Capstone 1: Machine Learning

https://github.com/mdubow/Springboard-Data-Science-Career-Track/blob/master/Capstone-Project-1/Capstone%201%20Final%20Code.ipynb

The goal of this project is to use data on opioid prescriptions along with demographic data to predict death rate, given a county and a year, and to examine the importance of each demographic feature. I decided that this was a supervised learning problem, given there are clear labels for both features and targets. Further, because this project involves mathematically computing rates from previous data instead of labelling/classifying data, I believed I should use a model with regression instead of classification. I started with the most fundamental regression model: linear regression.

On its face, I had over 250 potential features to use for this model, including year, state, county, prescription rate, population. Among these features were the demographic data, which consisted of population percentages for different groups divided by age, race, and sex. My aim in building these machine learning models was to discover which features would produce the best model, in terms of both performance and interpretability. Each model roughly follows the same construction plan - define target and features, drop rows with missing data since they cannot be used in the model, split the remaining data into training and test data, then fit the model and measure and accuracy.

Based on my previous visual analysis, I had a suspicion that there was a linear relationship between prescription rate and death rate. Consequently, I built my first linear model using Sklearn with just prescription rate as a feature. Its accuracy score was around 20%, which told me that prescription rate was indeed important, but would not suffice on its own. Adding year as a feature to prescription rate actually produced a much worse score, which told me year should probably be excluded from the model.

Next, I wanted to test county and state as features, but Sklearn does not allow categorical data to be used for linear models as is. This led me to use Pandas' get_dummies function to generate 1521 columns for each county and state, which hold a 1 if they represent a given location and a 0 otherwise. I also set the function parameter drop_first to True, in order to avoid redundant data which can negatively impact Sklearn's models. The score for this model was quite low - ~0.3% - but I thought it might be because of the large number of features. The negative impact of an overabundance of features, called the "curse of dimensionality", can be overcome with dimensionality reduction. For this I used Sklearn's Principal Component Analysis (PCA), which is able to convert these features to lower-dimensional space, in this case from

1522 features to 100 components. This model with reduced features was much improved, with a score of 35%. I also examined the model's explained variance ratio for its components, which showed that every component was making a significant contribution to the model.

For my next model I tested demographic data as the features, which had an accuracy of 43%. Combining demographic data with prescription rate improved the model even more, with it predicting at 48% accuracy. It seemed both features should be used in the final model.

To my disappointment, adding prescription rate to the county and state data with PCA only marginally improved the accuracy (35%). Running a similar model without PCA resulted in a clear case of overfitting, with a training accuracy of 72% and a test accuracy around 0%. My linear model with county and state data plus demographic data improved somewhat to 39%, which is about how well the next model performed when I added prescription rate again. There almost appeared to be a limit to how well the county and state could predict death rates even when other important features were added.

To this point, the best model I had was linear regression with demographic data and prescription rate as features. I decided to test other algorithms to see if I could get further improvement. I implemented Sklearn's Random Forest Regressor, which builds many decision tree estimators and takes the mean of them for predictive power. Instead of running this algorithm a number of times with different numbers of estimators, I instead used Sklearn's GridSearchCV, which allows you to cross-validate data and optimize model hyperparameters at the cost of runtime. With GridSearch I also utilized Lasso and Ridge regressions; these algorithms are similar to linear regressions, but they penalize some coefficients to reduce model complexity and overfitting, the difference being that Lasso can reduce some coefficients to 0 (which also allows for features selection; more on that later). GridSearch is useful for choosing the regularization parameter of these models, which determines how much the variables are reduced. Compared to the previous accuracy of 48% for linear regression, these models scored -.09%, 37%, and 42% for Random Forest, Lasso, and Ridge respectively. Rerunning the Random Forest model with Randomized Search CV (which runs more quickly than GridSearch by not running every single model) did not change the accuracy.

I was seeing only incremental improvements in my model accuracy, yet they were all still far from good models. I changed my strategy by shifting the target variable from death rate to number of deaths, which allowed me to use population as another feature. I reran my basic linear regression model with population, prescription rate, and demographic data as features and immediately saw drastic results: a 91% accuracy score. When I constructed another model adding county and state data, I saw a slight improvement in test accuracy, but because it was higher than training accuracy, I consider this result to be unreliable. Furthermore, there is good reason to not use a model with PCA in this project, as it makes interpreting the final result quite difficult. Because my aim is to see how individual features contribute to the model, dimensionality reduction actually makes this unclear.

For making predictions, I first used my best-performing model, the first linear regression with deaths as the target variable. However, this proved to be problematic, as it led to negative predictions for number of deaths, which obviously cannot be true. I tried to clip these predictions by holding 0 as the lower limit, but when I looked at the descriptive statistics for the resulting dataset I saw that the death rates had a much higher mean and standard deviation than I expected (based on my previous analysis using state, rather than county, data).

I created another predictive dataset, this time using Lasso. Lasso has two advantages: it can be programmed to ensure positive output values, and feature selection is quite easy because unimportant features are shrunk to 0. I compared both models visually, using violin plots, to examine the spread of values for death rate and “pain ratio” (death rate / prescription rate). The output for the Lasso model seemed more reasonable to me, so even though its accuracy was lower (80%), for this problem I felt more comfortable using it.

This final step was to examine the coefficients of the features to see which contributed the most to the model. For Lasso models this is quite easy, as you only have to look at the features with nonzero coefficients. Unsurprisingly, population and prescription rate were among the most important features. Other features included the black male population from ages 0 to 4, male Pacific Islanders ages 15 to 19, indiginous females ages 45 to 49, white males 85 and up, and the total population ages 80 to 84. Curiously, when examining the most important features of the linear model, the only common feature was male Pacific Islanders 15-19. In my opinion, this result does not indicate that these groups have the greatest risk of opioid overdose, but rather the model may be using these specific features to understand the overall demographics of the given county. The fact that our best models use demographic data and not county or state data tell me that opioid use and abuse is determined more by demographics than physical location.

The hyperparameter table for all my models is below. Note that models which used a search function to find the best parameters have unknown training accuracy.

model	target	n_features	features	test accuracy	training accuracy
Linear Regression	death rate	1	prescription rate	0.205	0.149
Linear Regression	death rate	2	prescription rate, year	0.007	0.0099
Linear Regression	death rate	1521	county, state (dummy variables)	0.003	0.005
Linear Regression with PCA	death rate	1521	county, state (dummy variables)	0.346	0.341
Linear Regression	death rate	249	demographic data	0.432	0.525
Linear Regression	death rate	250	demo data, prescription rate	0.484	0.553
Linear Regression with PCA	death rate	1522	county, state, prescription rate	0.351	0.343
Linear Regression, no PCA	death rate	1522	county, state, prescription rate	-3.68E+23	0.724
Linear Regression, no PCA	death rate	1770	county, state, demo data	0.031	0.038
Linear Regression with PCA	death rate	1770	county, state, demo data	0.392	0.409
Linear Regression with PCA	death rate	1771	county, state, prescription rate, demo data	0.391	0.411
Random Forest	death rate	250	demo data, prescription rate	-0.0009	-
Lasso Linear	death rate	250	demo data, prescription rate	0.368	-
Ridge	death rate	250	demo data, prescription rate	0.422	-
Random Forest with Randomized Search CV	death rate	250	demo data, prescription rate	-0.0009	-
Linear Regression	deaths	251	demo data, prescription rate, population	0.908	0.914
Lasso Linear	deaths	251	demo data, prescription rate, population	0.864	-

Ridge	deaths	251	demo data, prescription rate, population	0.871	-
Linear Regression with PCA	deaths	1772	county, state, prescription rate, demo data, population	0.912	0.889