Project McNulty / Will Hetfield

Background

Using insurance data to predict whether an insurance company will stop offering insurance plans in a county where they are currently offering health plans on Obamacare exchanges. Is there a way to create an early warning system for consumers and policymakers

Data

https://www.cms.gov/cciio/resources/data-resources/marketplace-puf.html
https://numeracy.co/standard-library/us-population/counties-latest

This data is not for all 50 states. State Based Exchanges which do not rely on Federal Infrastructure are reported in a different data set: "The Exchange PUFs exclude data from SBEs that do not rely on the federal platform for QHP eligibility and enrollment functionality. 4 For data on SBE states, see the State Based Health Insurance Exchange Public Use Files (SBE PUF). "

Cleaning

Made module **mcnulty_cleaning.py** to use for importing various cleaning functions. I called the functions separately in the Jupyter notebook for illustrative purposes but it would be easy to order them into a single main function. Note there was a flag to filter out all dental plans which I used

Data Exploration

Roughly 35% of observations are an exit. Total observations are about 23,000, representing county level stay/leave decisions for Issuers.

15,000 of the observations are for an insurer just offering one plan in a county. Note that by Plan I believe the system means a HMO or PPO. Another layer of detail is Plan Variant (gold,silver,bronze in exchange terminology). But this layer of detail is only in the Rate.csv and not linked to counties, only states.

I did groupbys by the target to look at the average values of different features f to get a feel for the data, as well as a few basic plots. There were no major correlations in the feature correlation matrix between the features other than Average Median Rate in the State and the Prior Year Percentage Change in that Median Rate (functions of each other).

Analysis Module

Created small python module **mcnulty_analysis.py** for helping create train, test splits with random Issuers. There are also some helper functions that run single feature Naive Bayes and Logistic Regression for a list of features, printing at classification metrics.

Holdout Set

Holdout set was based on 20% of the IssuerIDs, not on a random sample of observations. Insurers often decide to leave counties simultaneously, so want the model to learn what causes the decision, not a particular insurers decision to leave.

Metric Selection

Selected F1 score as this is binary classification and want to take into account both precision and recall. Policymakers can't prepare without a warning, so need to identify insurer exits (recall), but can't have so many predictions to warn them or they will not take meaningful steps after too many false alarms (precision).

Baseline Models

Using below features individually,

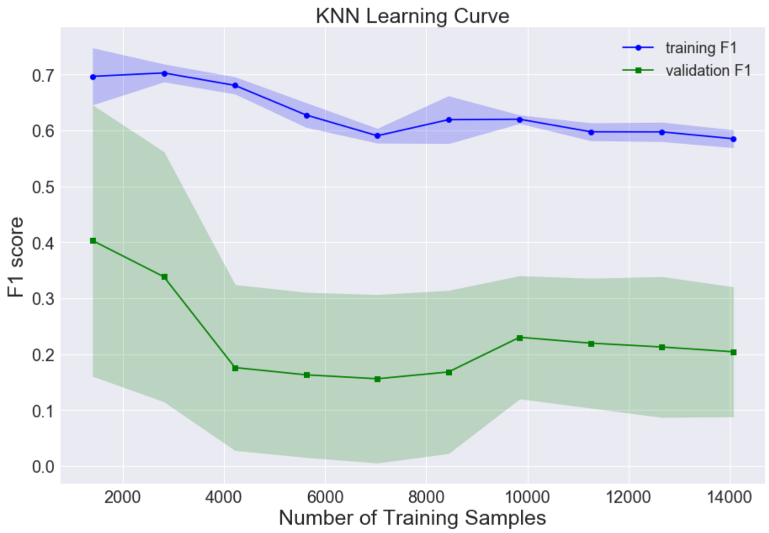
Population_per_All_County_Plans
County Population divided by Total number of Plans Offered by all Issuers in County
Medicaid_Expansion
Flag if Medicaid Expanded in State (1 Expanded, 0 No)
First_Year_in_County
Flag if First Year for Issuer Offering Plan in County (1 First Year, 0 Not First Year)
Average_Median_Rate_in_State
Average of each Issuer's Median Rate in the State
State_Percent_Change_Rate_From_Prior_Year
Percentage change from prior of above feature (*Average_Median_Rate_in_State*)

performed single feature Naive Bayes and Logistic Regression. Predictions were simply all 0 class (insurer stays) and no exit predictions. Naive Bayes and Logistic Regression were run with all the features and had F1 scores of 0.09 and 0.005, respectively.

Tried KNN to see if non-parametric model would handle the data better. From my 80% of data remaining for training, I did a further train/test split and then performed Grid Search. using this new training of the training set (80% of the 80%). The parameters I tuned were the number of neighbors (between 1 and 35) and for the weights parameter used to make the classification in validation ('uniform' or 'distance' (uniform weights all neighbors equally while distance weighs by inverse of the neighbor's distance)).

Returned parameters from the Grid Search were {'knn__n_neighbors': 31, 'knn__weights': 'uniform'}, which had an average F1 score in cross validation of 0.20. Using these parameters on the the test set (20% of the 80%), the F1 score was 0.34.

Next I examined the the learning curve for the KNN algorithm using these parameters from grid search. The learning curve object in sklearn uses cross-validation, so there are a new range of training and validation F1 scores at each of the 10 training sample sizes used to for the learning curve. Shading represents the +- one standard deviation of the F1 scores at the respective sample size.



The model was showing high variance as training F1 scores were much higher than validation F1 scores, ie, overfitting. I looked at Random Forest to see if that could reduce the variance in performance as an ensembling method.The returned parameters from the Random Forest Grid search were {'max_depth': 3, 'min_samples_leaf': 10, 'n_estimators': 10} and the average F-Score from grid search with these parameters was 0.28. I retrained on the full data set with these parameters and had a F1 of ~0.40.

I proceeded to use the Random Forest model on the initial holdout set. Note I was using no limit on maximum features as I had so few features. F1 on the final holdout was 0.30. Most meaningful feature importance was the County Population per Total Plans. Perhaps this is suggesting that at a fixed population level, the more plans being offered means more competition, making it likely than an issuer will exit the county.

Extensions

1. Insurers leave many if not all counties in a state simultaneously. So many of the observations probably not unique, or one large county is determining whether an insurer stays in a state at all. It would be worthwhile to further investigate the instances when an insurer leaves some but not all counties in the state. Maybe reframe the problem differently.
2. Engineer more features
3. Incorporate data from other states