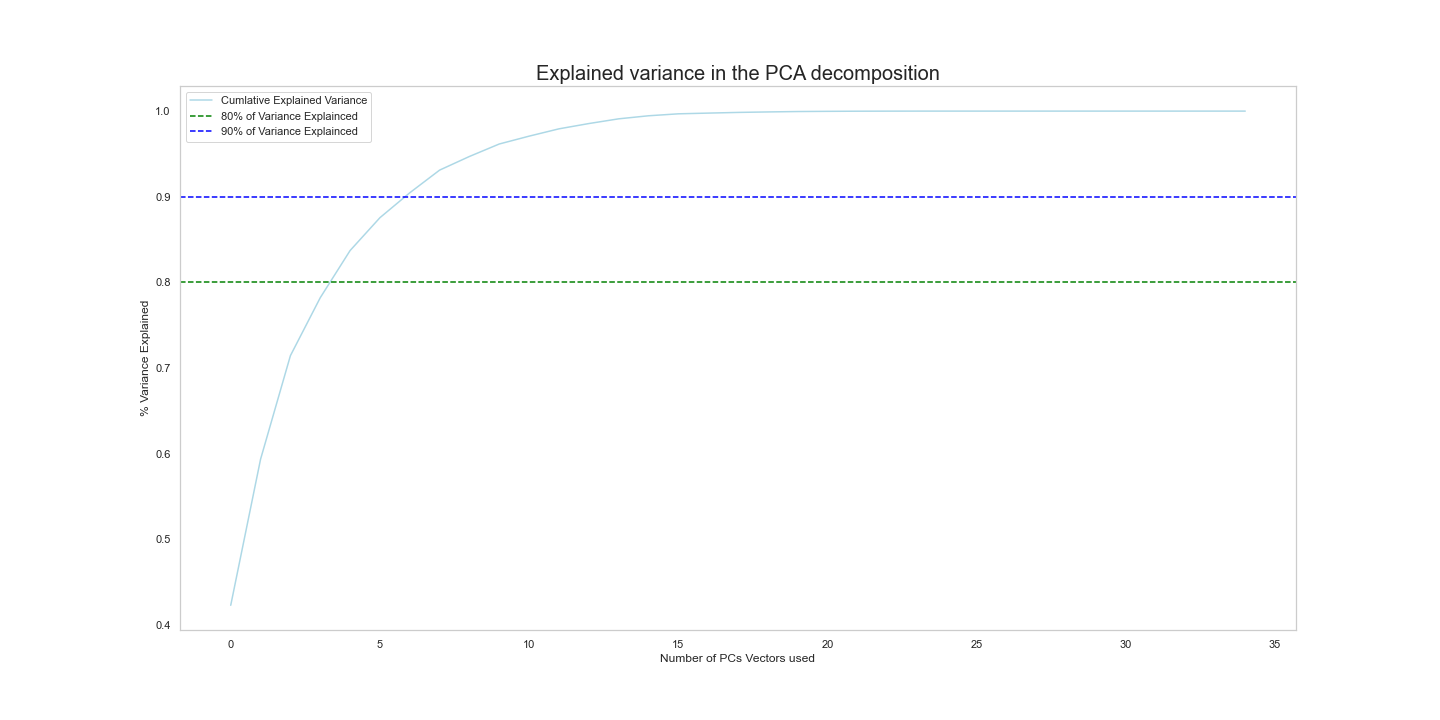
The first step of our modelling part was to scale our data. Indeed, the various different scales present in our dataframe would make a very difficult interpretation and scoring for our different predictions. We chose to standardize our data because the different continous features might have a shifted Gaussian Distribution (letting go of negative values).

On top of this standardized data, we computed the PCA decomposition of our dataframe. There were many reasons for PCA to work well with our dataframe:

* This would handle the high correlation among our predictors
* It would enable us to work with less features and have a better training of our model

Therefore, according to the model we use, we expect our models to perform well when they use between **3** and **10** Principal Component Vectors.

Therefore, we had two dataframes to play with: the scaled features and the PCA vectors.

* The models we have used are: Linear Regression, Lasso Regression, Regression Trees, Random Forest Regression and AdaBoost Regression.
* The scoring metrics we used is the Negative Mean Square Error
* We performed Hyper-Parameter tuning + Model selection via Cross-Validation. The Cross validation strategy was 9-fold Cross Validation. Please see the appendix for an exhaustive list of the Hyperparameters we tuned for every model, and their range of values.

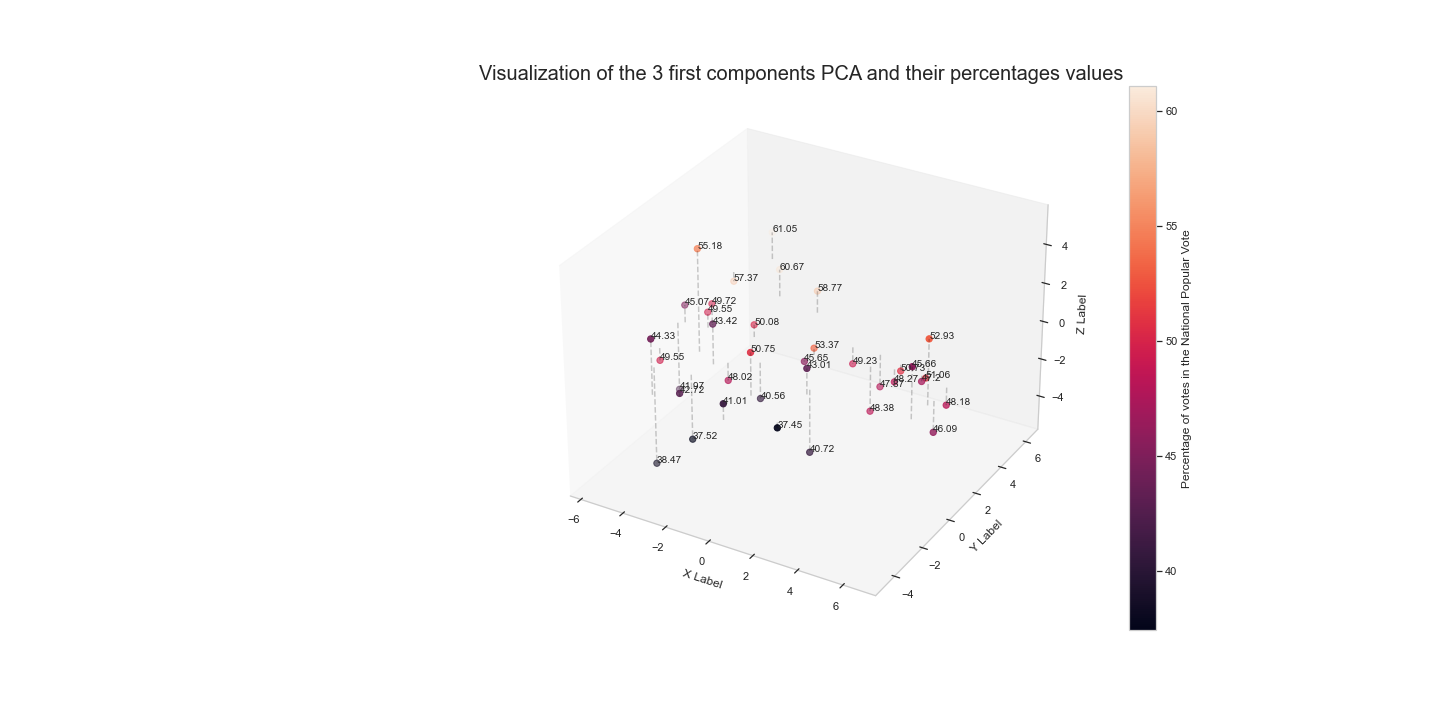
A priori, we expect the different models to perform very differently.

* We expect Linear Regression on scaled data to behave poorly because of the correlated features: the Gram matrix will not be invertible.
* We expect Tree models to perform poorly because they are to be fit on the whole features and we only have a few training points: high variance of the tree predictions
* We expect Random Forest to have a considerable increase in performances when compared with trees. There are several reasons for that. First, they grow trees with a small amount of features: less variance per tree, and then they average the predictions over trees: less resulting variance
* We expect penalized models to perform well: Lasso on scaled features and PCA vectors should perform well.

In order to see the complete results of our models + Hyperparameter tuning, please refer to the Appendix. The model that performed best according to cross-validation was:

* Lasso Regression on 3 PCA vectors with a regularization coefficient C = 0.1. The CV MSE was 7.535. The resulting MLE weights parameters of the Model are [-0.118, 1.857, 1.377]. The CV R2 score is 0.7268.

Let us visualize, for the sake of interpretation, the distribution of the repartition of the votes against the first 3 PC vectors.



As we can see, we have succesfully captured a pattern in predicting the national popular votes (increasing from bottom right to far left). This is what is captured by the weights of this model: negat(ive in the first component (X) and positive in the two others.

Therefore, this is the model we will be using in order to make the predictions for the National Popular Vote.

We computed the final prediction for the National Popular Vote and computed CI via Bootstrapping for this method. The final results is :

* For Donald Trump: NPV = 46.525 +/- 1.39
* For Jobe Biden: NPV = 52.46 +/- 1.88

*Note: The final actual result (47.3 for D. Trump, 51 for J. Biden* *falls right into our Confidence Interval).*