Temperature analysis and forecasting using Spark

KTH – Royal Institute of Technology

Data-Intensive Computing – ID2221

Project Report

October 2022

Group 9

Marvin Këçini – kercini@kth.se

Maryam Kheirkhahzadeh – markhe@kth.se

Zineb Senane – senane@kth.se

Gianluca Ruberto – ruberto@kth.se

1. Introduction

1) Problem statement

Temperature forecasting has always been a challenging problem. It is not an easy task during normal times, but today, because of the climate change, it has become critical. In the last years, global average temperature raised as an effect of human pollution. As result, extreme events are always more common and standard, non-machine learning models cannot accurately predict temperature as they did before.

2) Project objective

The scope of this project is to use big data tools such as Spark, and its modules (Spark SQL , Spark ML) to support the development of a machine learning model to solve the previously stated problem.

2. Dataset

The dataset is provided by Kaggle [1]. It consists of around two million measurements made from September 1st 2018, till February 28th, 2019. The feature vectors include both weather-related features such as sun evaluation at the current location, climate values of temperature, pressure and topography, and meteorological parameters on different pressure and surface levels from weather forecast model predictions. The total number of features is 111. The target value is air temperature measurements at 2 meters above the ground.

3. Method

1) Environment

The programming environment that we used is Google Colab. The language used is Python because of the visualization libraries that it offers, like MatPlotLib and Geopandas.

2) Data analysis

The number of rows is 1993574. There are 113 columns including the target variable, however some of them are not useful for training the model since they show if some other features are available or not. We removed these columns from the dataset. 108878 rows had null values and we decided to remove them also, since it is relatively small considering the entire dataset.

To have a visual perspective of our data we plotted the target variable on a world map.

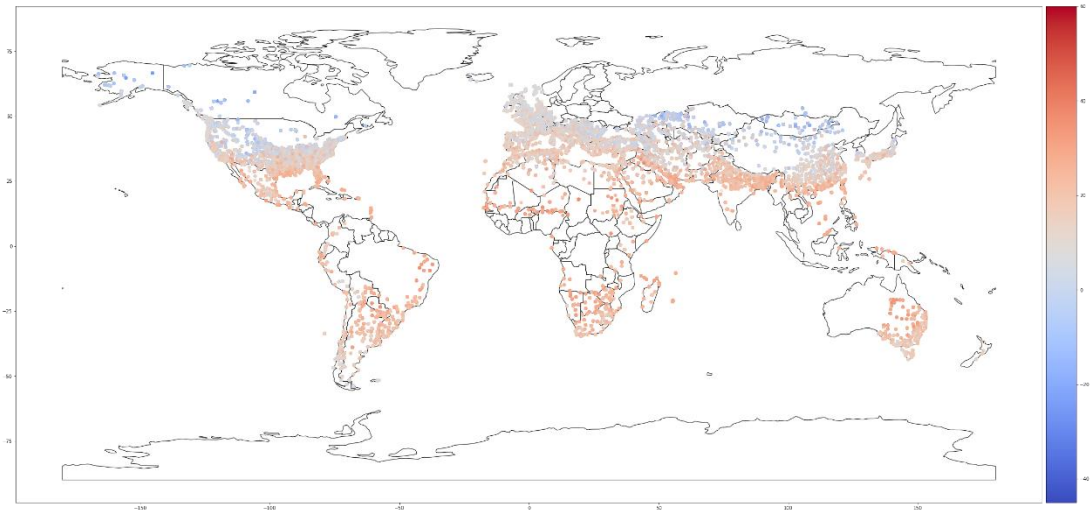

Figure1. Target variables on a world map

Moreover we plotted the distribution of the target variable and it appears to be a Gaussian. It gives the theoretical right to use some machine learning models such as normal linear regression.

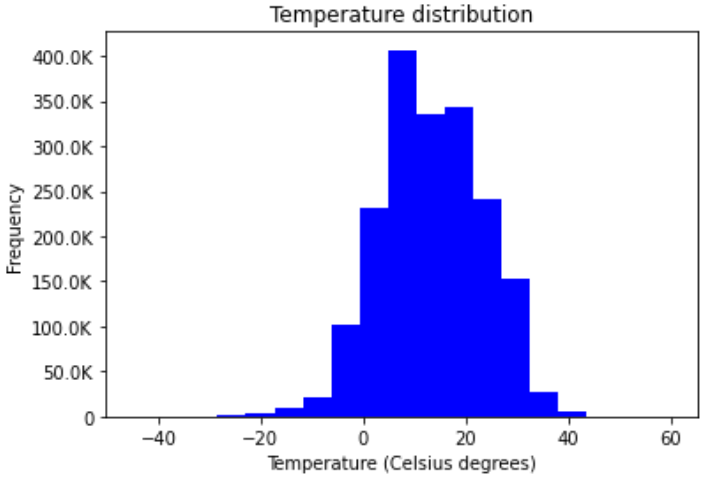

Figure3. The distribution of the target variable.

3) Preprocessing

The preprocessing step is done through a Spark pipeline. The first step is the normalization of the features and it is performed through the StandardScaler method. Then PCA is used.

PCA was almost a mandatory choice due to the huge number of features of the dataset. Our idea was to use enough principal components o reach 90% of the variance. We saw that 32 principal components were enough, however, the notebook could not handle this number of features when training the model. As a result, we decided to use just 5 featuresat accounts for around 50% of the variance.

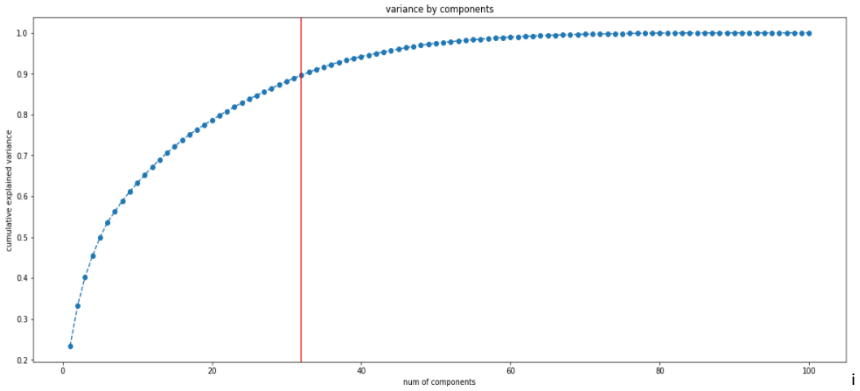

Figure4. Cumulative variance Vs number of components

4) Model selection

The dataset was split into train and test using a 80/20 ratio. The test set has been used only at the end to asses the performance of the best model. The training set has been split into three folds for crossvalidation. The best model has been selected according to the average value of the validation results.

We decided to train five different models. Due to time limitations of Google Colab, from now on results will be based only on 100.000 samples.

i. Linear regression using both ridge regression and lasso, it had a cross validation RMSE of 4.47546
ii. Linear regression using only ridge regression, it had a cross validation RMSE of 4.46286
iii. Linear regression using only lasso, it had a cross validation RMSE of 4.49961
iv. Random forest, it had a cross validation RMSE of 4.92387
v. Gradient boosting, it had a cross validation RMSE of 4.43474

4. Result

The best results were obtained with the gradient boosting model.
We trained it on the entire train set. Its RMSE on the test set was: 4.41304.
We acknowledge the fact that having more time and computing resources would have allowed us to train more models or go deeper into hyperparameter tuning, resulting in a more efficient model with better performance.
However, we are satisfied with the project outcome since we had the opportunity to apply what we have learned during the course.

5. How to run the notebook

The notebook can be uploaded on Google drive and it is ready to run on Colab.
However it is necessary to set up kaggle to download the dataset:

1) Register to the competition https://www.kaggle.com/competitions/eurecom-aml-2022-challenge-1/overview
2) Follow the guide https://www.kaggle.com/general/51898

6. References

[1]. Kaggle data set.[website]. Accessed 20 October 2022.