Aprendizagem Automática: *Course Project*

Grupo 29: Alexandre Monforte (54491)(10H), João Braz (60419)(10H), Joel Oliveira (59442)(15H)

Create the best model classifier in order to predict the variables of country population, fertility rate, and life expectancy for 2017 (and 2018). Process and transform the data to be able to apply the models Linear Regression, Support Vector Regressor and Decision Tree Regressor to the data.

INTRODUCTION

The main goal of this Project is to create 3 models to predict these 3 variables: country population, fertility rate, and life expectancy; from these 3 different files: *country_population.csv*, *fertility_ rate.csv*, and *life_ expectancy.csv*; respectively. Those files were obtained from the dataset used in the 2006 TED Talk *The Best Stats You've ever seen* by Hans Rosling. Those files contain values for the respective variables from 1960 to 2016. Each column of each file contains the following values: *Country Name*, *Country Code*, *Indicator Name*, and *Indicator Code*; and each row contains countries and some agglomerates of countries such as *Central Europe and the Baltics* or *East Asia & Pacific.* To sum up, each file has 61 columns and 265 rows. It's important to notice that there are some missing values present in each file, but the treatment of that is explained further.

Attending to the nature of our dataset and goal the way that we chose to build our train and test sets are explained further on this report. We decided to do the optional objective which is to predict 2018. To validate our models we'll use the real data for 2017 and 2018 obtained from the *World Bank.* We tested our models for 10 countries selected randomly.

DEVELOPMENT

Data Processing

For all the datasets there were some redundant columns, for instance, *Country Name* and *Country Code*, or *Indicator Name* and *Indicator Code.* Each country has a name and a code and each indicator also has a name and a code. These columns have a relation one-to-one. Since there is no need for this redundancy, we started by removing the variables *Country Code* and *Indicator Code.*

Besides, the indicator variable refers only to the type of data that the row corresponds to (total population, fertility rate, life expectancy). It is constant for each of the 3 files we were given. Since we are building 3 models, one for each of those different measures, and not a global model that takes this information into account for the predictions, we also removed this column. With this, we get a matrix where each column corresponds to a country and each row to a year.

In order to be able to predict the value of some years using the values of the previous years, the data needs to be restructured. We need to have a matrix X with a response y, to train the *Sklearn* models. We started by making this process for a single country and then replicated it for every country.

Our dependent variable will be the data for each year, which is what we want to predict. The independent variables will be the data correspondent to the previous years.
We can see this as a sliding window that traverses the time series. This window has a pre-defined fixed size N. An example is shown in Figure 1, where the window size is 2. In this figure, the numeric values are a year reference. The green set corresponds to the independent variables and the red arrow points to the expected y value. Each step of the window will be an entrance in the X matrix.

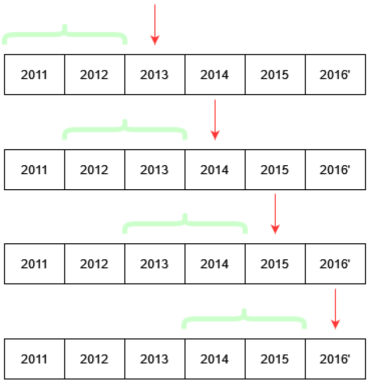

FIG. 1: Example of the Sliding Window (Green) used as the independent variables in the Training Stage to Predict the dependent Variable (Red).

In Figure 2 we can see the resulting X matrix (green) and the y vector (red). This resulting matrix is in a format that the *Sklearn* models can interpret. We can see the overhead in memory that this causes, as it has lots of redundancy.

Another aspect to notice is the fact that from 6-year values, we finished with 4 rows for training. This is because of the window size. We only considered rows without missing data for the training. Since in the example there were no data previous to 2011, both 2011 and 2012 can't be considered, as they would need information from 2009 and 2010. For this same reason, countries without previous data can not be predicted by the model. Since the model uses historical data to make predictions, it can not forecast values for countries that did not have past information.

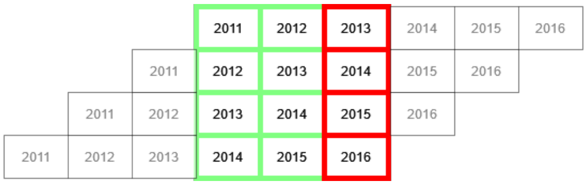

FIG. 2: X (green) and y (red) matrix that resulted from Fig. 1.

As mentioned, this procedure was made for each country, and then the final matrices were joined. Because we did not know if the model would get better results if it did have information about the country it is predicting we created a model which would use country information if a flag was set to *true*. If using country information, a further step of pre-processing was needed. Each instance in the matrix had an extra field in the independent variables, which was the country one-hot encoded.

Before making any further work on the data such as finding extra useful data that could improve the model, we made an experiment, training the model with data until 2015 and testing on the year 2016 using a random sliding window size of 5. We obtained unexpected impressive results, which led us to not look for further data, as the data we have is sufficient for fairly good outputs.

Model Selection and Criteria Selection

The models that were chosen to predict 2017 and 2018 are **Linear Regression**, **Support Vector Regression**, and **Decision Tree**.

The choice of using Linear Regression was supported by the fact that there are some strong correlations over some years. Indeed, Linear Regression is one of the simplest models, always optimal and very fast to run.

To compare with the simple Linear Regression model, we find it interesting to select Support Vector Regression, which is a complex model. Further, will compare and see the results obtained for each model. Actually, because we don't have a very large dataset, it was possible to use the Support Vector Regression model, if it was not the case, the model could take too long to train and test. However, whenever it's possible, Support Vector Regression is a nice model to use.

For last, we chose the Decision Tree model. Decision Tree can be considered a simple model. As said before, sometimes the trends over the years change a lot and the Linear Regression can be a little bit limited in those situations, so to gap those situations, we find that using a Decision Tree model could be a nice bet. In fact, the Decision

Tree is fast to learn and test, the inconvenience is that is very sensitive on the hyperparameters. Because the main goal is not to find the "best model", just a few parameters were tested that were shown further.

Because we are working with regression models, we chose as criteria the *root mean square error* and the *Pearson Correlation*. When training all models we compared the one with the smallest RMSE and the one with the best Pearson Correlation. To choose between them we decided to choose the one that uses the window (explained in the section *Data Processing*) with the smallest size because we considered that a small window size means a simpler model.

Model Evaluation

The results of the RMSE and Pearson Correlation for the three different problem sets are presented in Tables II to V, which contain the best parameters for each problem set. These parameters include the number of lags (previous years considered in the prediction), the starting year for which the model can make a prediction, and whether or not to include country information (encode_country).

The Support Vector Regressor (SVR) was trained using two different kernel functions: "rbf" and "linear". The results showed that the "linear" kernel performed the best. The SVR also has hyperparameters called "C" and "gamma", which control the complexity of the model and the smoothness of the decision boundary, respectively. In this experiment, the default values of "C" and "gamma" provided by the *scikit-learn* library (C = 1, gamma = "auto") were used and produced the results shown.

For the Decision Tree Regressor (DTR), the hyperparameters studied were the *max_ depth* with values between 1 and 3, *min_ sample_ leaf* with values between 1 and 3, and the default parameters, so there were tested 10 different models for each problem. The results indicated that the default values for these hyperparameters performed the best in every single case.

TABLE I: Training Results for RMSE and Pearson Correlation for the country population using the models Linear Regression, Support Vector Regressor, and Decision Tree Regressor with their best hyperparameters.

Statistic	**LR**	**SVR**	**DTR**
Pearson Corr.	0.9999999991	0.9998747104	0.9999998768
RMSE	1847.51	42847224.96	80602.88

TABLE II: Each model's best parameters for lags, start and encode_country to predict country population.

Parameter	**LR**	**SVR**	**DTR**
lags	4	46	8
start	2010	1969	1962
encode_country	True	False	True

TABLE III: Training Results for RMSE and Pearson Correlation for the fertility rate using the models Linear Regression, Support Vector Regressor, and Decision Tree Regressor with their best hyperparameters.

Statistic	**LR**	**SVR**	**DTR**
Pearson Corr.	0.9999872	0.9999889	0.9999892
RMSE	0.0081	0.0076	0.0100

TABLE IV: Each model's best parameters for lags, start and encode_country to predict fertility rate.

Parameter	**LR**	**SVR**	**DTR**
lags	11	6	2
start	2002	2000	2007
encode_country	False	False	True

TABLE V: Training Results for RMSE and Pearson Correlation for the life expectancy using the models Linear Regression, Support Vector Regressor, and Decision Tree Regressor with their best hyperparameters.

Statistic	**LR**	**SVR**	**DTR**
Pearson Corr.	0.9999957	0.9998677	0.9999354
RMSE	0.034	0.120	0.079

TABLE VI: Each model's best parameters for lags, start and encode_country to predict life expectancy.

Parameter	**LR**	**SVR**	**DTR**
lags	12	11	1
start	1960	1996	1963
encode_country	False	False	True

Final Results

The Linear Regression model was chosen for predicting country population due to its high Pearson Correlation, and simplicity. The results for the years 2017 and 2018 are shown in Table VII. As expected, the RMSE is higher for 2018 than it is for 2017, as the prediction for 2018 is based on the prediction for 2017, leading to a compounding error for each additional year that is predicted. Overall, the Linear Regression model performed well in this task.

TABLE VII: Results of country population predictions for 2017 and 2018 using Linear Regression model.

Year	2017	2018
RMSE	148444.70	192338.34
Pearson. Corr.	0.999995	0.999992

For predicting fertility rate, the Support Vector Regressor (SVR) model with the best hyperparameters obtained from the prediction for 2016 was selected due to its low error rate. The results of these predictions are shown in Figure VIII. The SVR model performed well in this task.

TABLE VIII: Results for Fertility Rate predictions for 2017 and 2018 using Support Vector Regressor model.

Year	2017	2018
RMSE	0.62	0.65
Pearson. Corr.	0.997	0.996

For the Life Expectancy prediction, the model selected was the Decision Tree Regressor, since it gave the least amount of error and is simple. Figure IX shows the results of those predictions. We can consider that it performed well too.

TABLE IX: Results for Life Expectancy predictions for 2017 and 2018 using Decision Tree Regressor model.

Year	2017 the	2018
RMSE	0.467	0.470
Pearson. Corr.	0.955	0.952

All of the models performed exceptionally well, except when there was a sudden shift in the trend.

CONCLUSION

To conclude, we can use simple models to make that nature of predictions and even get very good results. However, the further the year we want to predict, the greater the error we get, as shown by the results when comparing the predictions for 2017 with 2018. It's expected because the error for each additional year will be accumulated. We saw that for the country population the best model was Linear Regression, which means that using a linear model could be a great bet for predicting values that have a trend along the years.