Facultat de Matemàtiques i Estadística (FME)
Màster en Estadística i Investigació Operativa

Programació i Bases de Dades Estadístiques

Final Project Journal

Autors: Leonel Fernando Nabaza Ruibal

Professor: Dr. Àlex Barceló Cuerda

Barcelona, December 26, 2025

Resum

Aquest document esdevé un resum dels diferents procediments portats a terme i observacions personals durant la realització d'un projecte que comprèn l'aplicació d'un model d'aprenentatge automàtic i d'un test estadístic, per tal de poder estudiar dades econonòmiques i socials i el seu impacte en la desocupació als EEUU.

Abstract

This document contains a summary of the different procedures carried out and the personal observations during the materialization of a project that comprehends the application of a machine learning model and a statistical test, with the purpose of studying the impact of social and economical data on unemployment in the US.

Contents

Chapter 1

Data loading and first processing

For this project my objective was to study with economical data. In the first place, I wanted to try to use a random forest regressor. In the second place, I wanted to do a hypothesis test.

After a careful inspection of the proposed dataset sources, I didn't find any which had economical data I was motivated to explore. So I decided to look for other sources on the internet. I found that Yahoo has its own finance section and that there is a package that contains many economical datasets from this section (`yfinance`[1]). I downloaded data on the SP500 bonds and while reviewing it I came up with the following idea: study the unemployment (in the US, because SP500 is an index of American companies, mainly) depending on several social and economical variables. Later, I also found out that there is a package, named `pandas_datareader`[2], which contains lots of datasets. One of the sources of these datasets, is the Federal Reserve Economic Data. Both of these packages have proper documentation, and the functions for importing the necessary datasets are very user-friendly.

I decided to study the target variable `unemployment`, which came from the `pandas_datareader` package with respect to the variables:

- `CPI_Inflation`, from `pandas_datareader`.
- `Initial_Claims`, from `pandas_datareader`.
- `Intrest_Rate`, from `pandas_datareader`.
- `Consumer_Sentiment`, from `pandas_datareader`.
- `Industrial_Production`, from `pandas_datareader`.
- `SP500_Index`, from `yfinance`.

[1] https://pypi.org/project/yfinance/
[2] https://pandas-datareader.readthedocs.io/en/latest/readers/fred.html

The most important challenge regarding this part was merging all the the data into an appropriate format. These features were gathered with different periodicities; for instance, the price of the `SP_500` bonds was updated in a daily way, while `Initial_Claims` was a weekly variable. To tackle this problem I first outer-joined the columns, so that data from no column was lost. Then, I proceeded to group the data in a monthly format, which is the most informative and compatible for treating the features. For that I took the mean for most of the features (except for interest rate and industrial revolution, for which I took the last values of the corresponding period) and applied the function `resample` with argument 'ME' (monthly). One of the mistakes I did before coming up with the correct procedure was to inner join the variables, deleting essential information and getting a lot of empty values, because of the rigidity of the clause. Moreover, I had to dig a little bit into the documentation of `Pandas` before finding the `resample` function.

Another interesting challenge was regarding the data from the SP500. When I initially imported it, it contained several columns regarding the opening, closing, high and low values of the bonds. All these values are very similar and had to make a plot of the correlation matrix to make sure. One of my concerns regarding this is that it may cause problems for the model to learn from this data. That is why I decided to only make use of the closing price data.

Chapter 2

Applying a Random Forest Regressor

Once the data was properly prepared, I just added another modification: my target variable, which was the unemployment, had to be shifted in one month. This is because I was interested in predicting the unemployment level of the next month, not the one in which I was already in. That is why I discovered the function `shift` by reading the documentation. Since I wanted just one month (one row shift), the argument for the function was -1.

With the data properly prepared I distributed it into training and testing dataset (separating into target variable and data). One important remark is that the data was not randomly separated. Since we are working with time series, the test data was the 20% of most recent data, and the other 80% was for the training. This is because for this kind of problem, there is a dependence of the current variables with respect to their previous ones. So, an 'omission' of some past value can negatively impact the performance and generalization of the model.

Taking the above into account, for the training I followed the standard procedure learnt during the course, which consisted into calling the functions `fit` and `predict` from the `sklearn` library.

My metrics for testing have been the MAPE (Mean Absolute Percentage Error) and MAE (Mean Absolute Error). I consider both of this metrics complementary to understand the results. In this case, I obtained a MAE of 1.15 (which means that the average error in predicting the percentage of unemployment is of 1.15%) and the MAPE of 27.03%.

Although this results seem not great, we need to remember what we are doing. We have chosen a small set of features, each with different frequencies, regarding economical and social information and built a model that attempts to predict the level of unemployment next month. So, maybe there are many more variables that have a more direct influence over the unemployment and/or maybe the variables coming from "social" measurements (such as `Consumer_Sentiment`) may not be very accurate and thus negatively affect the model. Taking this into consideration, our results are very positive.

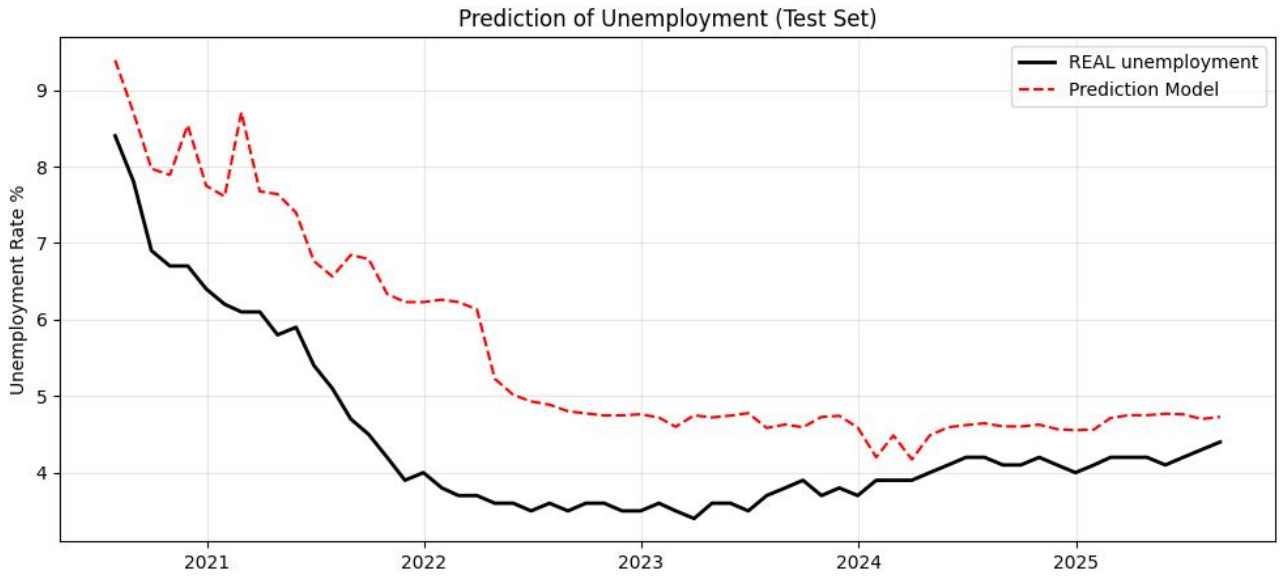

Figure 2.1: Result of predicting the unemployment with the Random Forest Regressor

Finally, I also investigated the importance of our features for the model, taking advantage of one of the attributes of the object that I was using: `feature_importances_`. Printing this (via a bar plot from the library `seaborn`) I observed something that I hadn't expected:

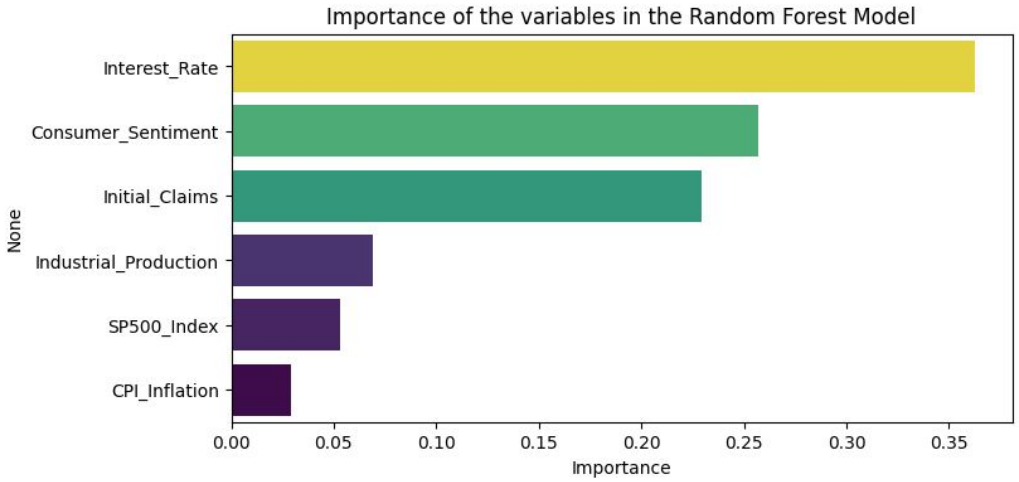

Figure 2.2: Importance of the features

This result is personally surprising for me, as I would've expected that for forecasting the variable `Initial_Claims` would be the most important. And I say that because this variable means people that have been recently registered as "job-seekers". A possible explanation for this effect is that since this is the employment demand for the previous month that we are

trying to predict, it looses importance with respect to other variables.

To further study this result I created another model that tried to predict the unemployment of the same month as the given data. This consisted in applying the same code as before that with no shift in the unemployment target variable. I reached the following barplot for the importance of the variables:

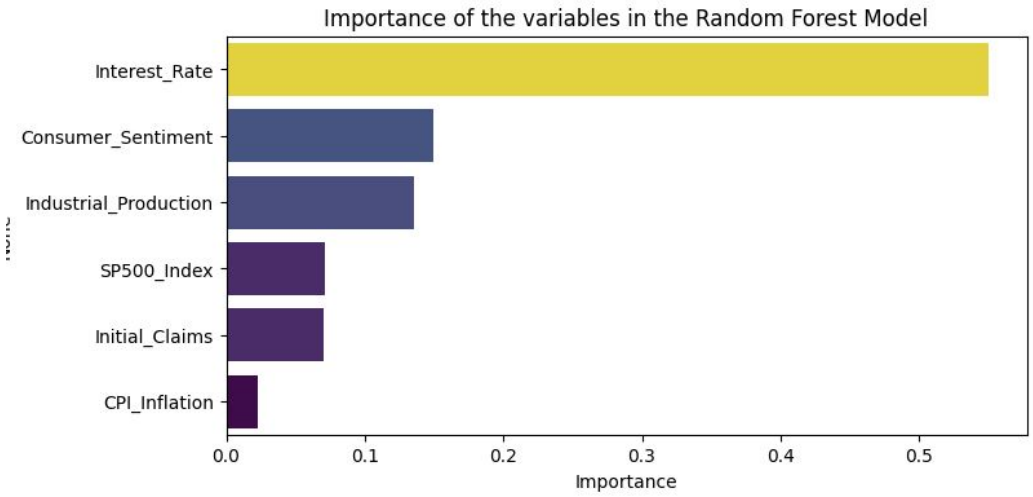

Figure 2.3: Importance of the features with no shift in unemployment

Understanding this result confused me a bit more (in the beginning). After putting some thought (thinking as an economist) and doing some research, I got to the following conclusion: higher rates increase borrowing costs, slowing business investment, consumer spending, and growth, leading to reduced hiring or lay-offs, while lower rates decrease borrowing costs, encouraging investment, boosting demand, and spurring job creation, thereby lowering unemployment. That's why the `Interest_Rate` is so important compared to the rest.

Chapter 3

Statistical analysis for a linear regression

My intention with this part is to check if for a given linear model, the coefficient for one of the features (I picked the one for `SP500`) should be zero or different from zero.

The procedure of training and testing was exactly the same as the one for the previous model, but selecting a linear one instead. Then I did a plot for the coefficients of the model, to visually see if there was any 'anomaly' (value for some coefficient that was very different from the rest). Since this was not the case, I proceeded with the statistical analysis, applying what I learn in the Linear and Generalized Linear Models Course.

$$\text{reject } H_0 \iff |T_j| \geq q_t \left(1 - \frac{\alpha}{2}, n - p - 1\right)$$

Although this idea is explained more in depth in the report, the implementation of this test was considerably straightforward: had to create some statistical variables, such as the estimator for the variance or find the number of observations, the number of features, etc. The most difficult part was for computing some inverse matrix that it is required, for which I had to make use of the function of `linalg.inv` function from the `numpy` library. After computing my test statistic I found the critical region using a t-Student distribution with the aid of the library `scipy.stats`.

The result of my test was that variable `SP500` should not be excluded from the model. Moreover, to further reinforce the results. I created a plot that compares the evolution of `Unemployment_Rate` and `SP500` over time, and it can be clearly seen that there exist some negative dependency (when one variable goes down the other one goes up, and viceversa).

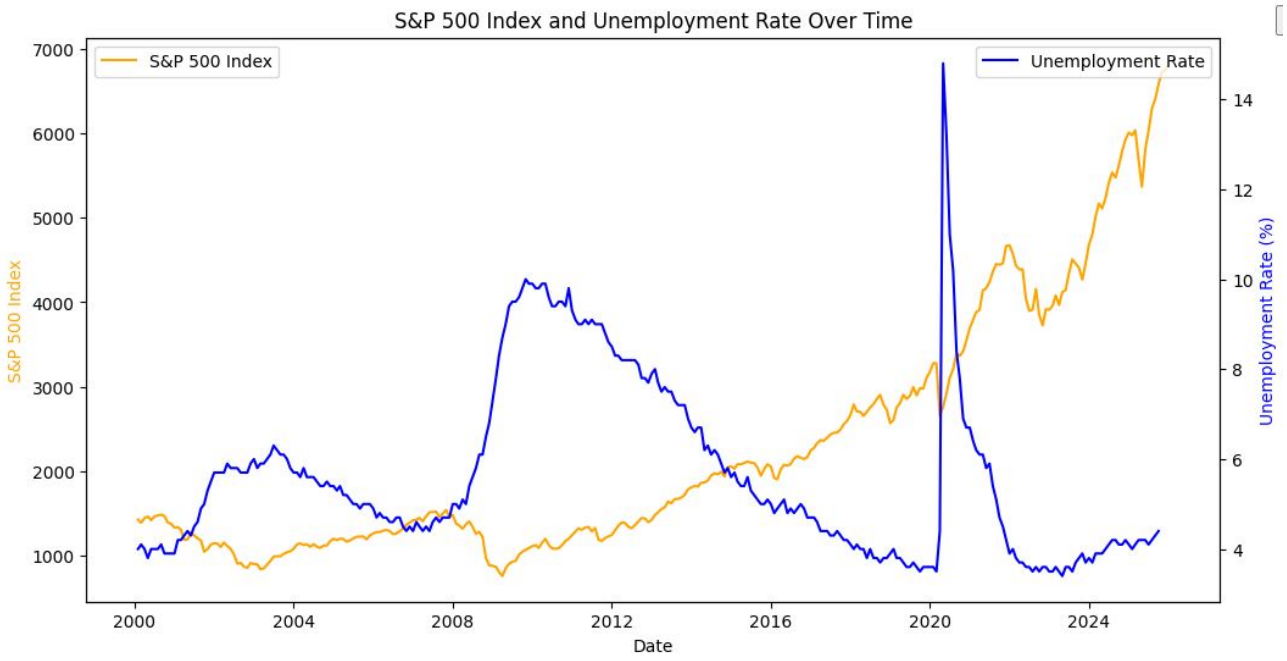

Figure 3.1: Evolution over time Unemployment and SP500

For this plot I used what we learnt on the library `matplotlib`.

Chapter 4

Appendix

For the corresponding project: see https://github.com/LeonelFNR/PBDE-Project.