Big Data Project

Hanciu Raluca-Maria

Cuculescu Andrei

Gr. 411

Introduction

For this project we have chosen a data set from Kaggle, called “Rain in Australia”. This dataset contains about 10 years of daily weather observations from many locations across Australia.

RainTomorrow is the target variable to predict. It means -- did it rain the next day, Yes or No? This column is Yes if the rain for that day was 1mm or more.

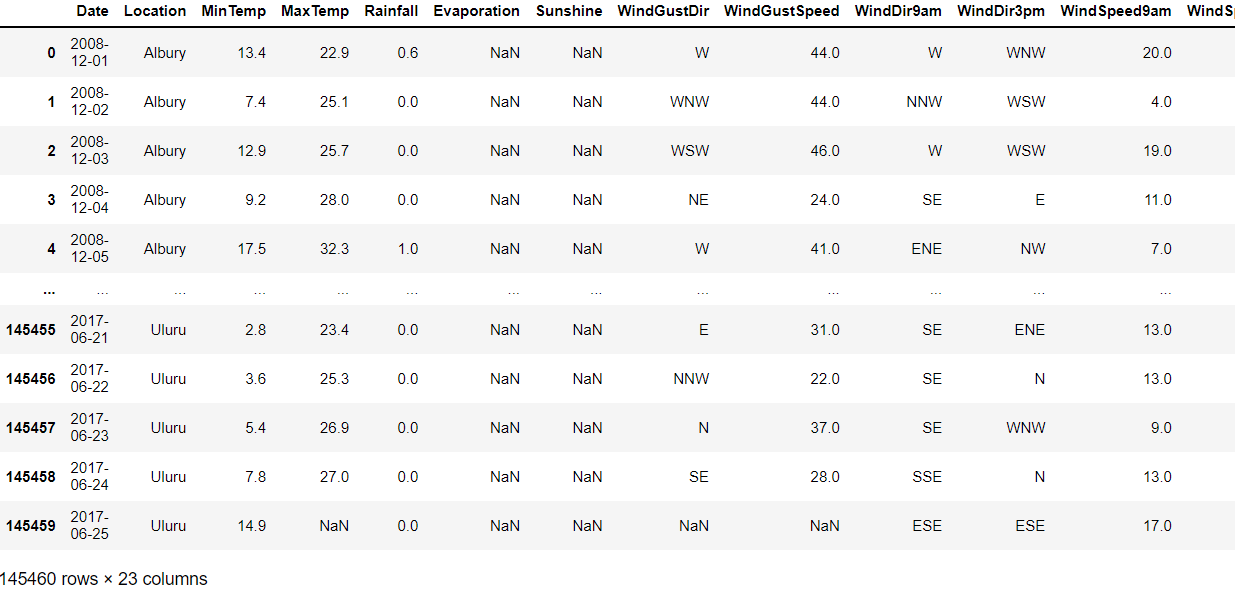
* Number of Features: 23
* Number of samples: 145460

Some of the features from the dataset are:

* DATE - The date of observation
* LOCATION - The common name of the location of the weather station
* MINTEMP - The minimum temperature in degrees celsius
* MAXTEMP - The maximum temperature in degrees celsius
* RAINFALL - The amount of rainfall recorded for the day in mm
* EVAPORATION - The so-called Class A pan evaporation (mm) in the 24 hours to 9am
* SUNSHINE - The number of hours of bright sunshine in the day.
* WINDGUESTDIR - The direction of the strongest wind gust in the 24 hours to midnight
* WINDGUESTSPEED- The speed (km/h) of the strongest wind gust in the 24 hours to midnight
* WINDDIR9AM - Direction of the wind at 9am

We have imported all the necessary libraries and then we began with the data cleaning.

How our initial data looks like :



Data Cleaning

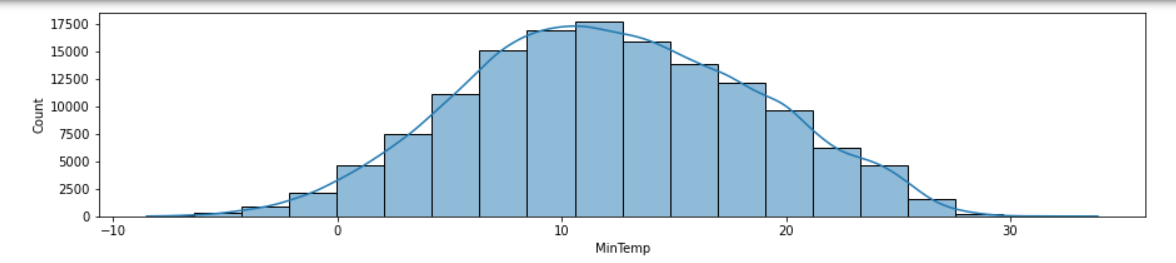
We have checked missing values and deleted features which had more than 15% of data values missing: 'Evaporation', 'Sunshine', 'Cloud9am', 'Cloud3pm’

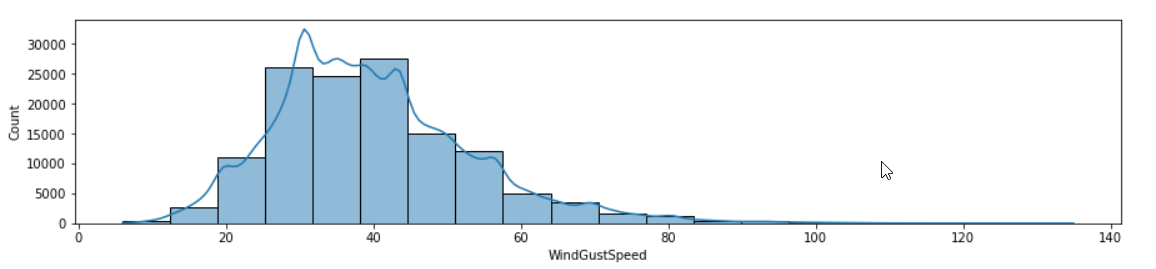
We have also removed rows where target variables are missing

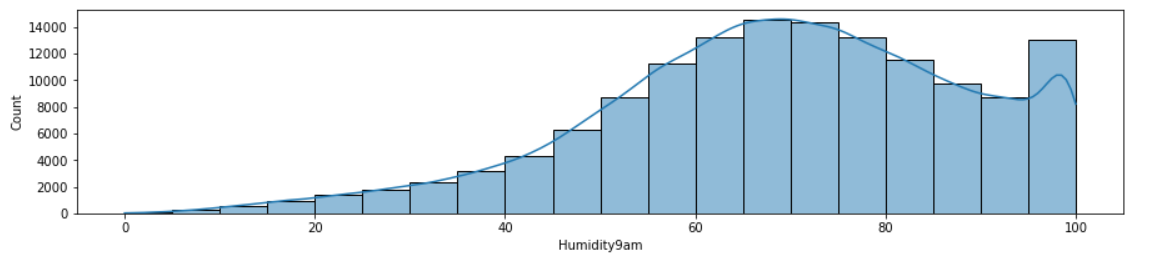
The dataset now contains 140787 samples

Data visualization

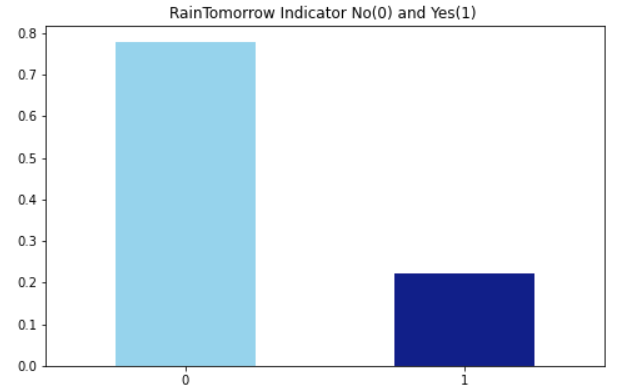
Numerical features distributions:







Target variable (Rain Tomorrow)



Data preparation

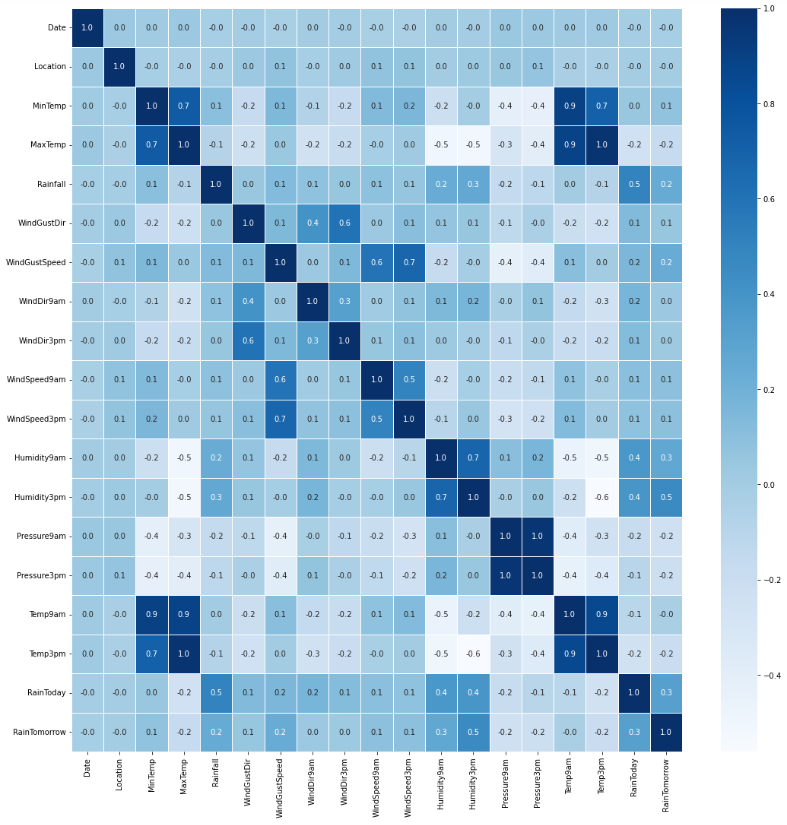
We have changed the categorical values for the feature Rain Today and for the target Rain Tomorrow into 0 and 1, where 0 means no rain and 1 means it rained.

We have used Label Encoder to transform all our categorical data into numerical data.

We have filled in missing values from numerical features with the mean of that feature.

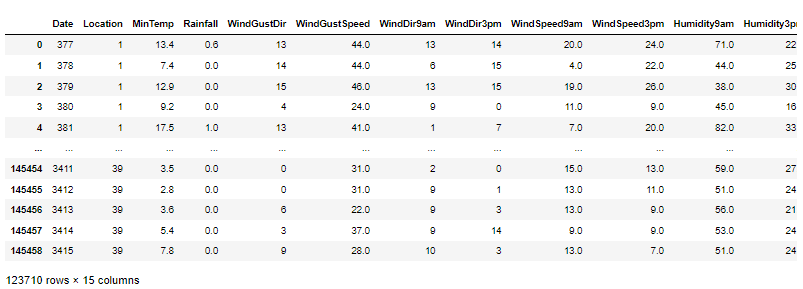
For categorical values we have dropped all samples that have missing values.

We have computed the heatmap:



We have chosen the upper triangle for our heatmap and dropped all features that have a correlation bigger than 0.7, so it won’t influence our model. These features are: 'MaxTemp', 'Pressure3pm', 'Temp9am', 'Temp3pm'.

Now our data looks like below:



We have then used Standard Scaler to standardize features by removing the mean and scaling to unit variance (dividing all the values by the standard deviation). This is especially useful for dimensionality reduction.

Models

For the modelling part we have used the following 4 models:

* Logistic Regression
* Random Forest Classifier
* Decision Trees
* Ada Boost Classifier

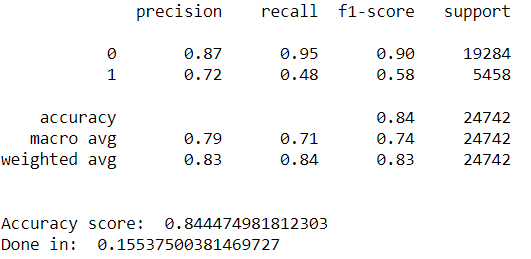
For dimensionality reduction in combination with the upper models we have used:

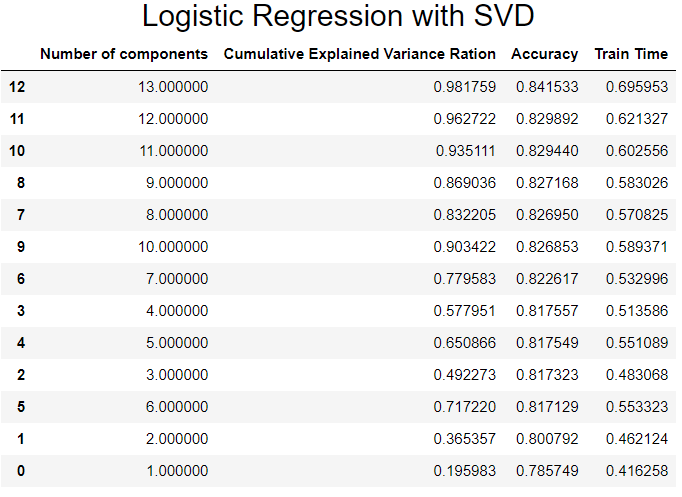
* **PCA**
* **SVD**
* Random Forest Built-in **Feature Importance**

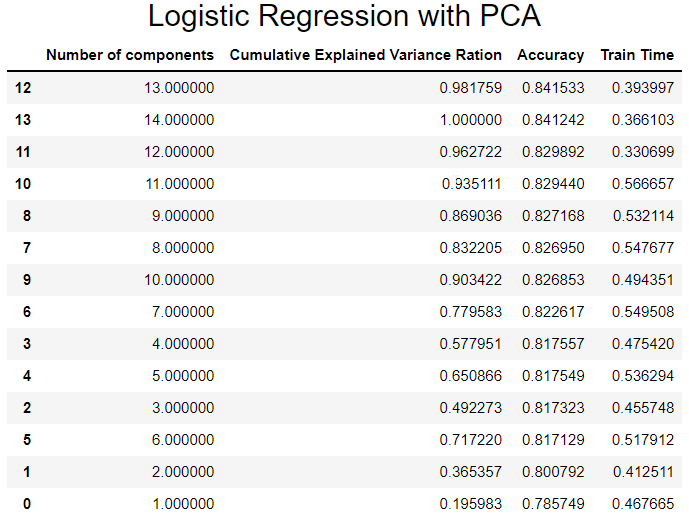
We have compared the models to see which performs best.

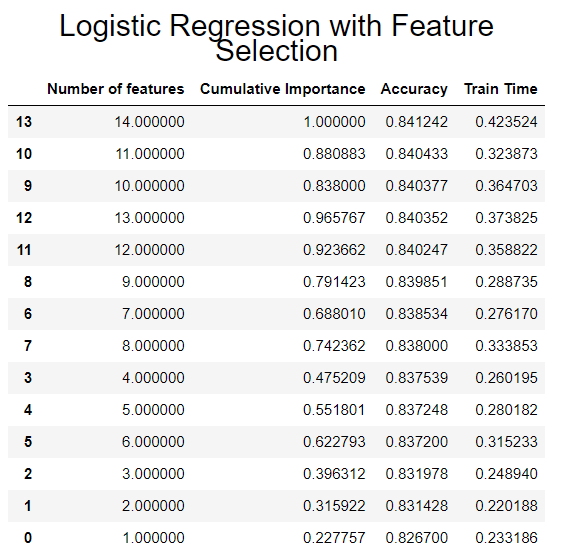
1. Logistic Regression

No dimensionality reduction





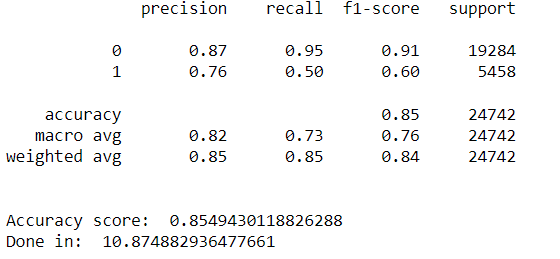


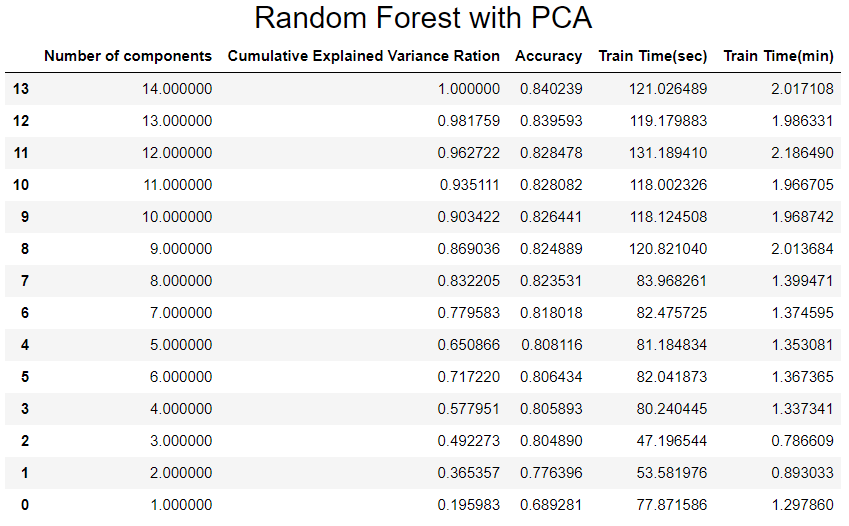


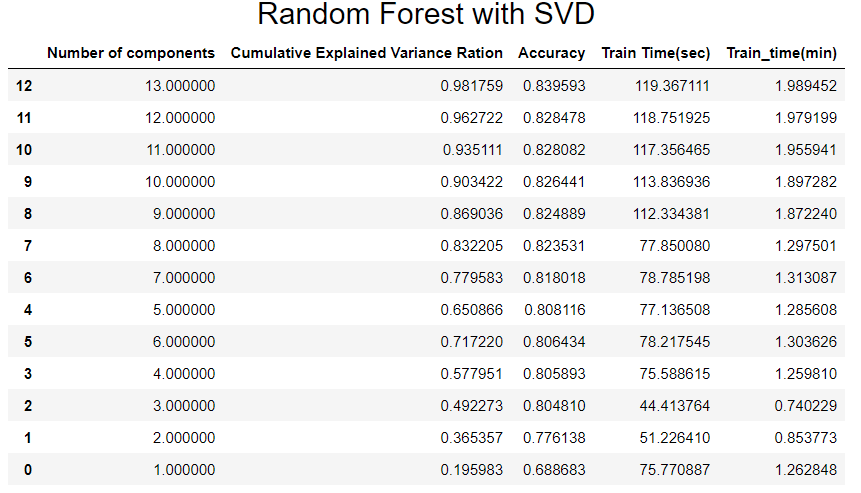
With logistic regression performed best the model without any dimensionality reduction. However, the differences are quite small between the results with/ without dimensionality reduction. Also, the train time for the model without dim red was better.

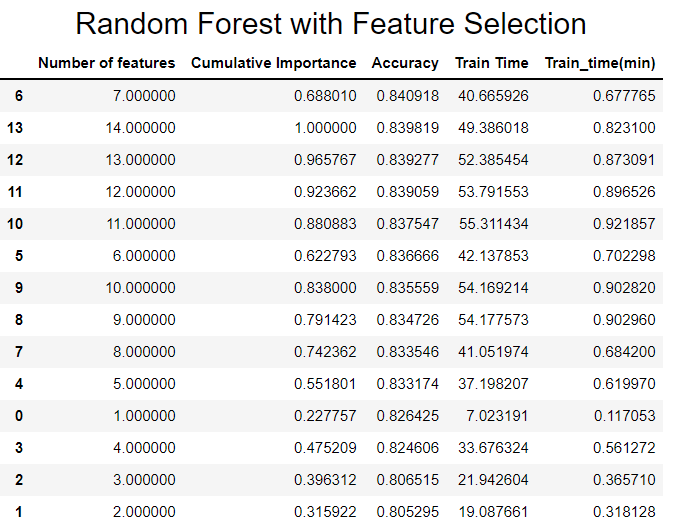
1. Random Forest Classifier

No dimensionality reduction





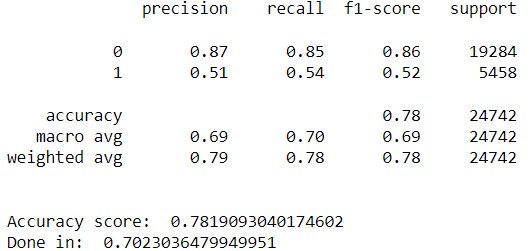


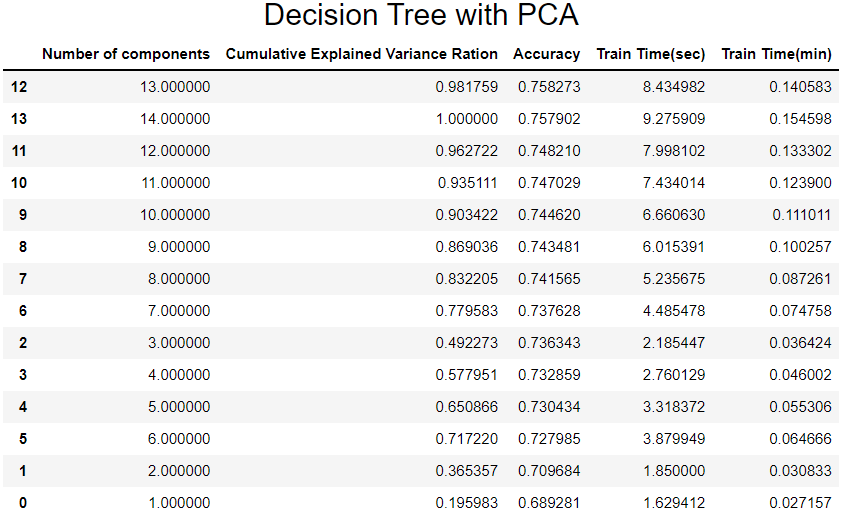


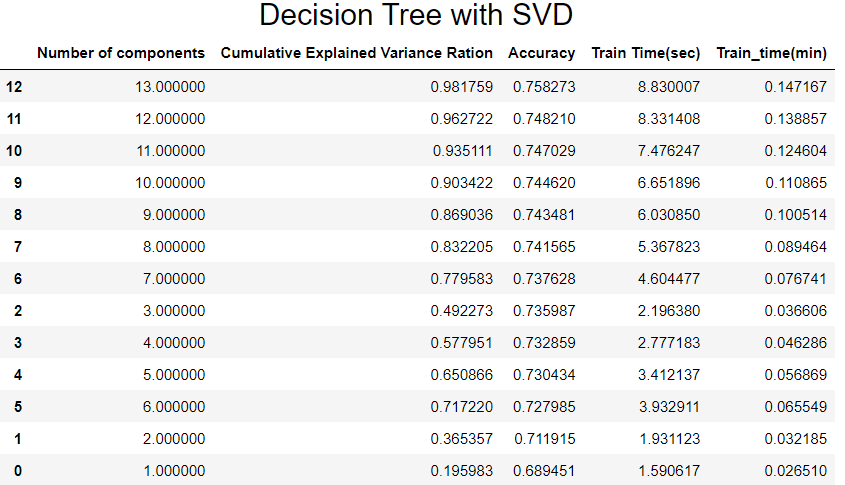
All dimension reduction models performed way worse than the model with no dimension reduction, especially when speaking about train time.

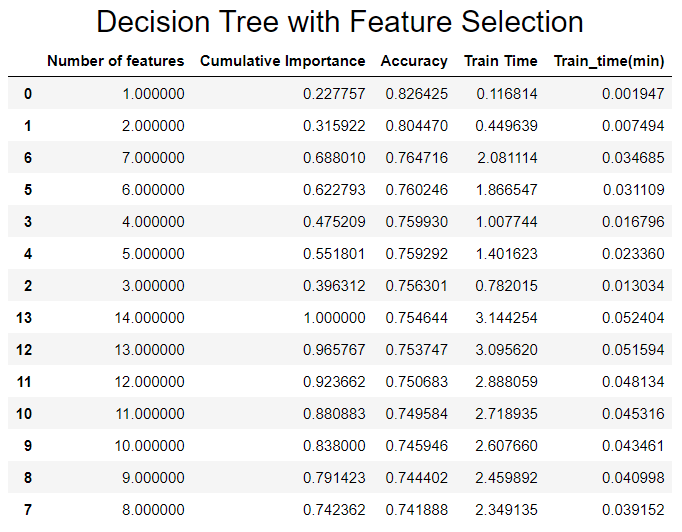
1. Decision Tree

No dimensionality reduction



****

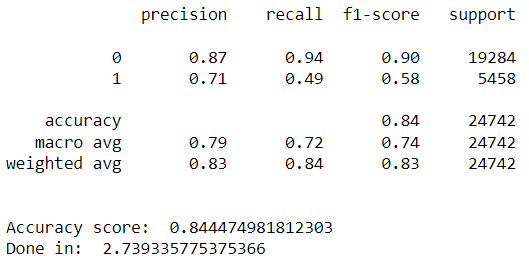
****

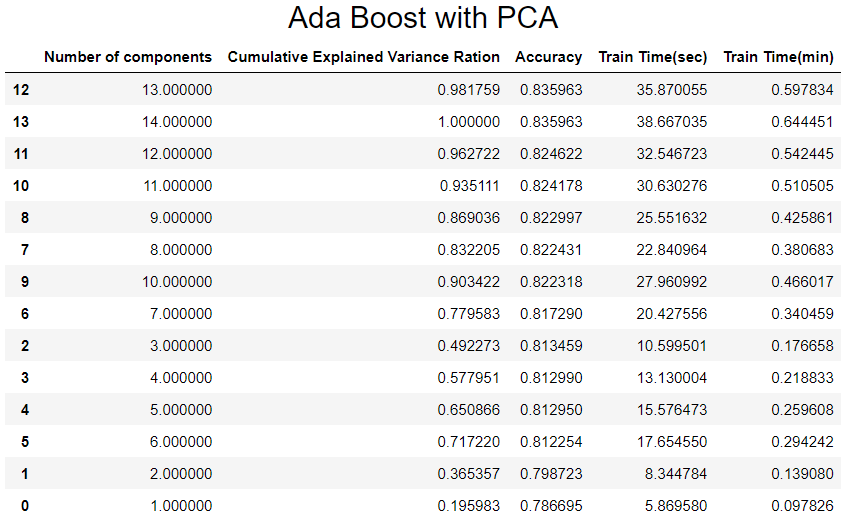
****

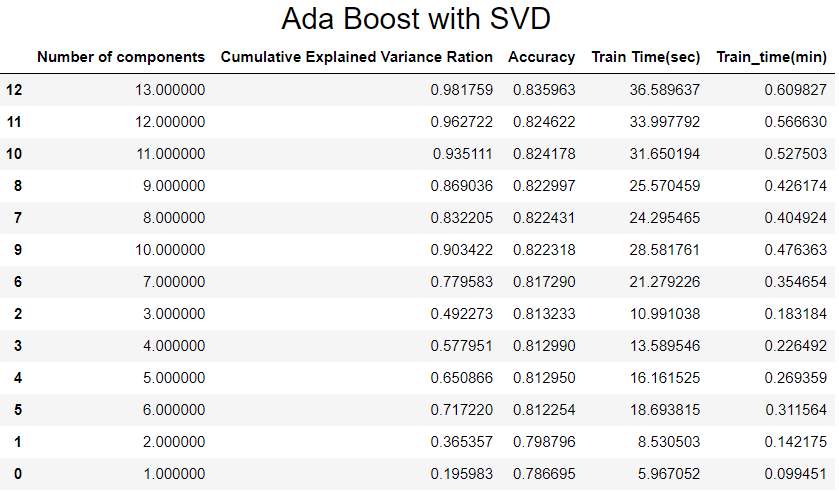
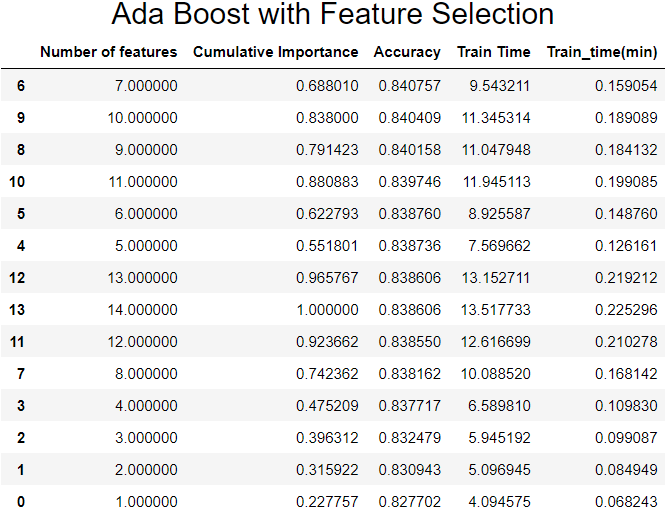
Decision Tree with feature selection has the best accuracy and the best time using only **one** component.

1. Ada Boost Classifier

No dimensionality reduction







Again our best model is Ada Boost with no dimension reduction.

Conclusions:

The best accuracy had Random Forest with no dimensionality reduction: 85,4% in 10 sec.

Feature selection using Decisional Tree had an accuracy of 82,6% in 0,11 sec, with only one feature.