Introduction

Many of the problems that we as a sociality face on a daily basis are qualitative as opposed to quantitative. We often are interested in classifying some incoming data into members of a known set. For example, a doctor may be interested in classifying a tumor image as either a benign or malignant. He will attempt to classify the image given a set of features such as size, location, color, etc. However, in the real world it is often very difficult, if not impossible, to come up with a comprehensive list of criteria for classification problems such as this one. Thankfully, we can now use the power of computation run algorithms that will receive as input a training set consisting of features and labels, and it will output a list of meaningful criteria. Thus, we can now efficiently classify a new set of data and output the most-likely labels for that set (e.g. benign or malignant).

This paper explores techniques in supervised learning in order to classify two sets of data:

1. Emails from a dataset into spam or not spam
2. A person political affiliation into democrat or republican given their twitter feeds

The relevant of such problems in the context of a digitalized world are obvious. Given the raise in digital marketing, the necessity for effective spam filters can be illustrated by the fact that virtually every email provider and social network deploys one in their service. Otherwise consumers would be bombarded by spam advertisement, and most likely leave the service. Data analyses of social networks in order to determinate the political affiliation of the user has been showing to be an extremely effective strategy for politicians to better understand and target their bases. What may not be obvious, however, is that fact that these two apparently unrelated problems can, in fact, be abstracted as a single dictionary classification problem.

In this paper I will analyze the performance of five Supervised Learning algorithms (Decision Trees, Neural Networks, Boosting, Support Vector Machines, k-Nearest Neighbors) that were implemented in python in order to classify both datasets.

Data collection

The data used in this project for training and testing proposes came from two different sources.

The dataset used in order to classify spams was download from the GitHub repository (github.com/SouravJohar) and can also be found in the attachment. The dataset is a collection of 5172 emails, equally divided and labelled as either spam or not spam (ham).

The dataset used in order to classify individuals as democrats or republicans was collected by me using twitters API and can also be found in the attachment. The dataset consists of 559 text files. Every text file is a compilation of the 200 most recent tweets from all active members on Twitter from congress, senate, and the president himself. The data was collected on February 8th, and can be found in the attachments alongside with the python script used for this purpose.

- **Decision Trees**

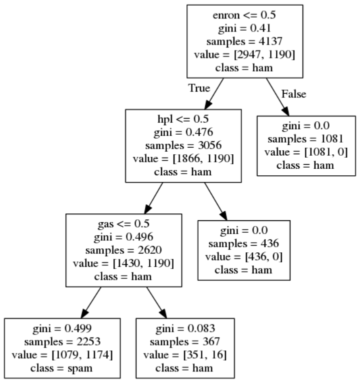
Decision trees are a simple model of a supervised learning algorithm that attempts to predict the label of a target variables by learning decision rules inferred form data features. Decision are simple to visualize and therefore to interpret. With minimal data preparation it is possible to build accurate models to resolve classification problems. However, if not pruned correctly, decision trees can on yield on over-complex models that tend to not generalize well. In order to perform the splits on our dataset we used cost function referred as Gini Index that ranges from 0 (certain) to 1 (uncertainty).

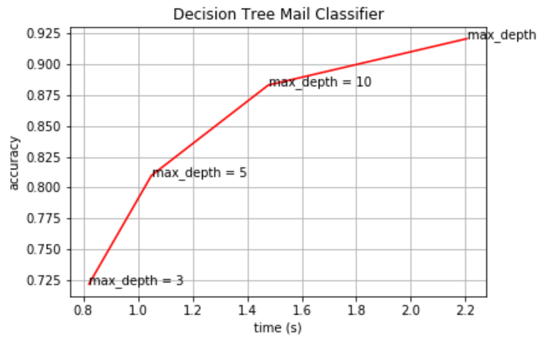
Mail Classification

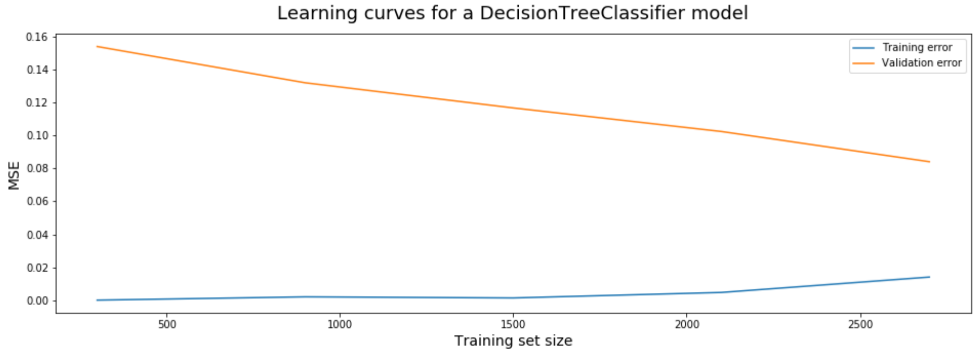
On the top left of the image bellow we see the representation of a Decision Tree with max depth = 3. Notice "enron" is the most meaningful feature on the classification. Just as expected, the Gini Index declined as we transverse the tree to its leafs.

On the top right of the image bellow we see a plot of maximum tree height as a function of accuracy and time. The more nodes we allow the model to have, to more complex it is, and therefore the more accurate it gets. However, overly fitted models are non-generalizable. Therefore, observing the trend in the data it is reasonable to assume that the optimal height for this model is around 10 nodes.

On the bottom of the image we can see the plot of the learning curve of this algorithm. We see that the validation error is decreasing much more raptly than the training error is increasing. Therefore, adding more training data would be beneficial to this model.





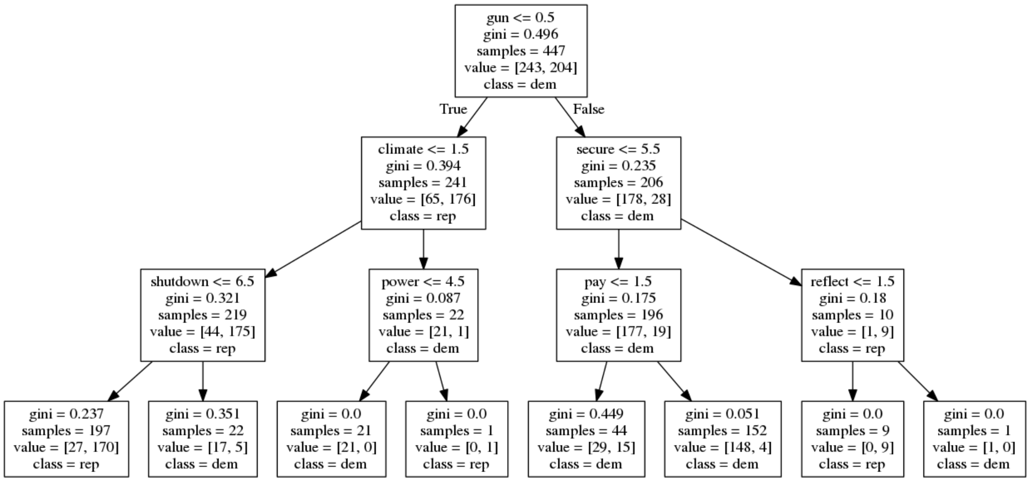


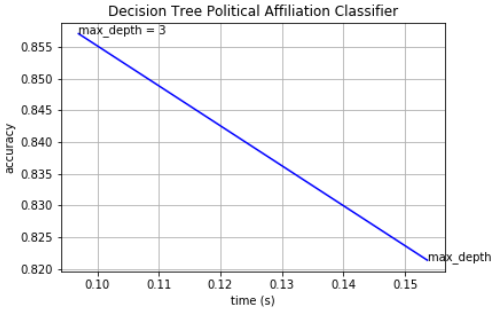
Political Affiliation Classification

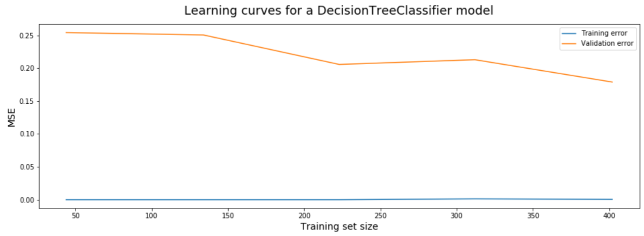
On the top image bellow we see the representation of a Decision Tree with max depth = 3. Notice “gun” is the most meaningful feature on the classification. Gini Index declined as we transverse the tree to its leafs.

On the bottom left of the image bellow we see a plot of maximum tree height as a function of accuracy and time. We notice that this model behaves differently from the mail spam model in the sense that adding nodes decreases the accuracy of the classifier.

On the bottom right image we can see the plot of the learning curve of this algorithm. We see that the validation error is decreasing much more raptly than the training error is increasing. Therefore, adding more training data would be beneficial to this model.







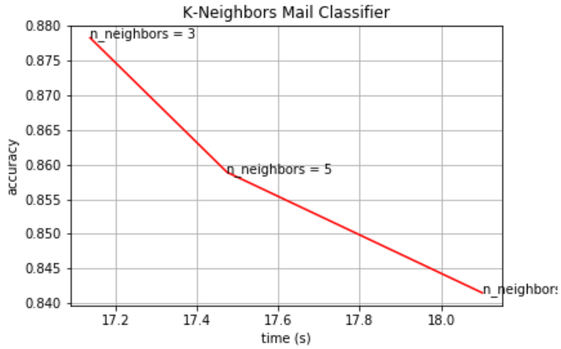
- ## Nearest Neighbors

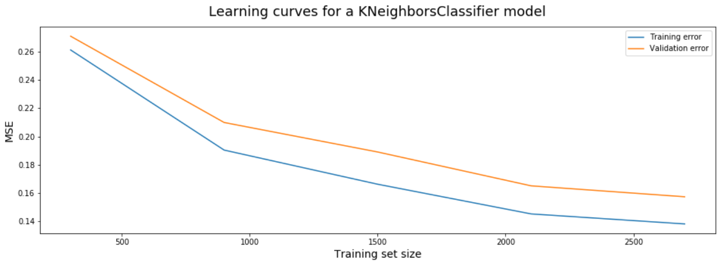
Nearest Neighbors is a supervised learning algorithm that attempts to predict the label of a target variables by predefining the number of training samples (input k) closest in distance to the target, and predict the label from these.

Mail Classification

On the left of the image bellow we see a plot of the number of k-neighbors used to calculate the target label as a function of accuracy and time. We see an overfitting behavior in which more neighbors detriment the accuracy of the algorithm. Therefore, observing the trend in the data it is reasonable to assume that the optimal number of k for this model is around 3.

On the right of the image we can see the plot of the learning curve of this algorithm. We see that the validation error and the training error are decreasing in a similar rate, and therefore the addition of more training data would be unlikely to improve thee model.

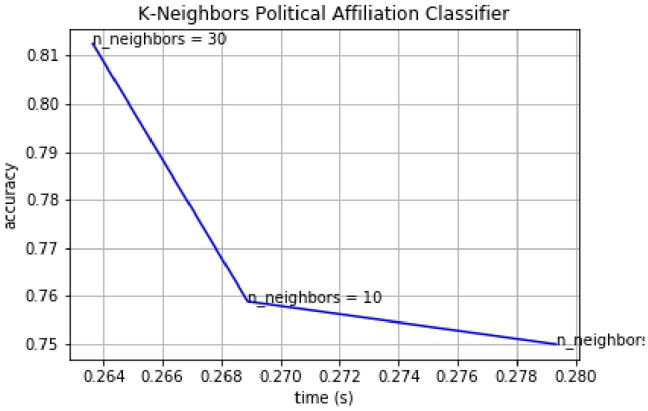


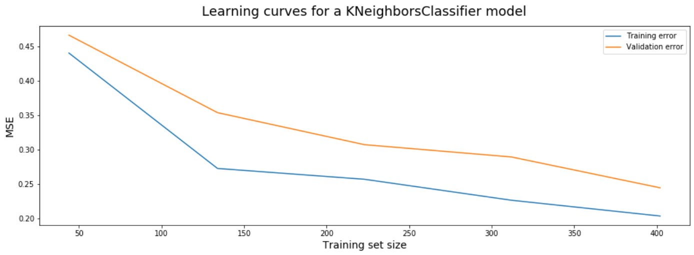


Political Affiliation Classification

On the left of the image bellow we see a plot of the number of k-neighbors used to calculate the target label as a function of accuracy and time. In this case we need many more neighbors than in the mail case in order to get an accurate label. More specifically, only when around 30 neighbors the label predictor starts getting reasonably reliable.

On the right of the image we can see the plot of the learning curve of this algorithm. We see that the validation error and the training error are decreasing in a similar rate, and therefore the addition of more training data would be unlikely to improve thee model.





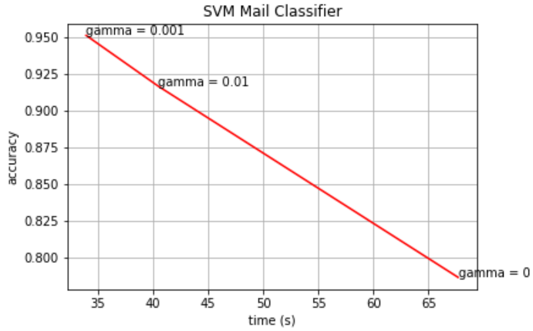
- **Support Vector Machines**

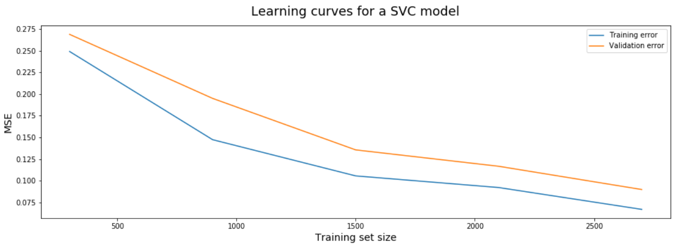
Support vector machines (SVMs) is a supervised learning algorithm that attempts to predict the label of a target variables by separating the data across a decision boundary. In our implementation we use the model SVC from the class SVM and tune the parameter gamma. Gamma is used as similarity measure between two points. A small gamma value define a function with a large variance.

Mail Classification

On the left of the image bellow we see a plot of the number of SVM used to calculate the target label as a function of accuracy and time. Because this mail data has a lot of variance within it, we observe a better fitted model with smaller gammas.

On the right of the image we can see the plot of the learning curve of this algorithm. We see that the validation error and the training error are decreasing in a similar rate, and therefore the addition of more training data would be unlikely to improve thee model.

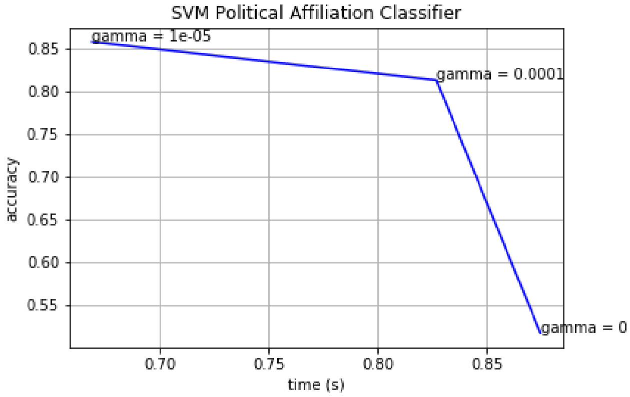


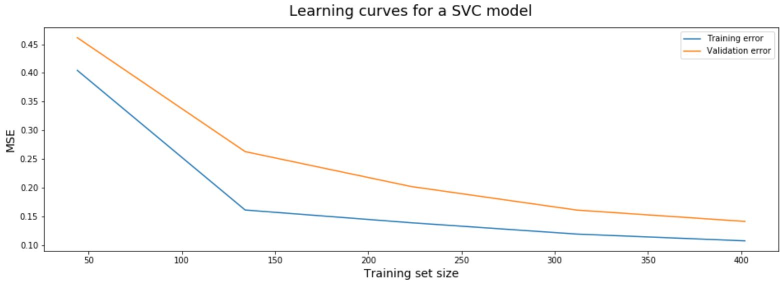


Political Affiliation Classification

On the left of the image bellow we see a plot of the number of SVM used to calculate the target label as a function of accuracy and time. In the case of political affiliation classification, we notice that gamma has to be significantly smaller comparted to the previous dataset in order to yield meaningful predictions

On the right of the image we can see the plot of the learning curve of this algorithm. We see that the validation error and the training error are decreasing in a similar rate, and therefore the addition of more training data would be unlikely to improve thee model.





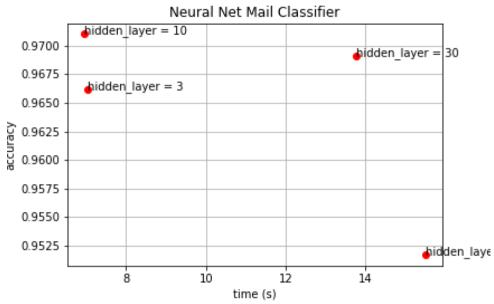
- **Neural Networks**

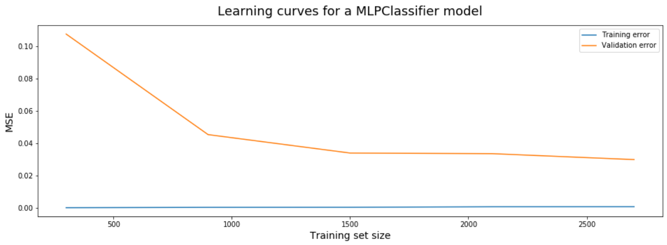
Neural Networks are a supervised learning algorithm that attempts to predict the label of a target variables by assigning a weight to each node of the network. In our implementation we use the model MLPClassifier from the class Neural Networks and tune the number of allowed internal nodes. The more internal nodes, the more complex the model is, and therefore the more likely to over-fit data; although this specific method implementation has safe-nets to prevent this.

Mail Classification

On the left of the image bellow we see a plot of the number of internal nodes used to calculate the target label as a function of accuracy and time. It is interesting to notice that the data fluctuates in such way that it is impossible to nicely fit the points into a line. We notice that 3 nodes was already enough to make an accurate model, and adding more nodes does not better the model.

On the right of the image we can see the plot of the learning curve of this algorithm. We see that the validation error is decreasing faster than the training error, and therefore more training data would benefit the model.

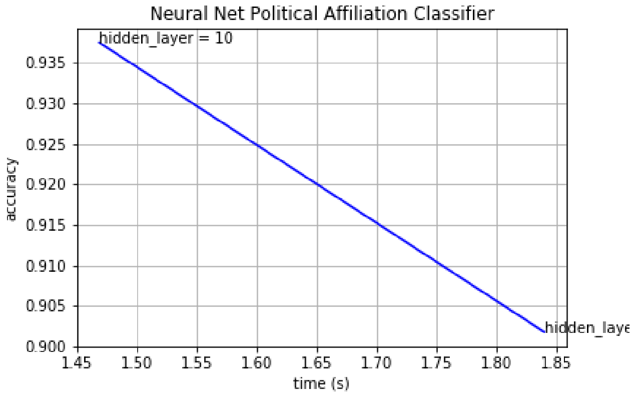


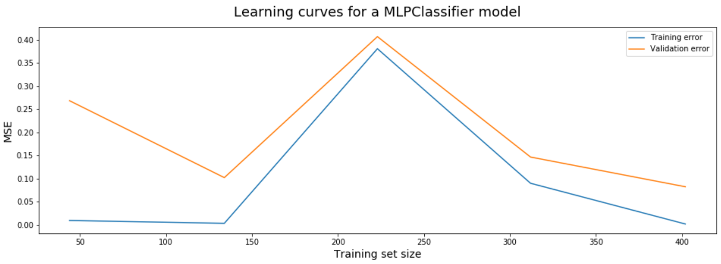


Political Affiliation Classification

On the left of the image bellow we see a plot of the number of internal nodes used to calculate the target label as a function of accuracy and time. We see a linear trend between 3 and 10 layers hitting to the fact that this particular set of data needs more complexity than the spam classifier.

On the right of the image we can see the plot of the learning curve of this algorithm. We see that the validation error and training error follow each other's trends very closely. Therefore, it is unlikely that adding more training data will improve this particular classifier.





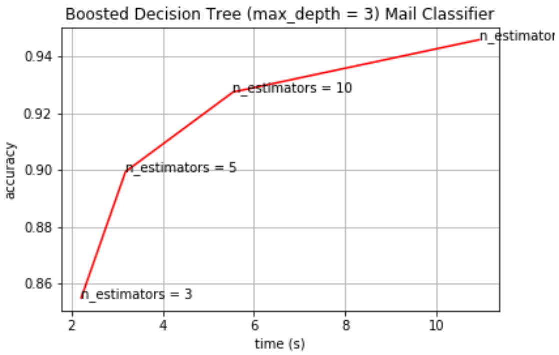
- **Boosting**

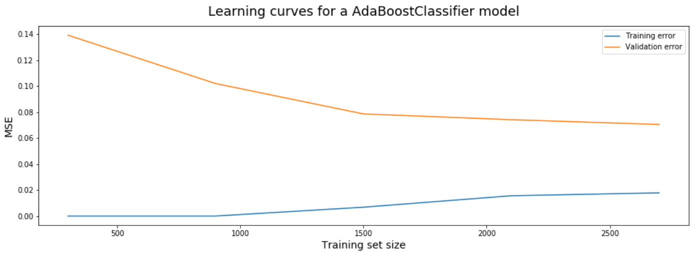
Boosting generally refers to a technic in which a dataset is modeled by a generic classifier, and subsequently we add additional copies of the classifier on the same dataset correcting for inaccurate weights. In our implementation we use the model of a Decision Tree and tune apply Adaboosting on it by tuning the maximum allowed number of estimators.

Mail Classification

On left image bellow we see a plot of maximum number of estimators as a function of accuracy and time. The more estimators we allow the model to have, to more complex it is, and therefore the more accurate it gets. Observing the trend in the data it is reasonable to assume that the optimal number of estimators for this model is around 10.

On the right image we can see the plot of the learning curve of this algorithm. We see that the validation error and the training error are tending to converge, but on a very slow rate. Adding more training data would be beneficial to this model.

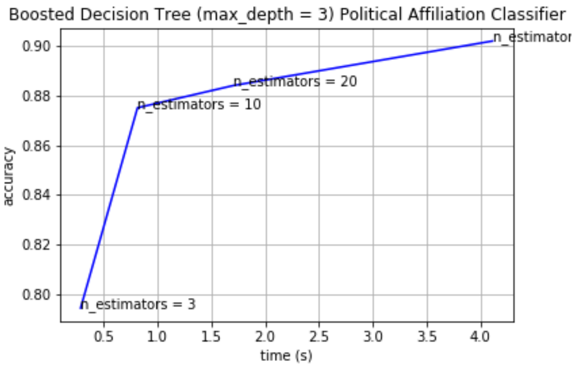


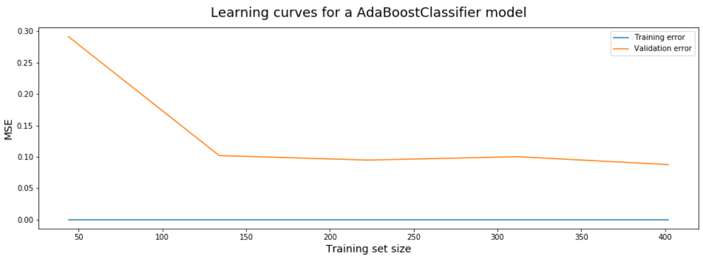


Political Affiliation Classification

On left image bellow we see a plot of maximum number of estimators as a function of accuracy and time. We see a similar behavior here and in the mail dataset. More estimators yield on a more reliable but more complex model. Again, it seems that 10 is the ideal number of estimators.

On the right image we can see the plot of the learning curve of this algorithm. We see that the validation error is trending downwards, but the testing error remains almost unchanged. In this case, more testing data is definitely needed in order to benefit the model predictions.





- **Analyzes on the choice of a classifier**

At first, when implementing the algorithms as they were described on scikit-learn website, the model predictions were around 60%, and therefore not very reliable. After some research and the necessary pruning and tuning, all models were able to achieve a prediction rate greater than 85%.

It because very apparent that even though both classification problems could be similarly model by a vectored alphabet, classifying spam emails turned out to be (not surprisingly) much easier than classifying peoples political affiliation. We could only achieve satisfactory results on the political classification after building much more complex models. After analyzing the Learning Curves for the algorithms, it became evident that we would need more training data in order to better the political classification. It turns out that the training set was relatively small because I inly used tweeter feeds from actual elect politicians, so that data remained reliable. A possible way of improvement the dataset is to somehow get the list of registered democratic and republican voters, and try to query their tweets on the database. This would allow for a much richer and brooder dictionary.

From all the tested models, Neural Networks performed the best for both sets, reaching a successful classification rate of above 97% with 10 layers on the mail dataset, and above 95% also with 10 layers on the political affiliation dataset.