CS 4780/5780: Machine Learning for Intelligent Systems **(Due: 12/15/20)**

Final Project: Election Result Prediction for US Counties

Instructor: Nika Haghtalab, Thorsten Joachims

Course Policy: Read all the instructions below carefully before you start working on the final project, and before you make a submission.

1 Introduction

The final project is about conducting a real-world machine learning project on your own, with everything that is involved. Unlike in the programming projects 1-5, where we gave you all the scaffolding and you just filled in the blanks, you now start from scratch. The past programming projects provide templates for how to do this, and the most recent video lectures summarize some of the tricks you will need (e.g. feature normalization, feature construction, other performance measures). So, this final project brings realism to how you will use machine learning in the real world.

The task you will work on is forecasting election results. Economic and sociological factors have been widely used when making predictions on the voting results of US elections. These factors vary a lot among counties in the United States. In addition, as you may observe from the election map of recent elections, neighboring counties often show similar patterns in terms of the voting results. In this project you will bring the power of machine learning to make predictions for the county-level election results using Economic and sociological factors and the geographic structure of US counties.

2 Dataset

The dataset you will be working with in this project is the county-level US election dataset in 2016 and 2012. Specifically, you will be provided with the following files:

- **train_2016.csv**: This is the dataset containing information about the county-level Economic and sociological factors. The columns are FIPS (county FIPS Code), DEM (total votes for Democratic Party in this county), GOP (total votes for Republican Party in this county), as well as median income in USD, net migration rate, birth rate, death rate, Bachelor rate and unemployment rate in each county. **Your label for a county will be 1 if DEM is strictly greater than GOP, and it will be 0 otherwise**. This needs to be standardized this way for how we are evaluating prediction accuracy in a private Kaggle competition, and you will need to generate these binary labels from vote counts.
- **train_2012.csv**: This is an extra county-level information file that you may find useful for your models and prediction. This file contains exactly the same set of counties and columns as the train_2016.csv file, except that the information for this file is from the 2012 US election. It may or may not help with your prediction task for the voting results for test counties in 2016. **Note that you do not have to use this dataset**.
- **graph.csv**: This is the dataset for the geographic neighbor structure of counties where each row represents the FIPS Code of two directly connecting neighbor counties. You are not required to use this information but you may find the geographic information useful, for example, by introducing extra features into your model using this file. **Note that we recommend using this dataset in the creative section while not in the baseline section as detailed in the Jupyter Notebook**.
- **test_2016_no_label.csv**: This is the dataset with the same columns as train_2016.csv for the test counties, except that the DEM and DOP columns which imply the target true labels for the task have been omitted. Note that you are not predicting the specific vote counts, but the labels indicating voting results. Your prediction to be submitted to Kaggle will be the binary prediction for the counties in this file.

- **test_2012_no_label.csv**: This is the dataset with same columns as train_2012.csv for the test counties (same test counties as in test_2016_no_label.csv), except that the DEM and DOP columns which imply the target true labels for the task have been omitted. **Note that you do not have to use this dataset**.

3 Your Task

You will be provided with a template Jupyter Notebook and you are required to do the necessary data preprocessing (e.g. normalizing the features), model construction and selection with validation, hyperparameter tuning, and generating test-sample predictions with this Jupyter Notebook. You are encouraged, although not required, to use existing machine learning packages and frameworks for your modeling process. Examples of such packages are Scikit-learn, PyTorch, Pandas. In a fast moving field like machine learning, these packages change all the time. So, part of the realism of the final project is that you can make decisions on which packages to use, and read up on these packages, on your own.

You are asked to give binary predictions (Democratic Party or Republican Party) on the test counties of the 2016 election result in **test_2016_no_label.csv**, which includes the features of the test counties. You will join a Kaggle competition to submit a CSV file with your prediction on the test counties and we will test your submission in terms of **weighted accuracy** (This will be explained in the Jupyter Notebook).

4 Collaboration

You can work on the final project on your own or form a group of 2 or 3 students. You cannot discuss ideas or share code with other groups. If you are stuck in any part for a long time, please feel free to discuss your issue with the course staff during office hours. In particular, each project team will have a particular TA assigned as a mentor. This mentor is a good person to reach out to first with any questions.

5 Grading

Roughly 75% of your grade will come from how well you craft a basic solution to the learning problem. You are required to do some kind of train and validation split for model selection and provide test-sample predictions for at least two machine learning algorithms from class. In more detail, this includes:

- Load and preprocess the dataset. Explain in words what you did and why you made these choices.
- Perform model selection via some form of train and validation split. Explain in words what you did and why you chose the methods.
- Use at least two different training algorithms from class. Explain why you made these choices.
- Reaching the baseline accuracy of 68% in Kaggle.

As you can see, writing and justifying your choices is just as important as writing the code and getting good performance on the test set.

The remaining 25% will be given for creative ideas that go beyond the basics. Again, this gives some realism to the final project, since no machine learning project is like the other, and they all require some creativity. Here are some ideas of what you could try, but there is really no limit to your creativity in improving on the basic solution.

- Create new features from the features you already have, or apply other learning algorithms that are better suited for this task, or modify some of the learning algorithms to better model this particular problem (e.g. choice of loss function to train with).
- Make use of the 2012 data to introduce new features about demographic change, get more training examples, learn more about participation in voting, and many other ideas.
- Make use of the graph data to introduce new features that capture geographic adjacency, how far apart two counties are, how voting results may be locally smooth (i.e. neighboring counties voting in similar ways), and many other ideas.
- In addition to running the experiments, clearly write up your reasoning and how each thing you tried improved - or did not improve - the results.

- Some of the ideas we tried were able to surpass an accuracy of 75% in Kaggle, and we will reward solutions that achieve this as well.

Clearly, you cannot do all of those, and this is not an exaustive list of things you could try. Be creative, clearly describe what you tried, and report your results in a convincing way. If you try creative and well-argued ideas that do not pan out in better prediction performance, this is a perfectly fine outcome and can be an excellent project.

We are also planning to give some extra credit for particularly accurate solutions, in particular, those that get 80% on the private Kaggle test set or that are the among top 10 in Kaggle on the private part of the test set. Note that the test set is split to two parts in Kaggle, the score (weighted accuracy) your see in Kaggle is the public part while the score on the private part of your submission is not visible until the final project is over. We will use the public accuracy for the 68% baseline and the 75% creative solution, but for the extra credit you have to do well on the private accuracy. Hence getting into the top 10 in the public leader board does not guarantee the extra credit, which is again part of the realism of working in machine learning. The proof is in how well your method does in the real world – and the more robust your model selection, the more likely your learned rule will do well.

6 Academic Integrity

The following are the important rules concerning academic integrity for this project. We will run your code to test that your submitted prediction accuracy is legitimate.

- The prediction task is for the 2016 US Election results, which is public information. So, you are not allowed to use outside sources that reveal any information about the election outcomes. So, using such external information to boost your prediction accuracy is definitely a violation of academic integrity, and your submission file to Kaggle should be generated from your notebook and not be further modified in any way.

- The only input that your training algorithm can use is the three files that we provide with you, **you CANNOT use any extra dataset resources beyond what is provided** to train your algorithm and boost your accuracy.

- Collaboration between groups is also a violation of academic integrity, and a key aspect of the project is to make your own choices for feature engineering, choose between training algorithms and tuning your models.

It is fine to use resources like software packages and read published papers. But make sure to reference and cite any resource you used in your notebook.

7 Due Date

The final project will be due on December 15th. You need to submit your Jupyter Notebook on Canvas and a PDF version of your notebook with answers to all questions on Gradescope. **Note that December 15th is a hard deadline for this project, no late submissions will be accepted and you cannot use any of your unused slip days**.

8 Useful Resources

Here are some official tutorials for some packages that you may find helpful for the project. Note that these packages are recommended but not required for this project.

- Pandas: `https://pandas.pydata.org/pandas-docs/stable/user_guide/10min.html`

- Scikit-learn: `https://scikit-learn.org/stable/tutorial/basic/tutorial.html`

- PyTorch: `https://pytorch.org/tutorials/`

9 Kaggle Competition

- Basic Solution: https://www.kaggle.com/t/3065eb182a9440b5ae296edab3fe77bc
- Creative Solution: https://www.kaggle.com/t/ad9c629d7c3847d89520d95a20b1f4a1