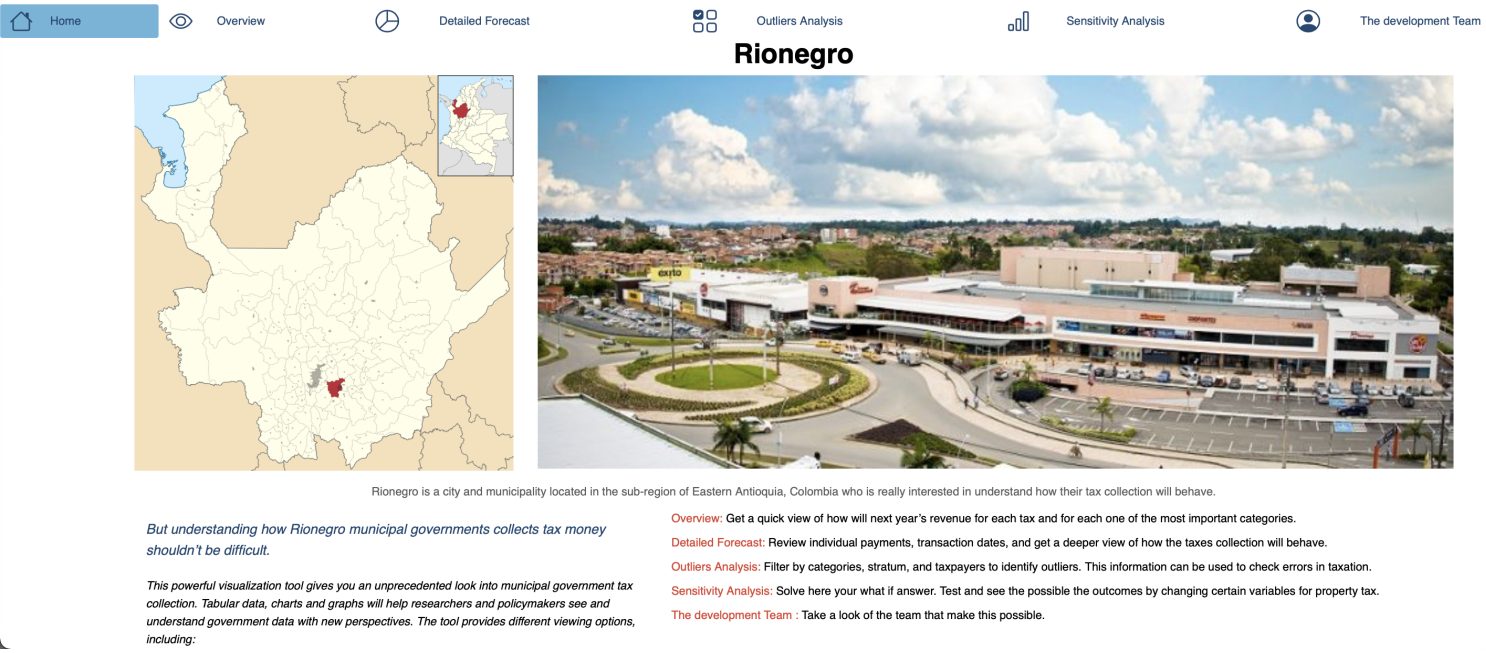


Figure 5.1: Home Page

Figure 5.2: Overview Page

The forecasting results from our predictive models is shown in the 'Predictive models' tab displayed in figure 5.3. Within it, we would see a series of interactive plots showing the results from the different models used in forecasting the tax revenue for the coming few years, as well as some comparisons of model performance with respect to the real data. On the top panel we have displayed the projection on tax revenue for the next eight years, for both, property and business taxes. In the bottom panel, we have presented the performance on predicting power of the different models when comparing with real data. Both panels show interactive plots that the user can play with.

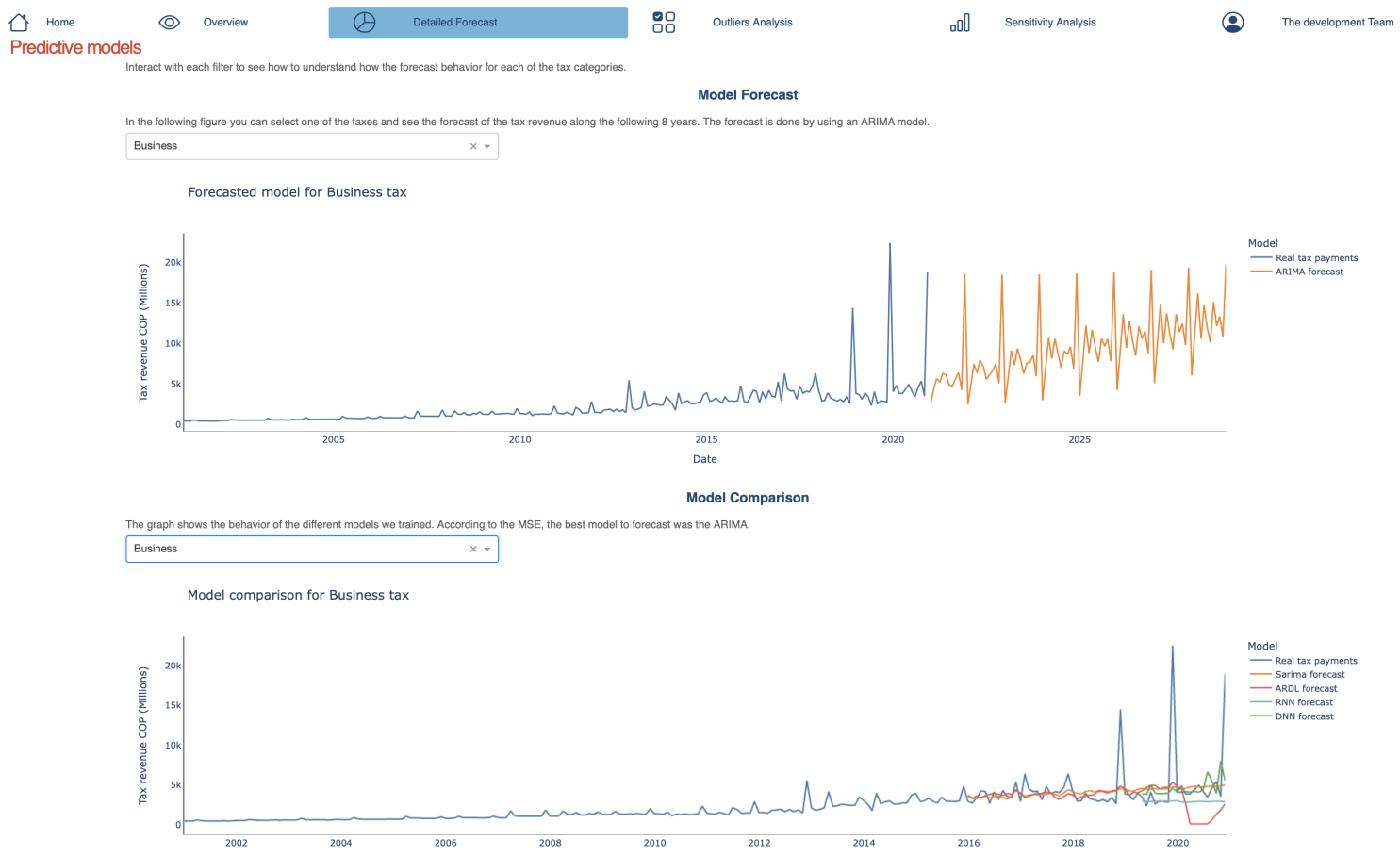


Figure 5.3: Predictive Models

The Outlier Analysis tab shown in figure 5.4 is intended to detect taxation errors or to redefine tax regulations, these points can be use as red flags for further study by Rionegro mayor's office.

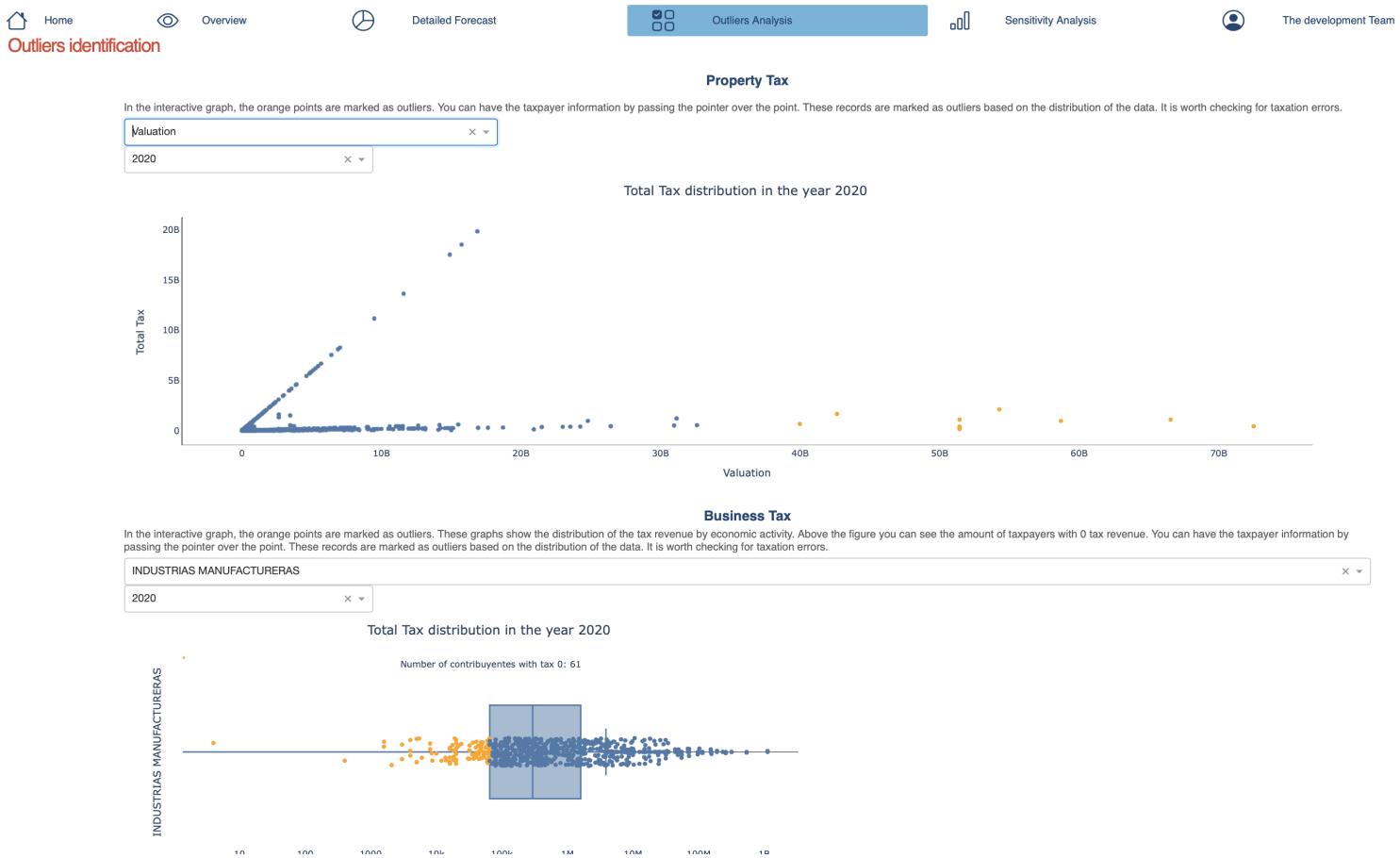


Figure 5.4: Outlier Analysis

The Sensitivity Analysis tab is dedicated to show the results from one of the models explained in detail in section 4.1.1. As seen in figure 5.5, it shows the sensitivity curves for tax revenue in terms of a given percentage variation in Valuation, Stratum and Property Area respectively.

Figure 5.5: Sensitivity Analysis

Last but not least, in the Development Team web page we present a picture along with a brief description of every one of the valuable members of the team, which have worked tirelessly to make this project come to light with the best of their abilities. We display the Development Team web page in figure 5.6.

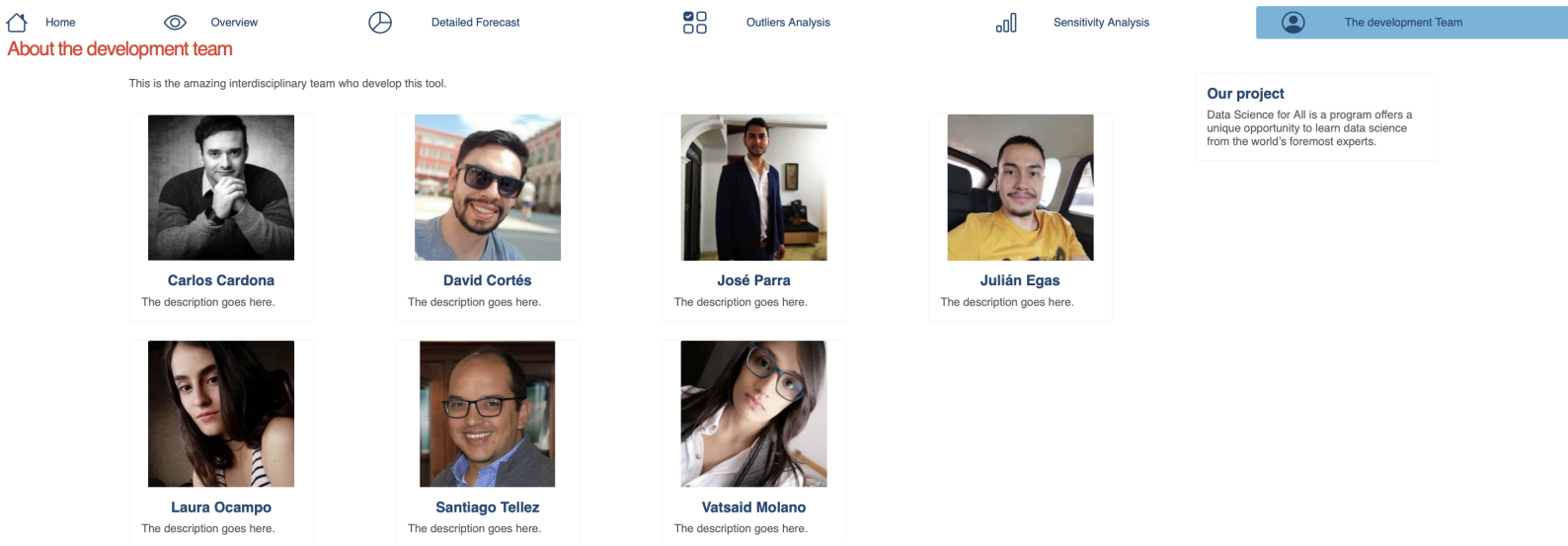


Figure 5.6: Development Team

5.2 Back End

The web application for the production environment has been deployed in DigitalOcean [8]. More specifically, in a droplet, which is a Linux virtual machine, whose main characteristics are summarized in figure 5.7.

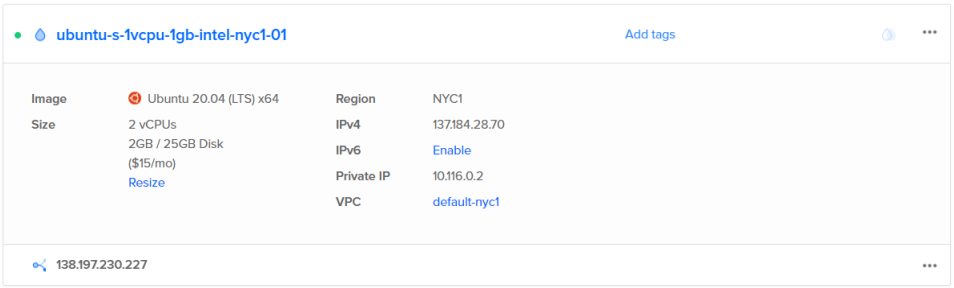


Figure 5.7: Linux server screenshot

In order to set a Linux server, some additional tools are required, such as uwsgi and nginx [11, 14]. Nginx is a high performance web server that can be used as a reverse proxy. uWSGI is a Python's web applications server, which bridges the web server with the Python application, or in other words, a client's request to the web server is taken by uwsgi, which in turns pass it on to the python application with the corresponding specifications. We illustrate the workaround of a server request in figure 5.8.

Bearing in mind the application deployment, the next step is the configuration. First, we connect to the virtual machine through an SSH tunnel, from which a folder called Rionegro has been created to store all the project files. Additionally, a virtual environment is created to carry on it all the necessary dependencies. The next step is to create an entrance point for our application, which instructs uWSGI how to interact with it, as seen in figure 5.9.

This is followed by the creating of a uWSGI configuration file, named rionegro.ini, and is presented in figure 5.10, which we break down here: The header calls uwsgi in order to take the configuration from it. Then the module is specified by referring to the previous file wsgi.py pointing

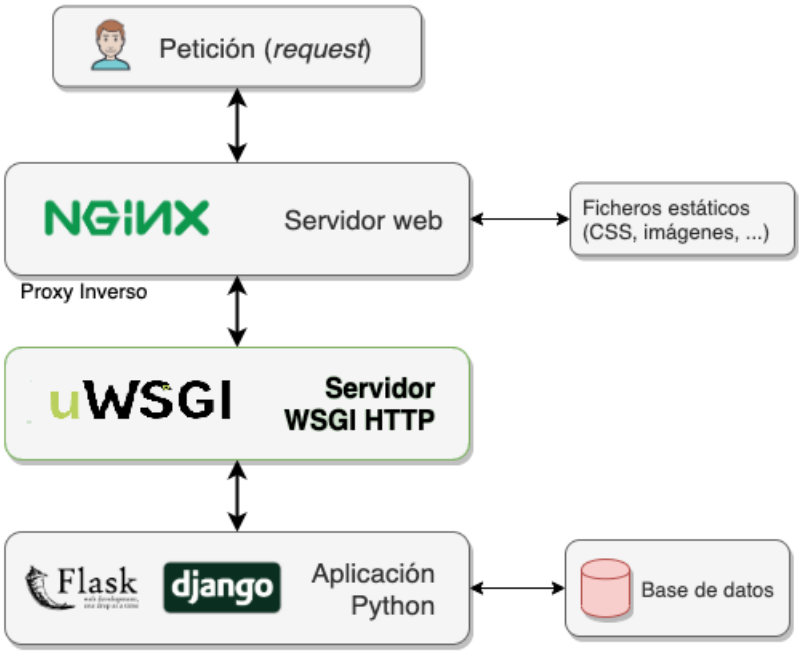


Figure 5.8: Linux server screenshot.Click here for image source credits.

```
rionegro > dashboard > wsgi.py
1  from index import server as connection
2
3  if __name__ == "__main__":
4
5      connection.run()
```

Figure 5.9: Entrance point

to "connection". Next we tell it to start the master mode and create five processes to provide real requests. After that, a Unix socket is created, which in turns serves as a bridge Nginx and uWSGI, alongside with the corresponding permissions and the option vacuum turn on that takes care of deleting the socket once the connection is terminated. Finally, the option "die-on-term" is turned on.

Being done with the configuration file, we need to create a service manager file "systemd", which allows to the init Linux system to start uWSGI automatically and put to work the python application once the server is loaded. To do that, the file rionegro.service is created in the proper location at "/etc/systemd/system/".

The service file is shown in figure 5.11, and we proceed here to breaking it down. It is started with the header [Unit], which contains a description of the service and instructs the system init to start only after the network has been reached. It is followed by a [Service] block, which specify the user and the group. We then endow "www-data" with the group property, to allow Nginx to communicate with the uWSGI processes. The paths corresponding to the dashboard as well as the virtual environment are established.

After this, we need to provide the service manager with the path to uWSGI, that needs to be installed inside the virtual environment. We then point to the configuration file .ini and tell uWSGI to execute in "emperor" mode. The last block is called [Install], and is in charge to tell systemd to start as soon as the multi-user is up.

rionegro > dashboard > rionegro.ini

```
[uwsgi]
module = wsgi:connection

master = true
processes = 5

socket = conn.sock
chmod-socket = 660
vacuum = true

die-on-term = true
```

Figure 5.10: Uwsgi configuration file

```
[Unit]
Description=uWSGI instance to serve rionegro project
After=network.target

[Service]
User=egas
Group=www-data
WorkingDirectory=/home/egas/rionegro/dashboard
Environment="PATH=/home/egas/rionegro/myprojectenv/bin"
ExecStart=/home/egas/rionegro/myprojectenv/bin/uwsgi --emperor rionegro.ini

[Install]
WantedBy=multi-user.target
```

Figure 5.11: Service Manager File

The last step is to configure Nginx to transmit the web requests to the socket by using the uwsgi's protocol. In order to do that, we need to set up a server block at the location /etc/nginx/sites-available.

```
server {
    listen 80;
    server_name rionegrodatascience.com www.rionegrodatascience.com;

    location / {
        include uwsgi_params;
        uwsgi_pass unix:/home/egas/rionegro/dashboard/conn.sock;
    }

}
```

Figure 5.12: Block server configuration file

A snapshot of the block server configuration file is shown in figure 5.12, and is described next. It starts by indicating to Nginx to listen to the http port 80 as well as to the web domain rionegro-

datascience.com. This domain has been already associated to the IP server. We open a location block to include the uwsgi_params file that specify some general parameters of uWSGI. We then pass the requests to the socket by the directive uwsgi_pass. Finally, this file is saved in the proper location /etc/nginx/sites-enabled. After restarting nginx our domain www.rionegrodatascience.com is ready to take visitors.

Summary: Front and Back End. In this section, we have shown how the Front End looks like and have provided a detailed explanation on the Back End construction. The Front End allows the end user to interact with different plots that will allow the entity to make immediate conclusions and take quick decisions based on revenue forecast and sensitivity analysis. We believe the user will get an automatic connection to the app and the experience is very enjoyable.

Conclusions and Future Work

In this project we have used tax billing and payment data provided by Rionegro Mayor's office (Alcadía de Rionegro) corresponding to property and business tax. After cleaning and filtering the useful variables, we have proceeded to perform in-deep exploratory data analysis to capture insightful statistical information that allow us to make early conclusions. Based on the aforementioned descriptive statistical analysis, we built a variety of models supported on classical econometric methods, statistical modeling and modern machine learning techniques in order to forecast future tax revenue as well as sensitivity analysis.

We saw a consistent trend towards increasing of tax revenue in both, property and business tax. We found a striking regular behavior in the property taxes data, which was not mirrored in the business data. This regularity translates into a clear seasonality in Property tax, which is related to government incentives in the form of tax discounts for early taxpayers. This in turn indicates that those incentives are indeed working.

A regular seasonality was not found in most of the sample on Business tax; However, the last three years had a seasonal pattern, probably caused by a recent reform on discounts.

The multivariate model uses external variables that were impacted by COVID-19, however this impact was not reflected in the real tax revenue *

5.3 Future work

There are relevant insights for Rionegro's mayor office in our analysis. However, there are some new directions that could be explored in future work. Those include:

- The data used in this project, created and generated by the Mayor's office, has the potential to be used for fraud prediction. If one is able to identify some cases in which fraud has been proven, that information could be used to train a model to predict fraud cases.
- The tool has the potential to be used in other municipalities to forecast their revenues. Information on tax payments is the main input for the models we estimated, which should be easily accessible for most municipalities in the country.
- The models we used could be improved if we had the chance to include detailed information about properties and business (e.g. discounts, socioeconomic stratum, location, etc.), which might explain payments' occurrence and values. We faced difficulties joining data on tax bills, which had more detailed information about properties and businesses, with tax payments. Moreover, having information about the location of properties and businesses facilitates geographical data analysis and visualization, which we consider might provide further insights for our models.
- Similarly, including more information about properties and businesses, might allow us to refine the model to detect outliers.
-
- Another potential improvement is to link the sensitivity analysis and the forecasting models. This might allow to asses the impact of variations in tax rates and tax base over revenue over time, which will support the discussions of reforms and improvements to the tax code.
-

Fraud prediction. One approach is a database with proven frauds, and then it might be interesting to train a model to predict fraud.

Apply the tool in other municipalities.

Include the influence of discounts. Also, influence of other factors to forecast future payments. Socioeconomic stratum, location, among others. This could not be explored in this work due to the difficulty to merge the declaration and payments databases.

Refine outlier detection controlling for characteristics other than the amount paid of each individual so that more precise statistical distributions can be constructed.

Link sensitivity analysis with forecasting models. So that the impact of variations in tax rates and tax base over revenue over time can be analysed to aid in the discussion of reforms and ultimately optimize the tax code.

Data visualization by zone. We found some Geojson information, but it was not possible to match polygons IDs with the zones used in the databases.

Acknowledgements

We would like to specially thank to our TAs Karen Figueroa and Nilson Mossos, for being with us at every step of the project, helping us with doubts, support, resources and make the process a fun and enjoyable ride. We want to thank to all the professors in charge of the main classes, for the hard work preparing the cases and the huge effort every one of them put on making a clear exposition of the cases despite the challenging conditions of this particular course. Of course, we are extremely thankful with Correlation One for putting this amazing course together, providing us with all the resources, but even more, the thoroughly built of quality content, that we were so lucky to get, as well as for choosing us to be here, among thousands of equally worth it applicants. Last but not least, we would like to thank to MinTic Colombia for providing the financial and logistic resources that made all of this possible, creating along the way a unique opportunity to improve the digital literacy in the country, which lately and in a longer term will benefit the whole socioeconomic status of our beloved country, as well as to contribute to two of the most important factors needed in the development of a nation, namely education and innovation.

6. Bibliography

[1] National Planning Department's Sisfut 2. *"Resultados Nuevo IDF 2020"*. https://sisfut.dnp.gov.co/app/descargas/visor-excel. Accessed: 2021-09-03 (cited on page 4).

[2] *Aerocivil*. https://www.aerocivil.gov.co/atencion/estadisticas-de-las-actividades-aeronauticas/bases-de-datos. Accessed: 2021-06 (cited on page 7).

[3] Elham Buxton et al. "An Auto Regressive Deep Learning Model for Sales Tax Forecasting from Multiple Short Time Series". In: *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. 2019, pages 1359–1364. DOI: 10.1109/ICMLA.2019.00221 (cited on page 22).

[4] *Camara del Comercio de Antioquia*. https://ccoa.org.co/wp-content/uploads/2021/07/Acer-Junio-2021.pdf. Accessed: 2021-06 (cited on page 7).

[5] François Chollet. *Deep Learning with Python*. Manning, Nov. 2017. ISBN: 9781617294433 (cited on page 22).

[6] *Comisión de Estudio del Sistema Tributario Territorial. Informe Final.* https://economia.uniandes.edu.co/sites/default/files/webproyectos/comisionstt/CESTT-Informe-web.pdf. "Accessed: 2021-09-03". 2020 (cited on page 4).

[7] *Dane*. https://www.dane.gov.co/index.php/estadisticas-por-tema/precios-y-costos/indice-de-precios-al-consumidor-ipc/ipc-informacion-tecnica. Accessed: 2021-06 (cited on page 7).

[8] *Digitalocean site*. https://www.digitalocean.com/community/tutorials/how-to-serve-flask-applications-with-uswgi-and-nginx-on-ubuntu-18-04. Accessed: 2021-07 (cited on page 32).

[9] Walter Enders. *Applied econometric time series*. John Wiley & Sons, 2009 (cited on page 24).

[10] *National Planning Department's Territata*. https://terridata.dnp.gov.co/index-app.html/perfiles. Accessed: 2021-05 (cited on page 4).

[11] *Nginx documentation site*. http://nginx.org/en/docs/. Accessed: 2021-07 (cited on page 32).

[12] Alcaldía Municipal de Rionegro (Antioquia). *Marco Fiscal de Mediano Plazo 2021 – 2030*. https://sisfut.dnp.gov.co/app/descargas/visor-excel. Accessed: 2021-09-03 (cited on page 4).

[13] *Statsmodels documentation site*. https://www.statsmodels.org/stable/index.html. Accessed: 2021-07 (cited on page 21).

[14] *uwsgi documentation site*. https://uwsgi-docs.readthedocs.io/en/latest/WSGIquickstart.html. Accessed: 2021-07 (cited on page 32).

[15] Jeffrey M Wooldridge. *Introductory econometrics: A modern approach*. Cengage learning, 2015 (cited on page 25).

[16] Liu Li-xia, Zhuang Yi-qi, and Xue-yong Liu. "Tax forecasting theory and model based on SVM optimized by PSO". In: *Expert Systems with Applications* 38.1 (2011), pages 116–120. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2010.06.022. URL: https://www.sciencedirect.com/science/article/pii/S0957417410005269 (cited on page 22).