Final Project

Aprendizagem Automática

2022/2023

Duarte Raposo
f59099@alunos.fc.ul.pt
42 WHs

1 Data Pre-Processing

1.1 Observation and planning

For the data preparation, first things first, a recognition of the three datasets and their correlation was made through graphic demonstration and a brief look at how the datasets were organized.

With the graphs shown in figures 1, 2 and 3, and through observation of the meaning behind each of them, it could be said that their variations are correlated. Whenever the fertility rate increases, the population should gain more people, and whenever the life expectancy increases, the population should lose fewer people.

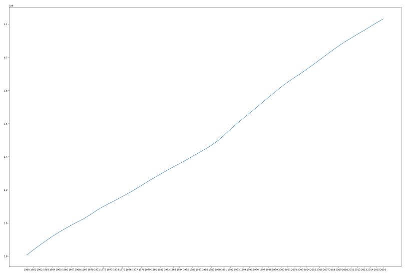

Figure 1: Country Population USA

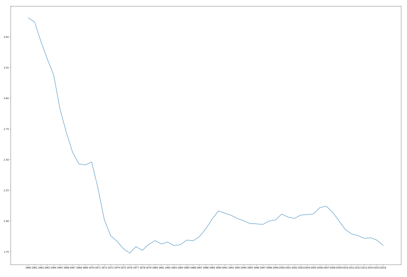

Figure 2: Fertility Rate USA

But, as observed, for the USA at least they balance each other out, showing the country population almost as a linear function, with a fertility rate decreasing and life expectancy increasing.

With this in mind, a general dataset that merges all the 3 variables, both to serve as observations and sequential inputs.

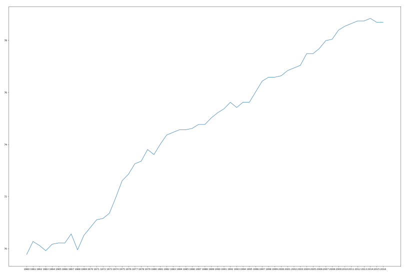

Figure 3: Life Expectancy USA

1.2 Getting Extra Data

Another factor that could be relevant for the fluctuations of population is the economic state of a country, thus it was also added a dataset with the **Gross Domestic Product** of **each country**, taken from the same source as the population datasets (**GDP Dataset Source** and USA graph that shows some correlation with the displayed previously for the country population).

1.3 Final Data-format and Handling NaN values

In the end, it was decided that the model should use the **2 previous** and the **current values** to try to detect the orientation of the data (country population is growing or fading, and so on), the country category (each country has different cultures, climates, and tendencies, to represent them we use the **three letter representation of the country**) and the **current GPD** for said country.

To **handle the NaN values**, after the data was organized sequentially with each data entry containing the previously depicted format, whenever a data entry had missing values, it would get discarded. This way, we maintain plenty of data for each country without inventing values that don't exist.

We also kept in the dataset the **use of delta** (difference between current values and future values) and **rate of change** (division) to later test and see if there are any differences on the raw results.

1.4 Final Data Processing Remarks

For this entire section, it was also tested scaling of data and one hot encoding, but the results seemed similar than what we obtained normally, so it was kept the use of the non-processed results, and later on for Support Vector Machines, dimensionality reduction is tested too.

Finally, the model was separated into a **train set** (**80%**), a **test set 15%** and an **independent validation set** (**5%**).

2 Choosing the Models

At least three models were to be selected. For this, I used a package named lazypredict, which runs fast tests using the Scikit-learn models and displaying both the performance and the speed of each of them (results displayed in the table 1).

After getting the displayed results, it was decided to try and test the **Linear Regression** model for the results that it displayed and for taking less time, a **Random Forest Regression** model and **Extra Trees Regressor**, as the Extra trees had the best results, even though it takes longer to compute, and a Support Vector Machine (**SVR**) to try and find out why the results were so bad and if a PCA Kernel implementation would improve it's displayed results.

3 Modelling Results

For each model, the procedure started by **running a simple test**, and then the ones which have **better results are optimized through k-fold**.

Table 1: Lazypredict Results

Model	Adjusted R-Squared	R-Squared	RMSE	Time Taken
ExtraTreesRegressor	1.00	1.00	219095.65	1.56
LinearRegression	1.00	1.00	347054.76	0.01
BayesianRidge	1.00	1.00	347114.26	0.04
RandomForestRegressor	1.00	1.00	370370.85	3.92
XGBRegressor	1.00	1.00	467744.83	0.24
BaggingRegressor	1.00	1.00	471813.57	0.41
DecisionTreeRegressor	1.00	1.00	525750.84	0.07
ExtraTreeRegressor	1.00	1.00	538276.67	0.02
Lars	1.00	1.00	538569.52	0.01
HistGradientBoostingRegressor	1.00	1.00	575385.28	0.64
GradientBoostingRegressor	1.00	1.00	697541.69	1.55
KNeighborsRegressor	0.99	0.99	956346.28	0.03
GaussianProcessRegressor	0.99	0.99	1472185.34	3.92
AdaBoostRegressor	0.97	0.97	2129968.67	0.43
Ridge	0.96	0.96	2466877.13	0.01
Lasso	0.94	0.94	3049254.85	0.49
ElasticNet	0.89	0.89	4121963.94	0.03
SVR	-0.13	-0.12	13162664.62	2.13
MLPRegressor	-0.13	-0.12	13188134.35	3.57

The k-fold focuses on the **mean squared error**, as the ratio of variance explained and the correlation are already with almost perfect scores, thus, we look into decrease the error margin. To better evaluate the predictions, we **multiply** the **country population** with the **fertility rate** and **life expectancy**, so they all have an equal weight in the predictions, since the country population has such a big influence on the values variations.

3.1 Linear Regression

The LinearRegression model is the one model which not only shows promise in the previous test done with the lazypredict package, but also is fast to train and predict, as it is a simpler model.

3.1.1 Results of the k-fold optimized LinearRegression model and it's hyperparameters

Table 2: Optimized LinearRegression Results

Metrics	For the country population	For the fertility rate	For the life expectancy	Total Flattened	Total Multiplied
RVE	0.9999998082694822	0.9995849998909416	0.9997100207975071	0.9999998222321135	0.9999844256987634
mse	329387.92678138695	0.0392008771047105	0.1776551402871918	190172.20819507548	114233412.21585716
Correlation Score	1.0000	0.9998	0.9999	1.0000	1.0000
Maximum Error	5898102.585692406	0.421196132695127	1.8786818075138356	5898102.585692406	1113991954.2149467
Mean Absolute Error	86338.19756183107	0.01957113113905548	0.08974679672894503	28779.43562658631	36130571.16479758

The LinearRegressor model doesn't have many parameters to tweek, but with the K-Fold Optimization, the represented model in table 2 was achieved using the hyperparameters:

fit_intercept=False; normalize=False; copy_X=False; n_jobs=60; positive=False.

And no other equivalent hyperparameter combinations.

3.2 Support Vector Regression

It was the only one where it was necessary the use of individual models for each prediction feature, as it doesn't allow the batch prediction. This could justify why the lazypredict package didn't have that good of a result for it, and it took more time to compute.

As for optimization, the k-fold wasn't applied, but it was tested in a PCA kernel to decrease dimensionality and see if it had any impact in its results, but the results were disappointing, as the SVM models already use kernels before computation.

3.2.1 Results of the simple SVR model testing

Table 3: SVR Test Results

Metrics	**For the country population**	**For the fertility rate**	**For the life expectancy**	**Total Flattened**	**Total Multiplied**
RVE	0.9997158594505441	0.9987148968636281	0.9991042130309291	0.9997178629051118	0.9996218720289838
mse	13162691.147930166	0.07138562313724392	0.3328081647534648	7599483.277517388	639481340.011289
Correlation Score	1.0000	0.9994	0.9996	1.0000	0.9999
Maximum Error	82426894.80390263	0.4328033571551808	3.622542837705211	82426894.80390263	6613668556.287201
Mean Absolute Error	4411002.638486993	0.05229561545246112	0.21586857890565528	1470334.3022170623	215681864.42249528

As we can observe, the results aren't too bad, compared with the ones that the lazypredict package demonstrated. But as this model has many hyperparameters, and not only it takes a longer time to compute than the Linear Regressor, but also it uses more resources as it needs to train and compute not one but 3 models, one for each prediction feature, it is still not optimal.

3.3 Random Forests Regressor

This model showed ok

3.3.1 Results of the k-fold optimized RandomForestsRegressor model and it's hyperparameters

Table 4: Optimized RandomForestsRegressor Results

Metrics	**For the country population**	**For the fertility rate**	**For the life expectancy**	**Total Flattened**	**Total Multiplied**
RVE	0.9999999367592421	0.9995229886744046	0.9996783482220076	0.9999999414315112	0.9999855161342714
mse	189012.21359089282	0.04206940948907107	0.18714529585551026	109126.25239688519	110110340.30430949
Correlation Score	1.0000	0.9998	0.9998	1.0000	1.0000
Maximum Error	1873316.109999895	0.36360999999999954	1.2571397560974589	1873316.109999895	1395071282.0091553
Mean Absolute Error	67946.93279591737	0.021383205685112767	0.10187123652001023	22649.01868345319	32210424.58954482

The RandomForestRegressor model with the K-Fold Optimization represented in table 4 was achieved using the hyperparameters:

n_estimators=10; criterion=squared_error; max_depth=None; min_samples_split=2; min_samples_leaf=1; min_weight_fraction_leaf=0; max_features=3; max_leaf_nodes=None; min_impurity_decrease=0; bootstrap=True; oob_score=False; n_jobs=None; random_state=None; verbose=0; warm_start=False; ccp_alpha=0; max_samples=None.

And no other equivalent hyperparameter combinations, but as the number of estimators was decreasing, the model seemed to be improving, so there could be a better optimized model with this parameter better explored.

3.4 Extra Trees

This model shouldn't differentiate too much from the RandomForestRegressor, except it did show some better results with lazypredict (being the one it found as the best for this problem when disregarding processing speed), and it is also faster.

Table 5: Optimized ExtraTreesRegressor Results

Metrics	**For the country population**	**For the fertility rate**	**For the life expectancy**	**Total Flattened**	**Total Multiplied**
RVE	0.9999999367592421	0.9995229886744046	0.9996783482220076	0.9999999414315112	0.9999855161342714
mse	189012.21359089282	0.04206940948907107	0.18714529585551026	109126.25239688519	110110340.30430949
Correlation Score	1.0000	0.9998	0.9998	1.0000	1.0000
Maximum Error	1873316.109999895	0.36360999999999954	1.2571397560974589	1873316.109999895	1395071282.0091553
Mean Absolute Error	67946.93279591737	0.021383205685112767	0.10187123652001023	22649.01868345319	32210424.58954482

3.4.1 Results of the k-fold optimized ExtraTreesRegressor model and it's hyperparameters

The ExtraTreesRegressor model with the K-Fold Optimization represented in table 5 was achieved using the hyperparameters:

n_estimators=10; criterion=squared_error; max_depth=None; min_samples_split=2; min_samples_leaf=1; min_weight_fraction_leaf=0; max_features=3; max_leaf_nodes=None; min_impurity_decrease=0; bootstrap=True; oob_score=False; n_jobs=None; random_state=None; verbose=0; warm_start=False; ccp_alpha=0; max_samples=None.

And no other equivalent hyperparameter combinations, but as the number of estimators was decreasing, the model seemed to be improving, so there could be a better optimized model with this parameter better explored.

4 Conclusions

To conclude, the models that showed more promise for the computation of the predictions for 2017 and 2018 are the LinearRegression model and the RandomForestRegressor/ExtraTreesRegressor. To compute this prediction, we will need still the GDP for 2017, since it does not predict this value and this is a sequential model which was trained only to compute the next year. So, the 10 random countries were selected, their GDP for 2017 was inserted, and we predict the values for 2017, and then with the GDP and the predicted values, we use it for 2018.

For these predictions, we use both the ExtraTreesRegressor and LinearRegression models, both trained with the entire dataset until 2016, and the mean of their outputs was computed, leaving the final results shown below.

4.1 Predictions for 2017 and 2018 (only for 2 of the randomized countries)

		2017		2018	
		Predicted	Real	Predicted	Real
	Country Population	65970067.9	66058859.0	66288088.7	66460344.0
United Kingdom	Fertility Rate	1.80	1.74	1.80	1.68
	Life Expectancy	81.026	81.256	81.135	81.256
	Country Population	284309.2	279187.0	281907.7	279688.0
Barbados	Fertility Rate	1.79	1.62	1.79	1.62
	Life Expectancy	76.058	76.936	76.211	77.067

Other 8 country predictions in the Jupyter Notebook file.