Using Logistic Regression to predict whether a USA citizen is going to be the president of the USA or not

Introduction

It's insane how much power can one hold. It doesn't matter were you live, the effects of the USA president are seen worldwide. Would world be more prepared for the new president if we would predict the future USA presidents? This Machine Learning Project is focusing in the domains of politics and discussing the fact that can we predict if a specific USA citizen is going to be the president of the USA in some point one's life. This project is implemented on a course's framework scale and provided for learning experience. In this report we are going to familiarise ourself with this ML problem and discussing the results given. In following sections "*Problem Formulation*" and "*Method*" the projects ML problem is formalised and all the details about the data are given. In addition the used hypothesis space and loss functions are presented. In last sections "*Results*" and "*Conclusion*" the given errors are compared and suggestions for future improvements are discussed.

Problem Formulation

This ML problem's application can be modelled with datapoints representing individuals who are in the moment USA citizens. Each individual citizen is characterised by features witch are one's birth day, birth month and education level. The label (quantity of interest) is the fact that is the individual going to be the president of the USA or not. This is clearly a Logistic Regression problem.

The gathered datapoints with known labels are collected from Wikipedia pages of all the former USA presidents (https://en.wikipedia.org/wiki/List_of_presidents_of_the_United_States) and the currents senators of USA (https://en.wikipedia.org/wiki/List_of_current_United_States_senators). The former USA presidents are representing datapoints with labels “yes” and the current senators are representing datapoints with labels “no”. On a course’s framework scale, datapoints representing normal USA citizen are not available so senators are filling this spot. Affects of following will be discussed in the section “*Conclusion*”.

Using these sources we gather up two different datasets “*dataset1”* and “*dataset2”.* The *dataset1* consists of 70 datapoints witch we are using to train and validate data and then choosing the model based on the results. *Dataset2* is used to test the chosen model, witch was trained with *dataset1.* With this test data, *dataset2*, we are assessing the quality of the model chosen. *Dataset2* consists of 50 datapoints witch are not involved or used in *dataset1*.

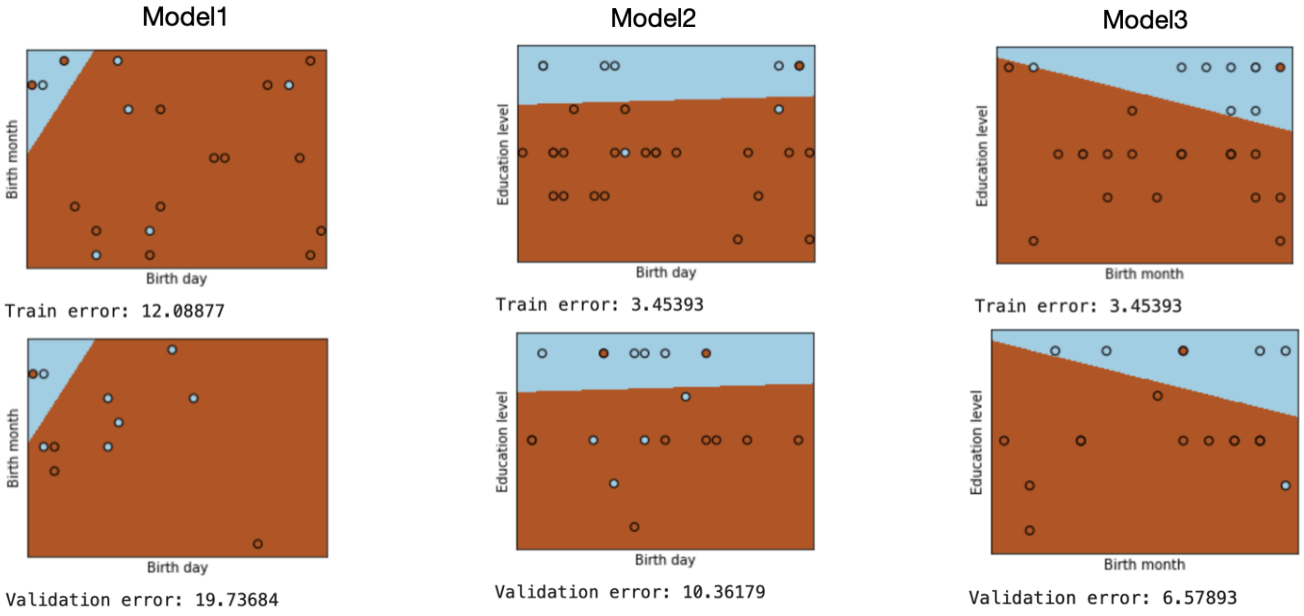
Method

According to the two categories the application has, which are encoded by a label, “*yes, one is going to bee the USA president*” and “*no, one isn’t going to bee the USA president*” we are clearly on the brink of a Logistic Regression problem with to classes. Therefore, we chose Logistic Regression. Logistic regression learns a predictor out of the hypothesis space. We measure the quality of a hypothesis $h(x)$ via the logistic loss $log(1 - exp(-yh(x)))$ incurred when predicting the label y of a datapoint by $h(x)$. Logistic loss is a differentiable loss function that is useful for classification problems and it depends smoothly on h such that we could define a derivative of the loss with respect to h [Ch. 2.3, MLBook]. In addition, with dataset2 we are using the 0/1-loss when assessing the quality of the model chosen to provide more accurate and variant results with different loss function.

Our hypothesis space consists of thee different models withs are all Logistic Regression models characterised with different features. *Model1* uses features as birth day and birth month, *Model2* uses features as birth day and education

level and *Model3* uses features as birth month and education level.

For finding the best result we are first randomly shuffling the data in *dataset1* and then single splitting it in training data (60%) and validation data (40%) with a random split. All the steps discussed in the report are implemented using Python. Below we are having first the plots of the classes and then a table (Table1) for train and validation errors. Blue class represents the non-presidents and orange the presidents.



	Model1	Model2	Model3
Train error	12.08877	3.45393	3.45393
Validation error	19.73684	10.36179	6.57893

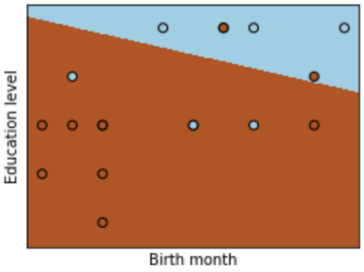
(Table1. Train and validation errors. The model with smallest validation error is highlighted)

Results

According to Table1, we chose the *Model3* which resulted in the smallest validation error. It seems that for the other models *Model1* and *Model2,* the ML method doesn't overfit but it does lack correlation between the used features and label. For *Model3*, the training and validation errors are both quite big, but

there is still notably more correlation. To improve the method we would need to consider more valuable information containing features and more consistent and real-life bounded data.

To assess the performance of the resulting hypothesis (*Model3*) we compute its logistic- and 0/1 loss on the test set “dataset2”. The resulting test errors are 11.513 and 0.33, which are significantly larger than the validation error in *Model3* (6.58).




```
Test set error: 11.51310
Test set error in 0/1: 0.33333
```

Conclusion

In this report we have studied three different models for learning a hypothesis to predict if a random USA citizen is going to be the president of USA or not. These three different models are obtained from using Logistic Regression with different labels. We chose the *Model3* that resulted the smallest validation error 6.58. The corresponding training error was 3.45 which is smaller. In this model these is not overfitting happening.

The finally chosen hypothesis, *Model3* with smallest training error, has been tested on the test set *dataset2*. Datapoints in *dataset2* are not included in dataset1 and therefore are discrete from dataset1. In contrast, the training and validation of the three considered models were based on dataset1. The test error and the validation errors are both large.

It’s obvious that there is much to improve. In order to have more successful model we would need to gather much better data including data of ‘normal’ USA citizens and not just senators, because senators are yes more likely to be more highly educated than average USA citizen. In addition, we would need to consider more valuable features. We can clearly see that birth day or month are both too random features and have no correlation between the label. We would too think about using different more roboust model (e.g decision trees) to come up with better results. All in all it’s a really interesting topic but maybe not resolvable with this much straight forward model.

References

[MLBook] A. Jung, “Machine Learning. The Basics”, 2021, mlbook.cs.aalto.fi

[Wikipedia] https://en.wikipedia.org/wiki/List_of_presidents_of_the_United_States

[Wikipedia] https://en.wikipedia.org/wiki/List_of_current_United_States_senators

[Scikit-learn] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.zero_one_loss.html

[Scikit-learn] https://scikit-learn.org/stable/auto_examples/linear_model/plot_iris_logistic.html#sphx-glr-auto-examples-linear-model-plot-iris-logistic-py

[Scikit-learn] https://scikit-learn.org/stable/modules/cross_validation.html