Predicting Heart Disease Mortality

Thiha Aung, July 2018

Executive Summary

The goal of this analysis is to explore the dataset and develop a predictive machine learning model. The model is to predict the rate of heart disease per 100,000 people across the United States. The dataset include the heart disease mortality rate and its related characteristic and observations. The dataset is publicly provided by the United States Department of Agriculture Economic Research Service (USDA ERS) and which is compiled from various kind of sources.

The label to be predicted is the mortality rate (death rate) which is the continuous Integer value. So, the project is going to be developing a regression machine learning model.

The analysis include of exploring the category and numerical features and finding the key observations. And explaining them with statistic and data visualization techniques. After that, the dataset is preprocessed and ready for model.

Tree based machine learning algorithms with ensemble, boosting and bagging techniques were used in this predicting analysis. Python based technologies, tools and libraries were mainly used in this analysis and Microsoft Excel.

The analysis can be briefly concluded as following:

- **area__rucc** - People living in non-metro area were more suffered with heart disease.
- **econ__economic_typology -** Physical and manufacturing workers had more mortality heart disease rate.
- **econ__pct_civilian_labor** - military men and women were more resilience to the heart diseases.
- **demo__pct_adults_with_some_college, demo__pct_adults_bachelors_or_higher** - People with higher educations had less likely to face the death with heart problems.
- **health__pct_excessive_drinking, health__pct_adult_smoking** - Excessive drinkers had less rate of mortality with heart disease while chainsmokers faced more deaths with heart problems.

Understanding the Dataset

The analysis is based on 33 features and 3198 observations of the people who had lost their lives with heart diseases. The observations information can be grouped as:

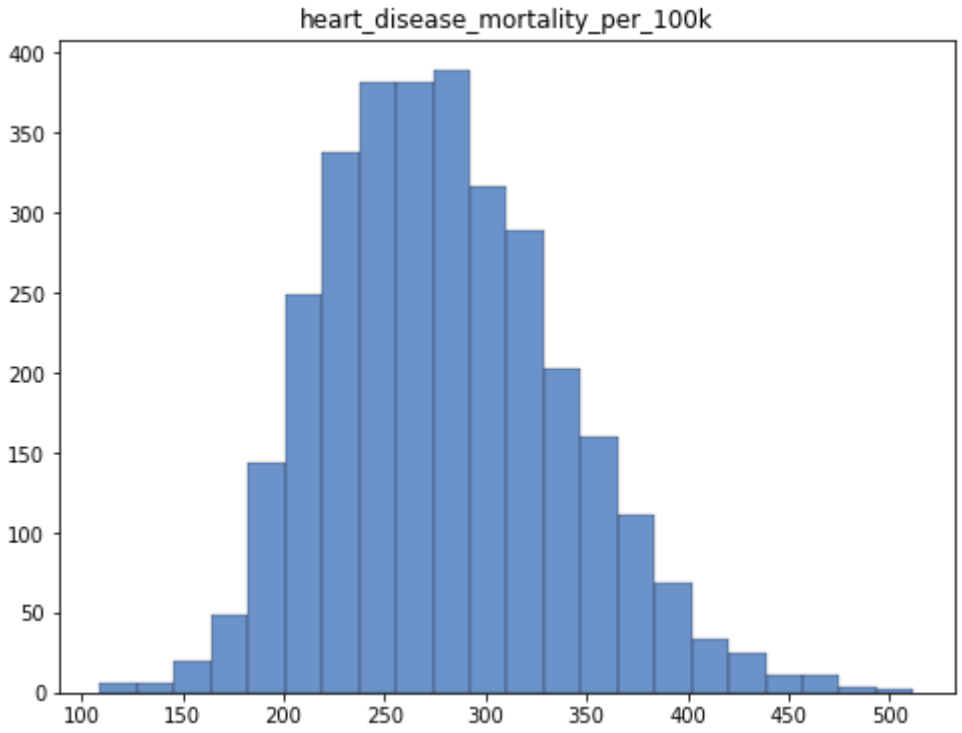
- Area
- Economic
- Health
- Demographic

Summary statistic

The summary statistics for min, max, mean, median, standard deviation are calculated for numerical features.

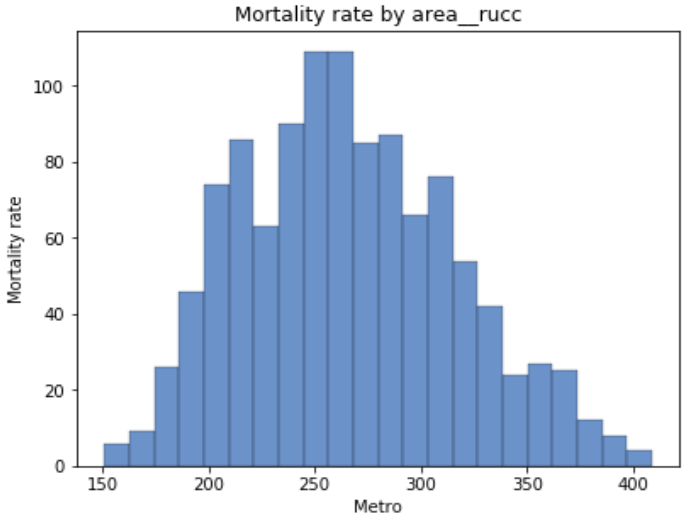
Feature	Min	Max	Mean	Median	Std Dev
econ__pct_civilian_labor	0.207	1	0.467190744	0.468	0.074399587
econ__pct_unemployment	0.01	0.248	0.059696373	0.057	0.022947426
econ__pct_uninsured_adults	0.046	0.496	0.217463079	0.216	0.06736233
econ__pct_uninsured_children	0.012	0.281	0.086067272	0.077	0.039848604
demo__pct_female	0.278	0.573	0.498810701	0.503	0.024399213
demo__pct_below_18_years_of_age	0.092	0.417	0.227714956	0.226	0.034282201
demo__pct_aged_65_years_and_older	0.045	0.346	0.170043179	0.167	0.043694436
demo__pct_hispanic	0	0.932	0.090206821	0.035	0.142763126
demo__pct_non_hispanic_african_american	0	0.858	0.091046308	0.022	0.147165429
demo__pct_non_hispanic_white	0.053	0.99	0.769989049	0.853	0.207849686
demo__pct_american_indian_or_alaskan_native	0	0.859	0.024680225	0.007	0.084563079
demo__pct_asian	0	0.341	0.013108573	0.007	0.025431459
demo__pct_adults_less_than_a_high_school_diploma	0.015075377	0.473526474	0.148814992	0.133233664	0.06820773
demo__pct_adults_with_high_school_diploma	0.065326633	0.558912387	0.350566598	0.355014793	0.070554063
demo__pct_adults_with_some_college	0.109547739	0.473953013	0.301143089	0.301587302	0.0523181
demo__pct_adults_bachelors_or_higher	0.011077543	0.798994975	0.199475321	0.176470939	0.089308363
demo__birth_rate_per_1k	4	29	11.67698562	11	2.739516437
demo__death_rate_per_1k	0	27	10.3011257	10	2.786143377
health__pct_adult_obesity	0.131	0.471	0.30766771	0.309	0.043227777
health__pct_adult_smoking	0.046	0.513	0.213627652	0.21	0.06289512
health__pct_diabetes	0.032	0.203	0.109260013	0.109	0.023215901
health__pct_low_birthweight	0.033	0.238	0.083895889	0.081	0.022251206
health__pct_excessive_drinking	0.038	0.367	0.164841441	0.164	0.050473585
health__pct_physical_inacticity	0.09	0.442	0.277161452	0.28	0.053003306
health__air_pollution_particulate_matter	7	15	11.62586751	12	1.557996187
health__homicides_per_100k	-0.4	50.49	5.947497969	4.7	5.03182206
health__motor_vehicle_crash_deaths_per_100k	3.14	110.45	21.13261776	19.63	10.48592253
health__pop_per_dentist	339	28130	3431.433649	2690	2569.450603
health__pop_per_primary_care_physician	189	23399	2551.339286	1999	2100.459467

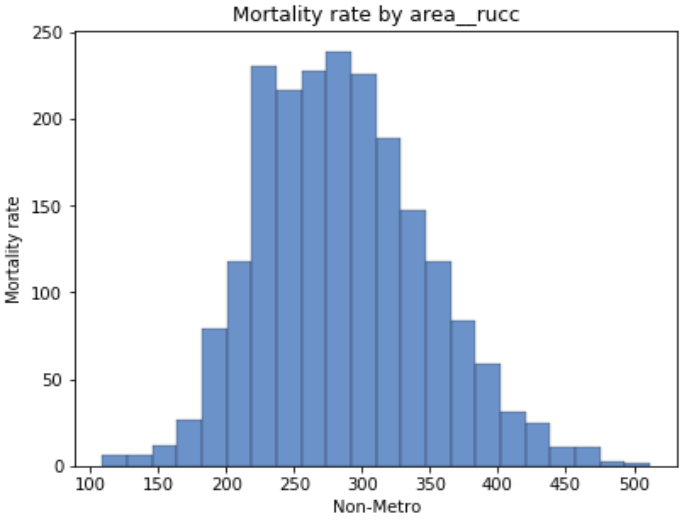
The distribution of heart disease mortality rate looks like normally distributed with a little bit of skewness to the right. It is distributed with the **mean** value 279.37 , **median** 275 and **standard deviation** of 58.953338.

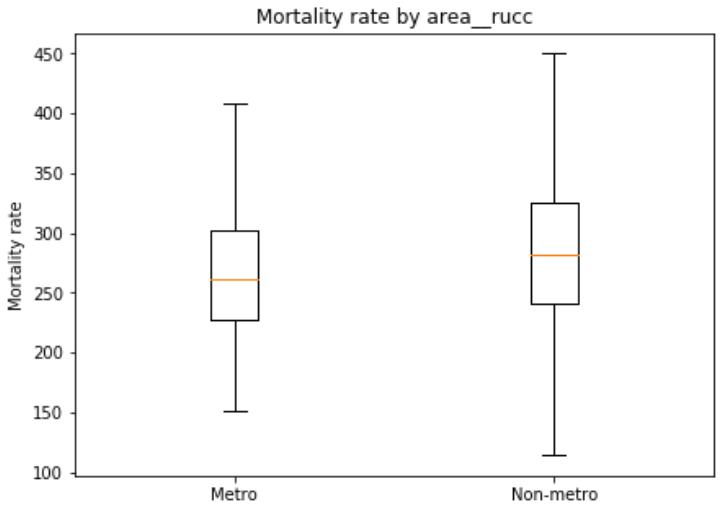


Key observations of category features

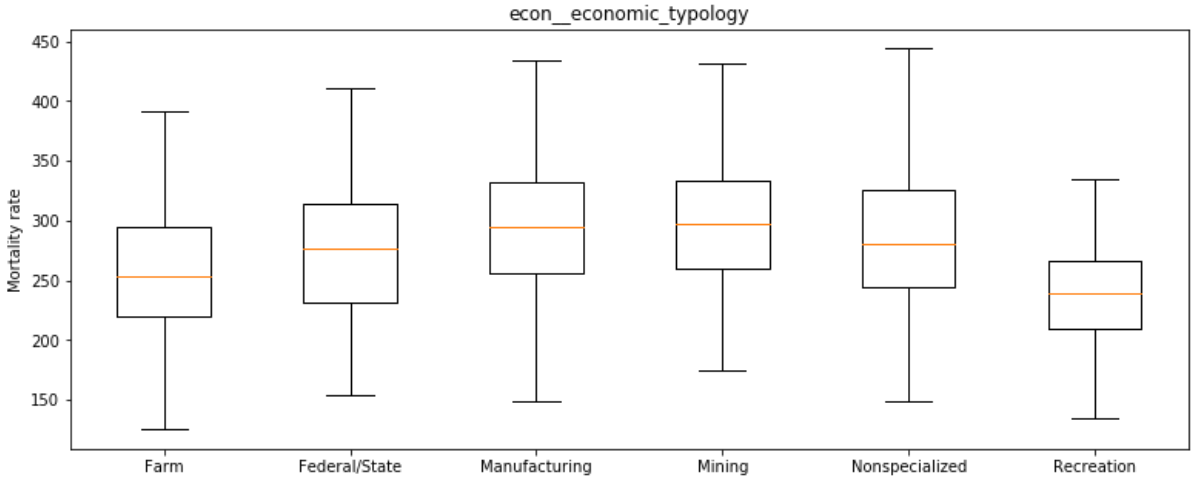
Most of categorical features are related with area and economy. One key observation was found in this (**area__rucc**) category feature. By studying Rural-Urban Continuum Codes observations, the people who lived in non-metro area are were more suffered with heart diseases. The mortality rate distribution of non-metro people is more closely resemble to the overall mortality rate distribution. On the other hand, it can be clearly seen in the boxplot, the mortality rate of metro people are more spread throughout the distribution. The first quartile and third quartile (IQR) of non-metro people is higher than metro people.







Another key category observation is County Typology Codes (**econ__economic_typology**). It makes sense that people with less stressful jobs like Farming and Recreation were less likely to face the deaths with heart diseases. People with more physical, energy consuming and long hours jobs were more likely to face the deaths with heart diseases. Federal government officers and other non-specified jobs were also faced more deaths with heart diseases..



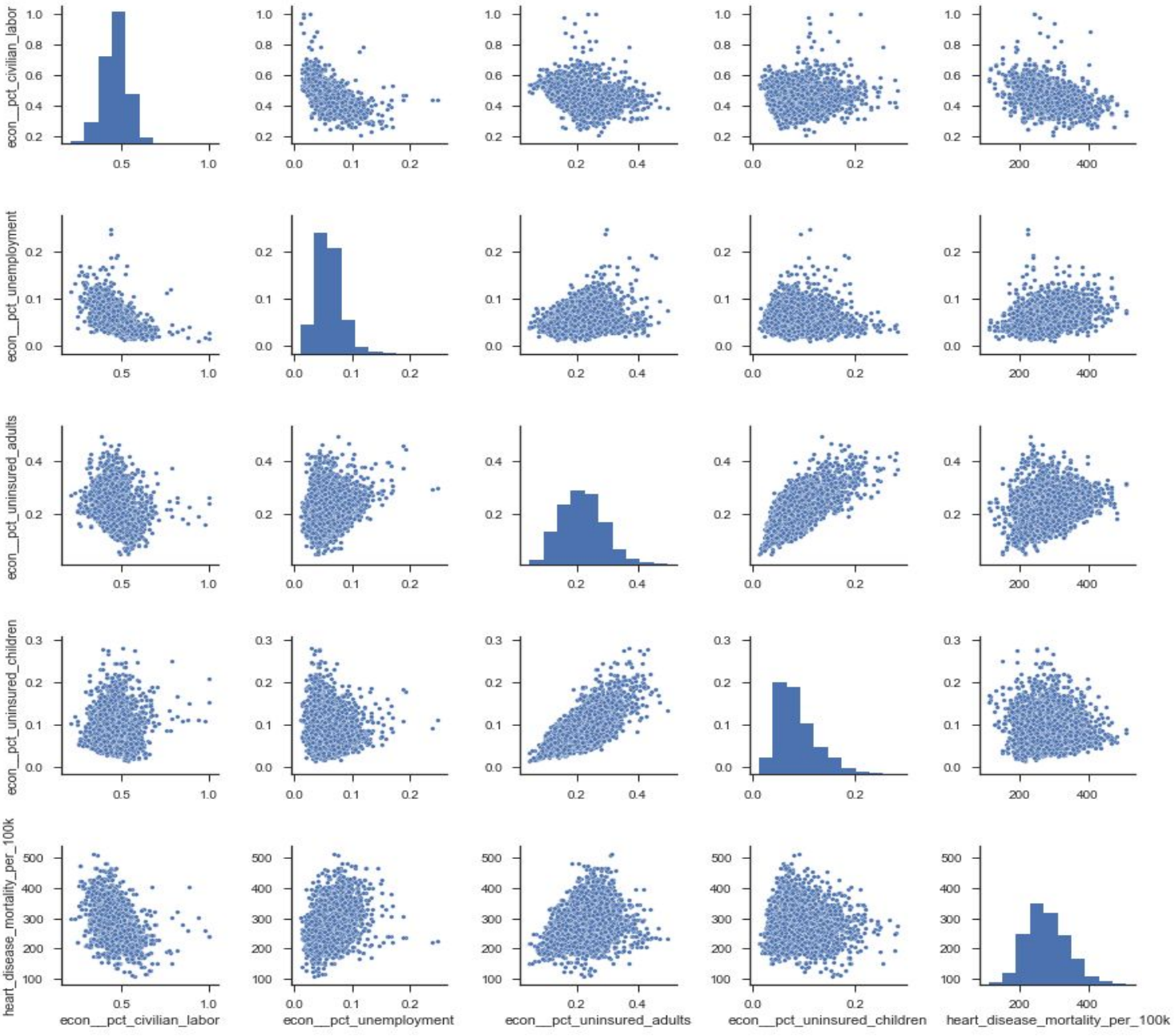
Key observations of numerical features

Most of numerical features are come from economic, demographic and health sectors. The first thing to find out was the correlation of economic related features. The scatter-plot matrix and correlation matrix were created for each economic, demographic and health sectors.

Correlation relationships of economic features

The correlation matrix and scatter-plot matrix were created for economic numeric features.

	econ__pct_ civilian_labor	econ__pct_ unemployment	econ__pct_ uninsured_adults	econ__pct_ uninsured_children	heart_disease_ mortality_per_100k
econ__pct_civilian_labor	1	-0.61905	-0.406961	-0.027352	-0.476644
econ__pct_unemployment	-0.61905	1	0.271667	-0.104056	0.37162
econ__pct_uninsured_adults	-0.406961	0.271667	1	0.717686	0.333963
econ__pct_uninsured_children	-0.027352	-0.104056	0.717686	1	-0.034456
heart_disease_mortality_per_100k	-0.476644	0.37162	0.333963	-0.034456	1

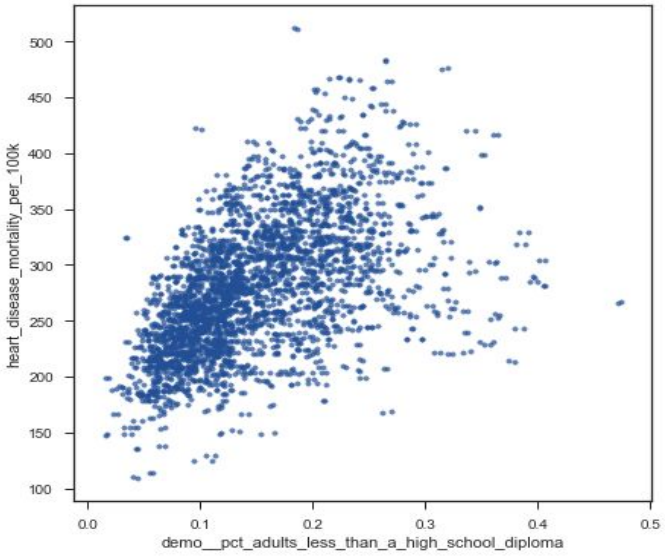


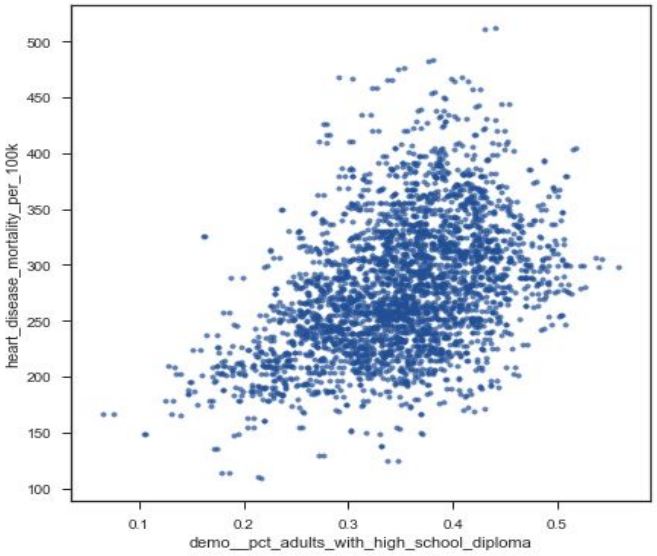
All economic features except **econ__pct_civilian_labor** seem to have a negative correlation with mortality rate, others have slightly positive correlationships . The unemployments had also caused some heart problems for the residents since it has positive correlations with heart mortality rate. Meanwhile, Uninsured adults and uninsured has strong correlation each other.

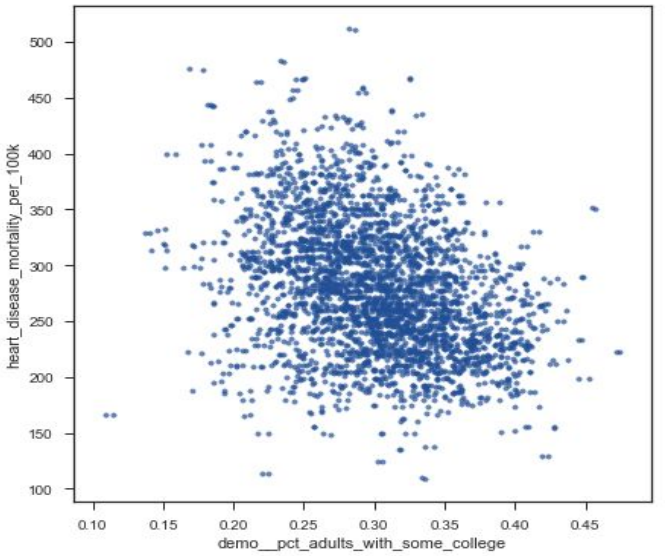
Correlation relationships of demographic features

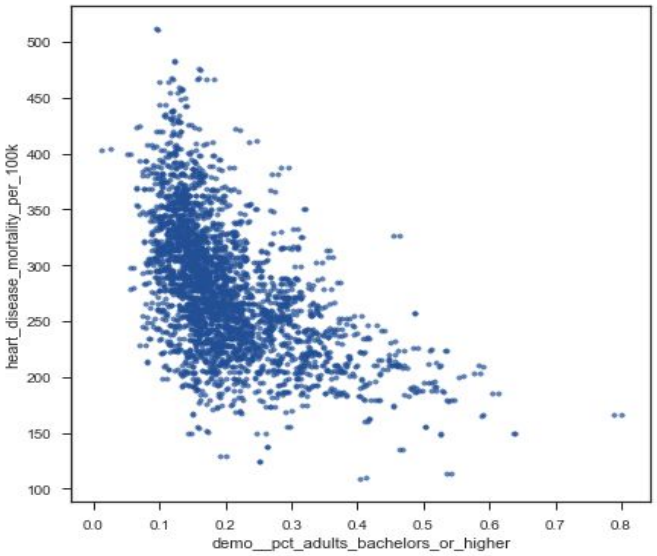
In the demographic numeric features, some of the key observations are related with education.

The insight of educational demographic features is interesting and it looks like a important predictors. People with higher education with college and bachelor or higher degrees had negative correlation with death rate of heart disease. Meanwhile, people with less or with high school diploma had positive correlation with mortality rate.









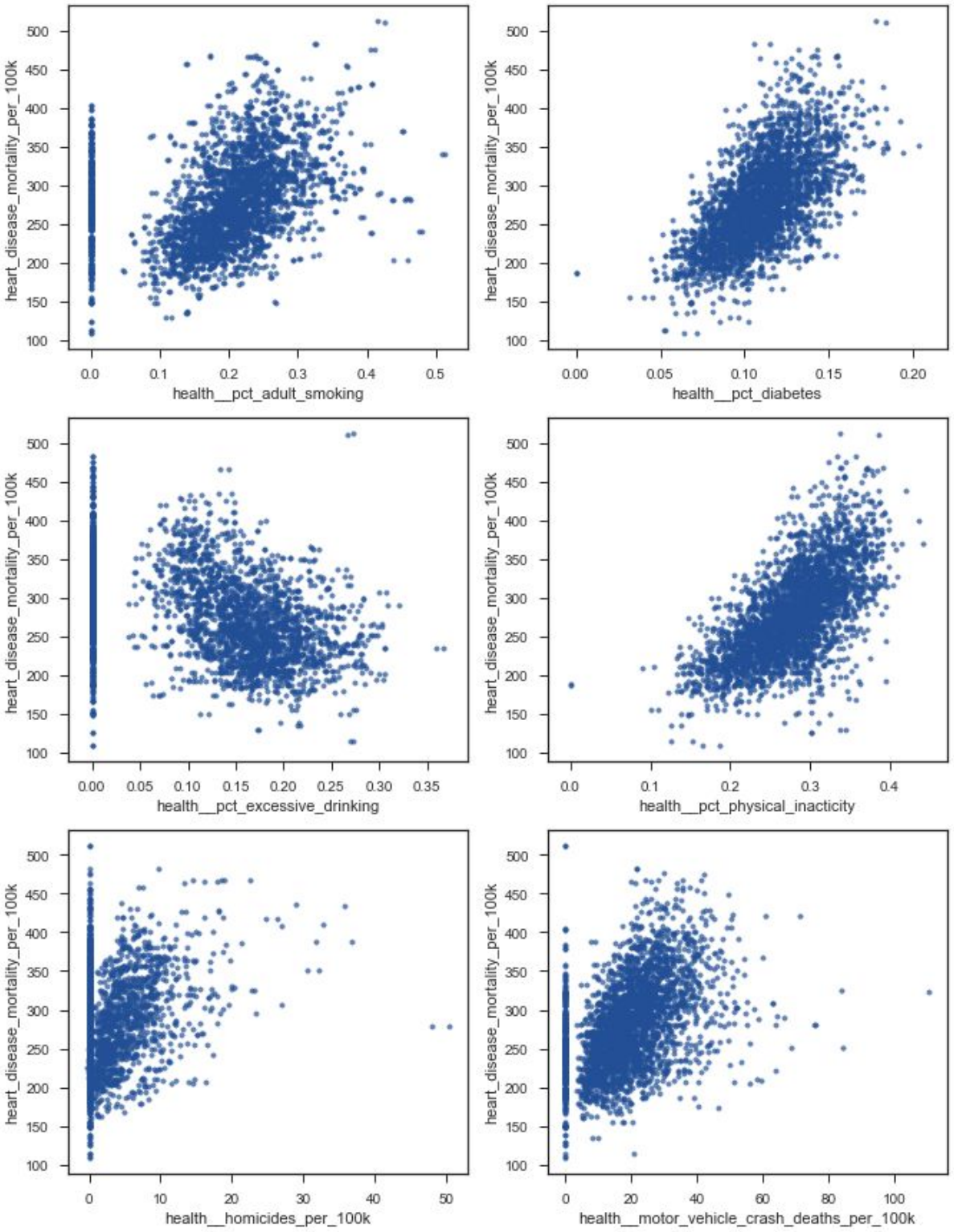
Here is the correlation matrix calculation of demographic features with heart disease mortality rate. The **demo__death_rate_per_1k** also had a strong correlation with heart disease mortality rate.

	heart_disease_mortality_per_100k
demo__pct_female	0.086974
demo__pct_below_18_years_of_age	0.121863
demo__pct_aged_65_years_and_older	-0.05616
demo__pct_hispanic	-0.112352
demo__pct_non_hispanic_african_american	0.375099
demo__pct_non_hispanic_white	-0.157424
demo__pct_american_indian_or_alaskan_native	0.004622
demo__pct_asian	-0.267255
demo__pct_adults_less_than_a_high_school_diploma	0.527382
demo__pct_adults_with_high_school_diploma	0.428137
demo__pct_adults_with_some_college	-0.340764
demo__pct_adults_bachelors_or_higher	-0.541385
demo__birth_rate_per_1k	0.142176
demo__death_rate_per_1k	0.444757
heart_disease_mortality_per_100k	1

Correlation relationships of health features

The key observations inside health features include human habits, diets and society incidents. The most interesting observation is that excessive drinking had negatively correlation with heart disease mortality rate. But the adult smoking is positively correlated with heart mortality rate. The diabetes patients were also suffered with heart problems as diabetes also has positive correlation.

People who lived in dangerous society also faced the death with heart problems. Because the incidents of homicides and vehicle crashes also had positive correlations.



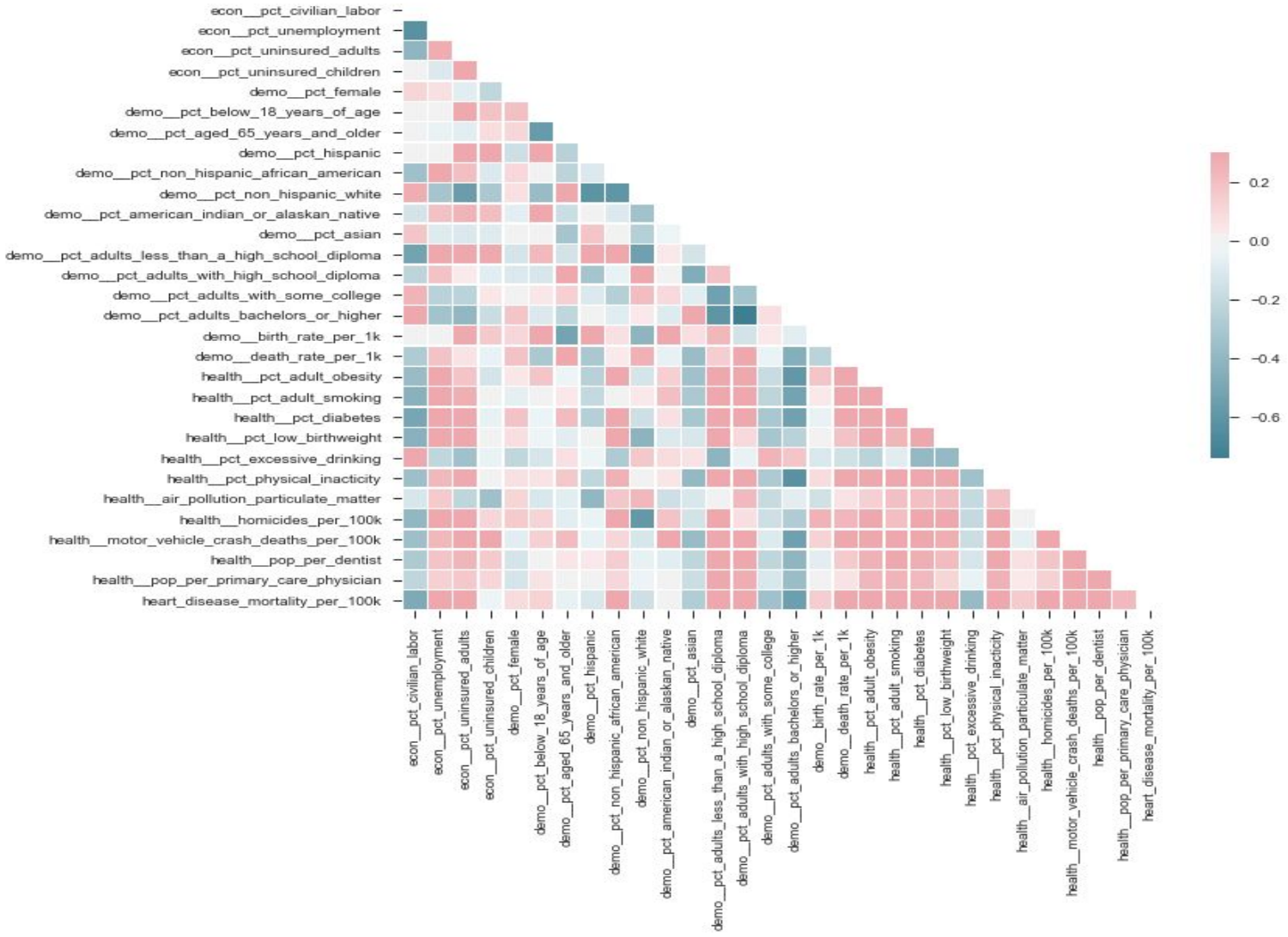
The straight columns in some of plots are the NAN values which were temporarily filled with zero. It is clearly stated that NAN values needed to be take care in preprocessing section.

The calculations of correlation matrix for health features are described as below.

	heart_disease_mortality_per_100k
health__pct_adult_smoking	0.497063
health__pct_diabetes	0.631765
health__pct_low_birthweight	0.476757
health__pct_excessive_drinking	-0.382172
health__pct_physical_inacticity	0.650305
health__air_pollution_particulate_matter	0.150019
health__homicides_per_100k	0.441164
health__motor_vehicle_crash_deaths_per_100k	0.459803
health__pop_per_dentist	0.301232
health__pop_per_primary_care_physician	0.219111
heart_disease_mortality_per_100k	1

Correlation of the whole numerical features

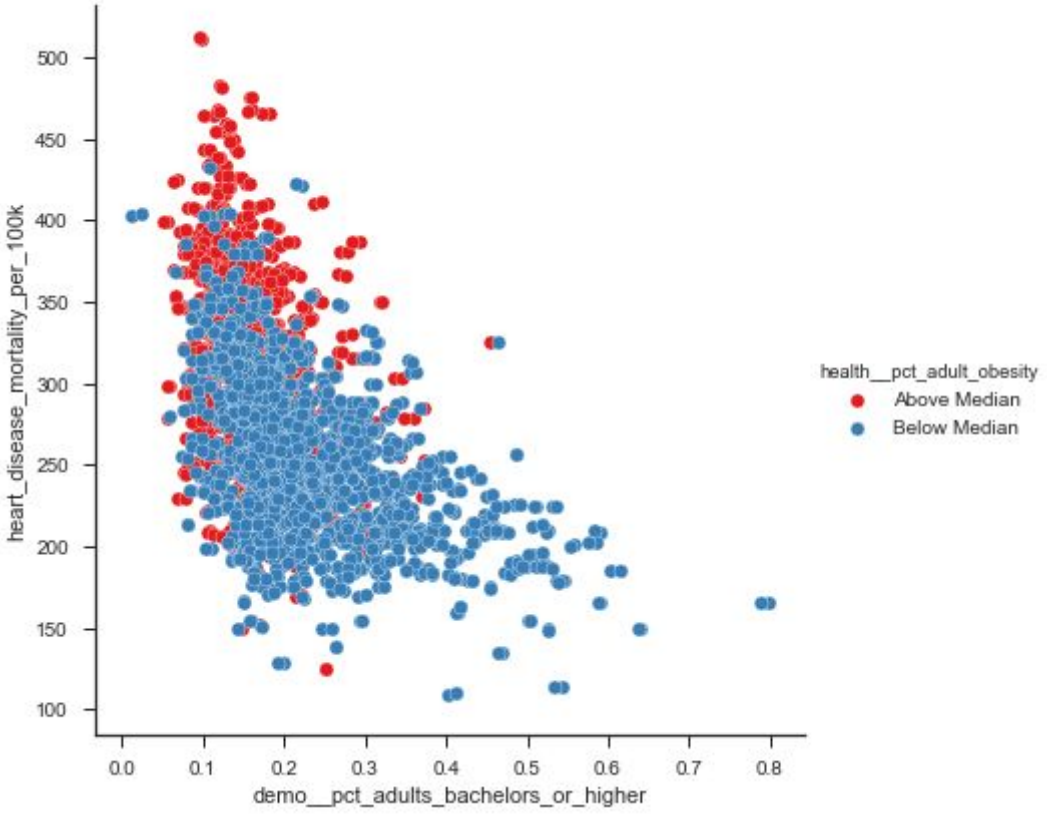
To oversee the correlation of the whole numerical features, this heatmap is created.

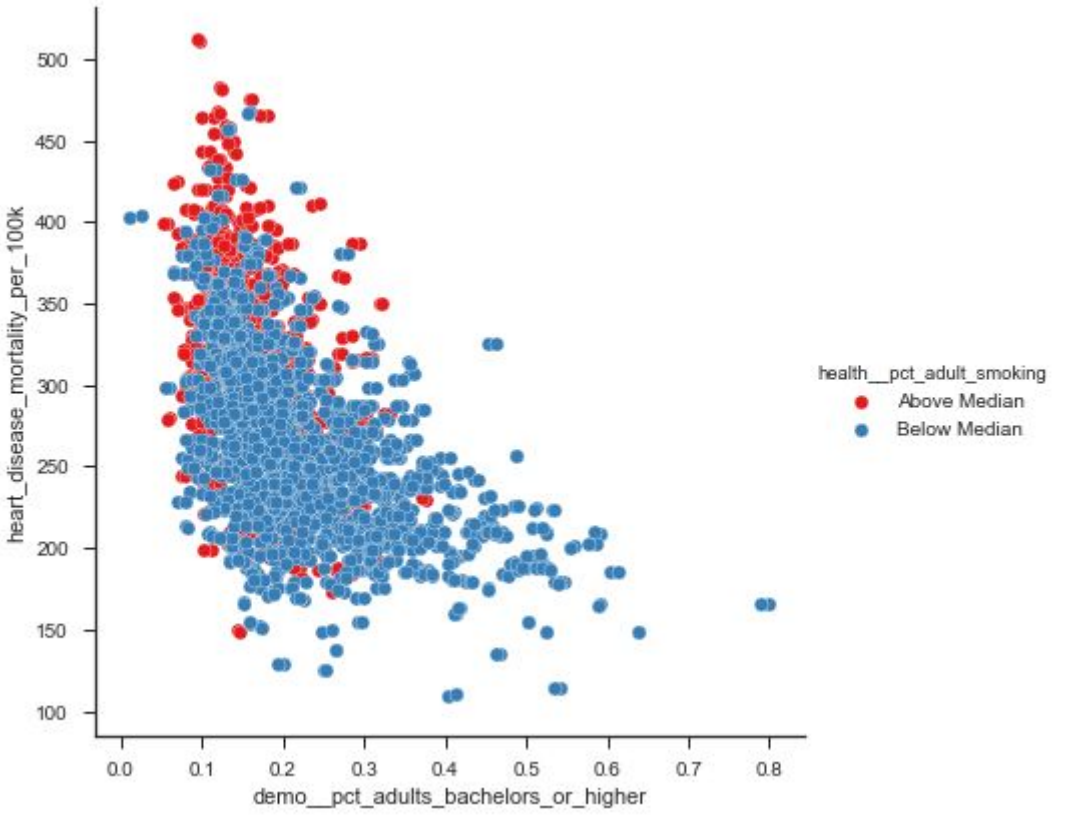


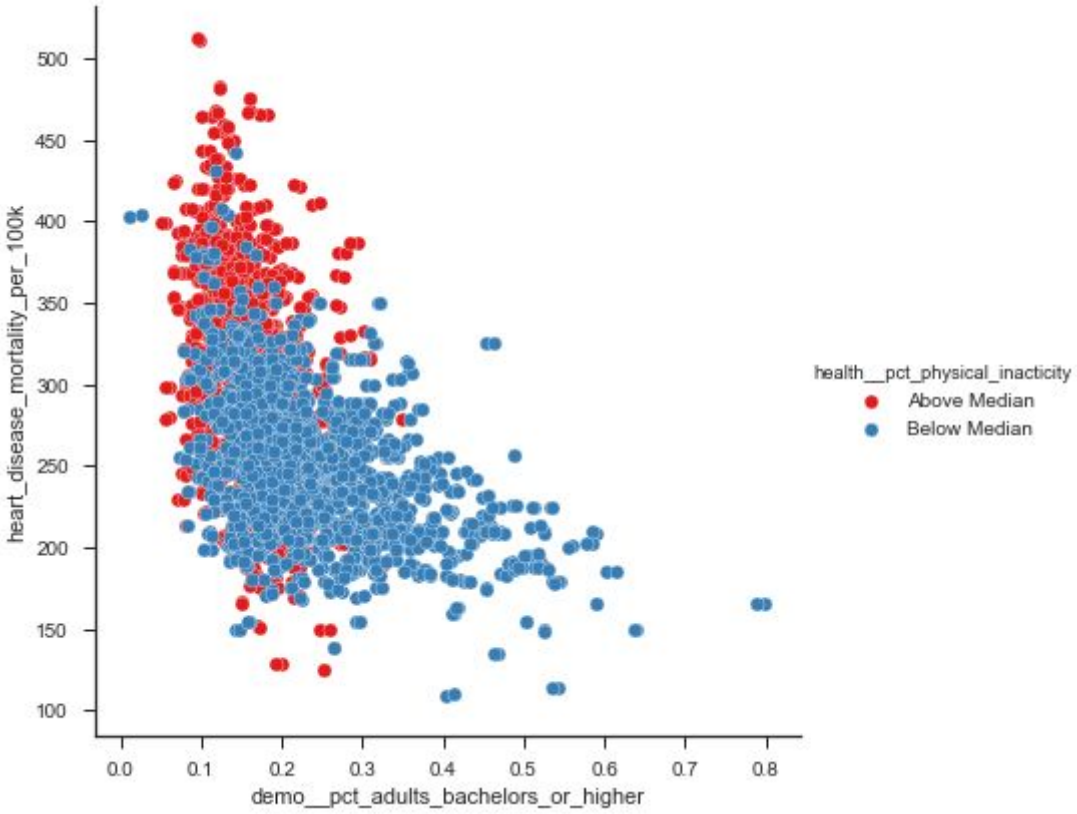
Multi-Facet relationships

Following Multi-Facet plots are created as the counterparts of the above heatmap. In the heatmap, it can be noticed that **demo__pct_adults_bachelors_or_higher** has strong correlation with **health__pct_adult_obesity, health__pct_adult_smoking** and **health__pct_physical_inacticity**. With multi-facet scatter plot, the more insights can be extracted from mortality rate correlation by adult with bachelor and higher degree.

This facet scatter plot indicate that the higher in the education, the lower the obesity rate. The people who has obesity rate higher than the median were faced more death with heart diseases.
Likewise in the adult smoking case, money of the higher education degree holders had smoking rate lower than median.
In physical inacticity case, it also carry similar scenario but it has a small group of lower mortality rate relationship below the median, around the higher educational rate of 0.2 and 0.3.




heart_disease_mortality_per_100k
500
450
400
350
300
250
200
150
100
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
demo__pct_adults_bachelors_or_higher
health__pct_adult_smoking
Above Median
Below Median


heart_disease_mortality_per_100k
500
450
400
350
300
250
200
150
100
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
demo__pct_adults_bachelors_or_higher
health__pct_physical_inacticity
Above Median
Below Median

Preprocessing the Dataset

The first thing to do was wrangling and cleaning the data. Dropped the duplicated rows and replaced NAN values with mean value of each feature. NAN values were replaced with median and zero values replacement but the RMSE value is the best with mean replacement for the model. Mean is sensitive to outliers but may be the datasets do not have significant and performance hurting outliers.

Category data transformation

Did tried with both Pandas cat.codes got and get_dummies (one hot encoding) and cat.codes get marginal improvement on RMSE.

```
pandas.DataFrame.astype("category").cat.codes

pandas.get_dummies()
```

Building the regression machine learning models

According to the understanding the dataset section, It seem like the categorical features are seem to be pretty much important in predicting the label. The tree based algorithms were chosen to use for further experiments. Throughout the Data Science course, Random Forest method is the most exciting and interesting machine learning method for me because of it's easy to implement, highly optimizable and not very sensitive to outliers. The models also can be optimized by using various combination of hyper parameters tuning.

Scikit-learn has random forest regression algorithm implementation and it was decided to use alongside with XGBoost (Extreme Gradient Boosting), a gradient boosted trees method and LightGBM from Microsoft. The Ensemble methods such as Bagging and Gradient Tree Boosting methods are good with multiple amount of features.

Split local test data set

The training data set was split into two parts for local testing and evaluation purposes before submission for public score. The dataset was divided into 25 percent for testing dataset and remaining dataset for training the model.

Experiment with Random Forest model

Initially plugged in the preprocessed dataset, NAN replacement with mean value, to random forest model with default parameter setting. And then, the random forest model was doing a good job, it scored straight to public RMSE 36 which is lower than RMSE 40 benchmark.It seen to be that tree based models would be a good fit for the dataset.

Hyper parameters tuning of Random Forest model

The next step was tuning the hyper parameter values with grid search and cross validation technique.

```
param_dist = {
    'bootstrap': [True, False],
    'max_depth': [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, None],
    'max_features': ['auto', 'sqrt'],
    'min_samples_leaf': [1, 2, 4],
    'min_samples_split': [2, 5, 10],
    'n_estimators': [200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000, 2500]
}

random_search_rf_model = RandomizedSearchCV(
    estimator = RandomForestRegressor(),
    param_distributions = param_dist,
    n_iter = 100,
    cv = 3,
    verbose = 2,
    random_state = 42,
    n_jobs = -1)
```

This is the initial hyper parameter combinations for random forest model. It narrowed down the hyper parameters combination by using Scikit-learn's RandomizedSearchCV. After a long iterations process, the parameter values for grid search values were able to finalized.

```
    param_grid = {
        'bootstrap': [False],
        'max_depth': [None],
        'max_features': ['sqrt'],
        'min_samples_leaf': [1,2],
        'min_samples_split': [2,3],
        'n_estimators': [1000, 2000]
    }

    grid_search = GridSearchCV(RandomForestRegressor(),
        param_grid = param_grid,
        cv = 3,
        n_jobs = -1)
```

The random forest model was trained with the above parameter setting with 3-fold cross validation by using GridSearchCV from Scikit-learn. And predicted the label with test data set.
The hyper parameter tuning worked out pretty well and the tuned model was able to reduce RMSE score from 36 to 33.1.

Experiments with Gradient Boosting methods

The next method is the Gradient Boosting method. Scikit-learn has standard implementation of Gradient Boosting method. But XGBoost method was chosen for gradient boosting method..

About XGBoost method

Since the capstone project does not restricted to specific tools and technologies, it is a great opportunity to try out the new things. XGBoost method is currently the most popular and award winning boosting machine learning technology.
Getting to know this new technologies would be very helpful to me in the real world.

Experiments with XGBoost

First submission with XGBoost method with default parameter setting and scored to RMSE 32.4625. The RMSE was even better than the RMSE of tuned random forest. Then, the next thing was to try to tune the hyper parameters of XGBoost method. It was able to reduced to RMSE to 32.2 by tuning xgb hyper parameters. The tuned parameters values are as below with 3 kfolds cross validations:

```
param_grid = {
        'n_estimators': [2000, 2500],
        'gamma': [0],
        'learning_rate': [0.05],
        'colsample_bytree': [0.8],
        'subsample': [0.7],
        'max_depth': [12],
        'min_child_weight': [9],
        'seed': [1337]
}
```

Experiments with LightGBM

LightGBM is also a another popular gradient boosting method developed by Microsoft. The lightGBM model with default parameters was able to score RMSE 32.19 with cross validation with kfold value 3 which is better than current tuned hyper parameter model of xgb. Some combination of hyper parameters was tried but it cannot able to reduce from RMSE with default parameter value. So, the only parameter combination for GridSearchCV is **n_estimators**.

```
param_grid = {
        'n_estimators': [100, 1000],
}
```

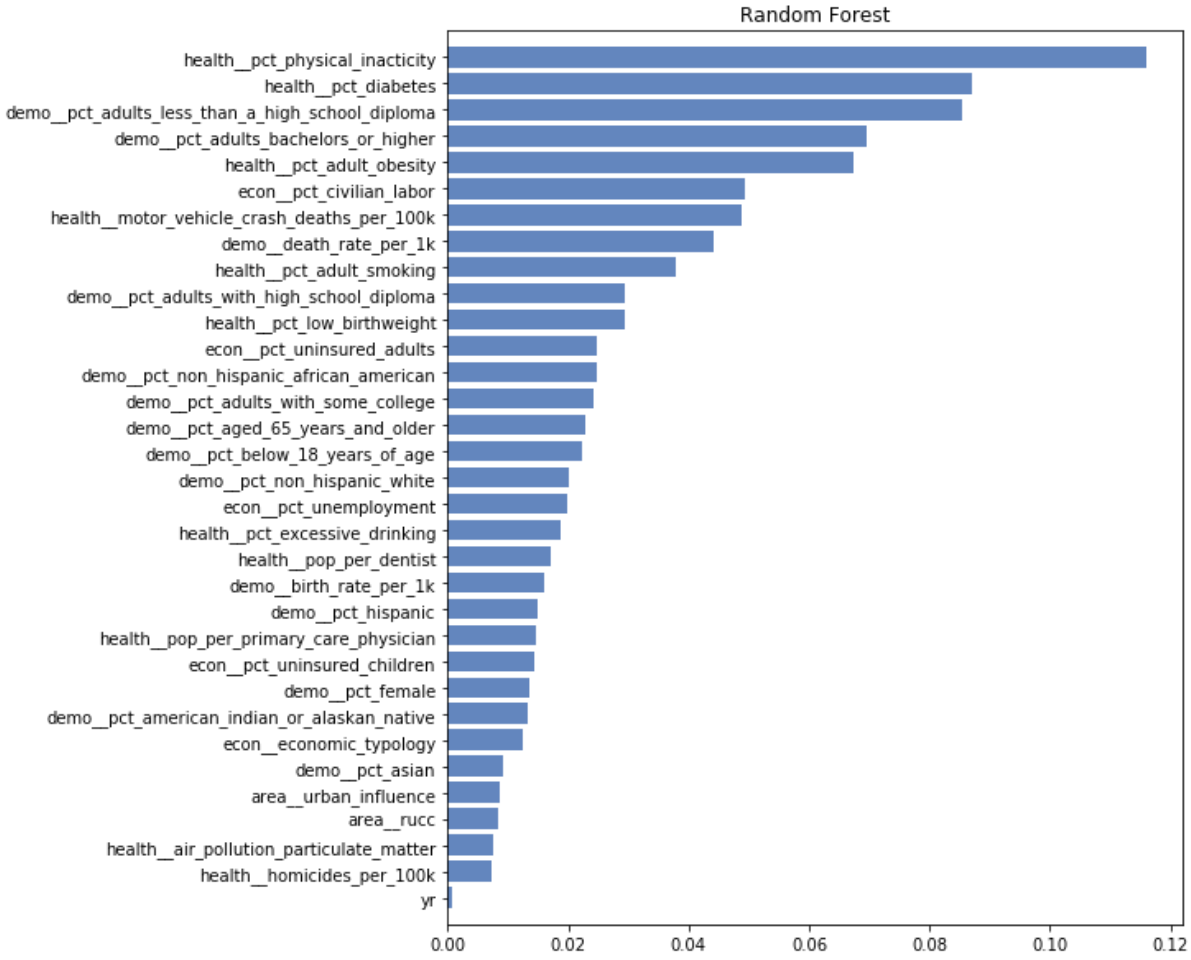
Feature importances

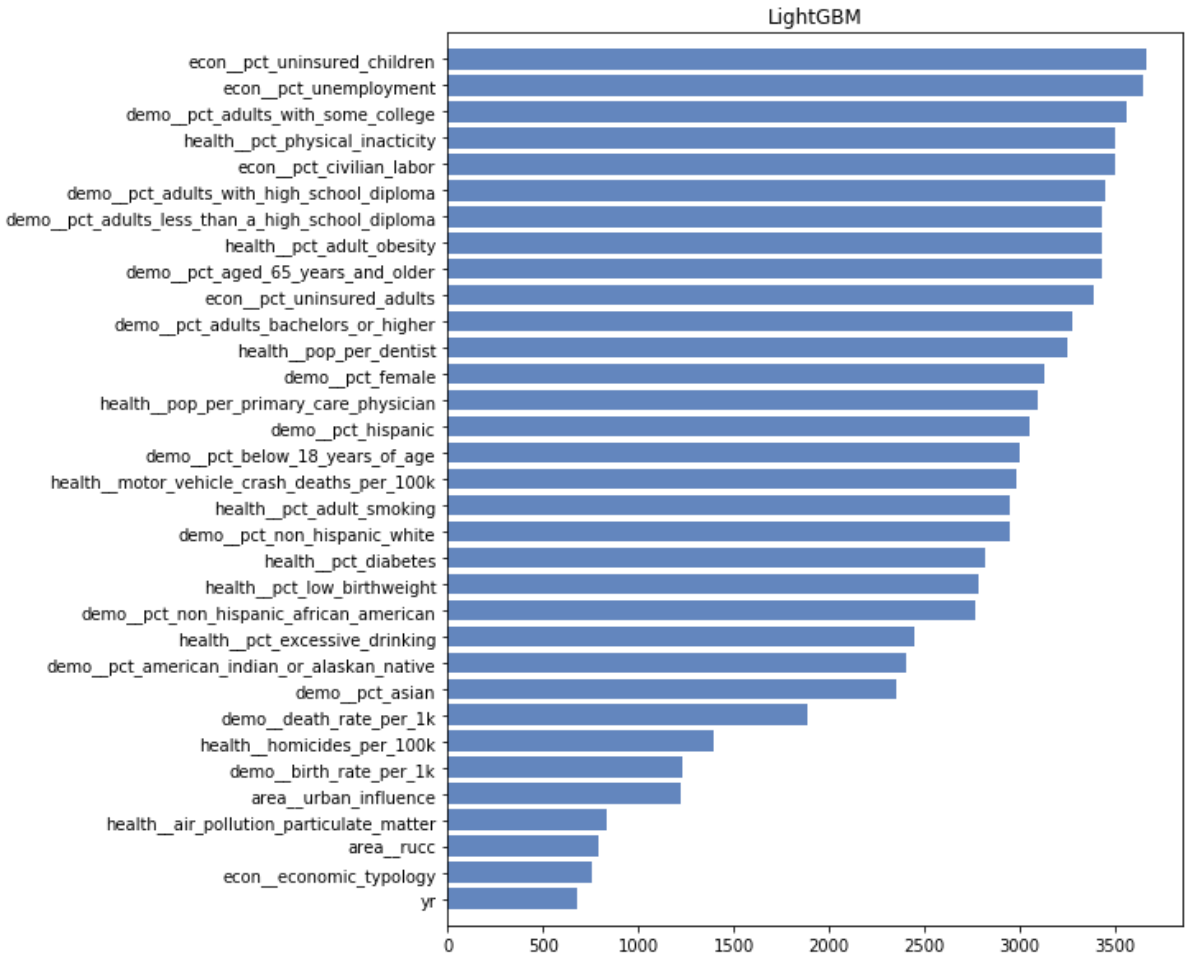
These feature importances bar plots are extracted from each previous three models which they are trained and tested by split test datasets. It can be seen in the plots, the important predictors for each model is vary.

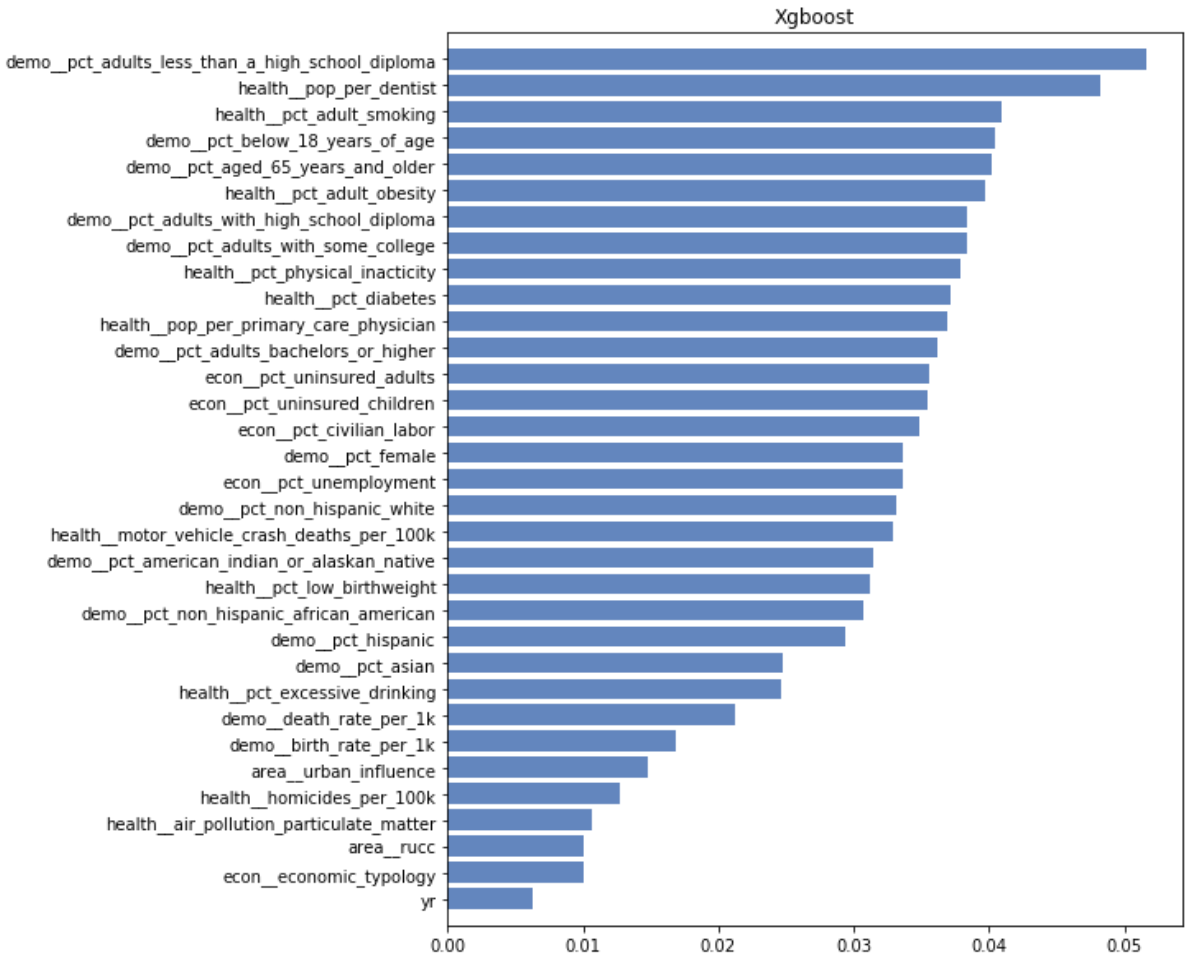
People with physical inacticity, diabetes and less than high school education are good predictors for random forest. But uninsured child and unemployments are the good predictors for lightGBM model.

And for xgboost, people with less than a high school diploma and dentist populations are good predictors.

Most of the correlated features with the label are also important predictors for each model.

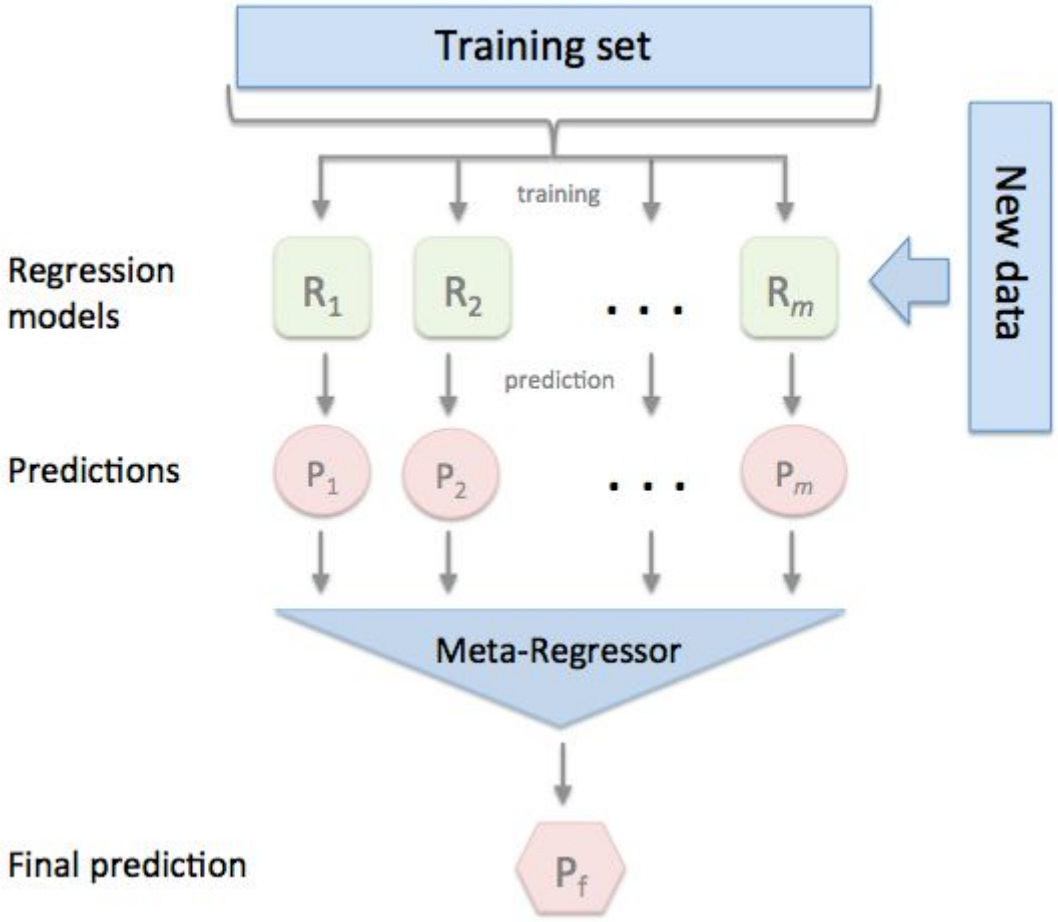



LightGBM
econ__pct_uninsured_children
econ__pct_unemployment
demo__pct_adults_with_some_college
health__pct_physical_inacticity
econ__pct_civilian_labor
demo__pct_adults_with_high_school_diploma
demo__pct_adults_less_than_a_high_school_diploma
health__pct_adult_obesity
demo__pct_aged_65_years_and_older
econ__pct_uninsured_adults
demo__pct_adults_bachelors_or_higher
health__pop_per_dentist
demo__pct_female
health__pop_per_primary_care_physician
demo__pct_hispanic
demo__pct_below_18_years_of_age
health__motor_vehicle_crash_deaths_per_100k
health__pct_adult_smoking
demo__pct_non_hispanic_white
health__pct_diabetes
health__pct_low_birthweight
demo__pct_non_hispanic_african_american
health__pct_excessive_drinking
demo__pct_american_indian_or_alaskan_native
demo__pct_asian
demo__death_rate_per_1k
health__homicides_per_100k
demo__birth_rate_per_1k
area__urban_influence
health__air_pollution_particulate_matter
area__rucc
econ__economic_typology
yr
0
500
1000
1500
2000
2500
3000
3500



The stacking regression method

The final method is called stacking regression method. This method was the key to reduce RMSE and it was able to reduce RMSE to 31 territory just a little bit. But the previous methods were struggling to reduce RMSE from 32.2. Stacking regression method has two parts: individual regressors and meta regressor. It ensemble the individual regressors to form a multiple regression models via a meta-regression. The individual regression models are trained based on the complete training set; then, the meta-regressor is fitted based on the outputs of the individual regression models in the ensemble.



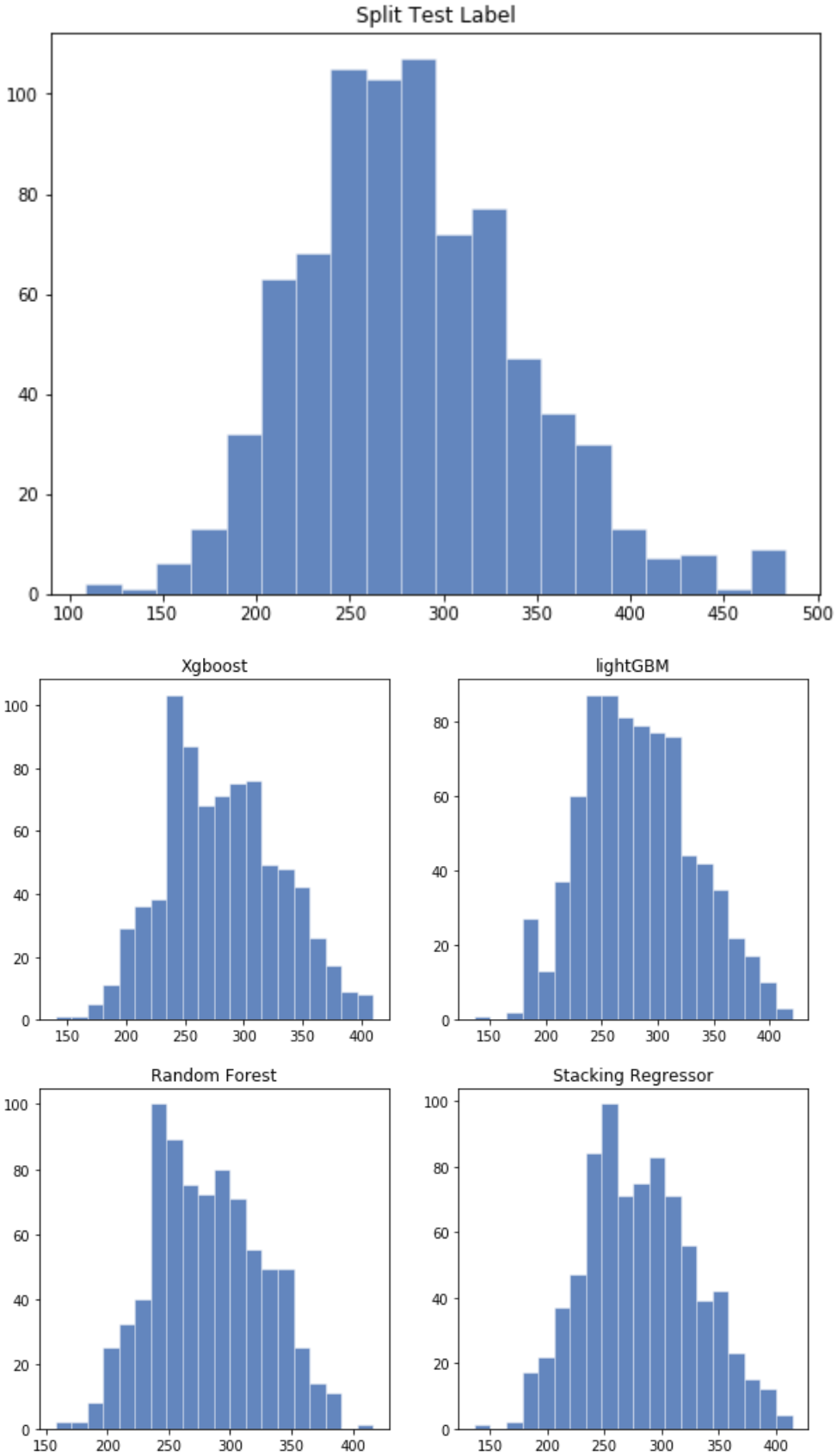
Stacking regression method form the linear combinations of different predictors to give improved prediction accuracy. The idea is to use cross-validation data and least squares under non negativity constraints to determine the coefficients in the combination.

```
stack_reg = StackingRegressor(regressors=[build_xgb(), build_lgb()], meta_regressor=build_rf())
```

So, it is not a completely new method as it is the combination of previous regressors. Xgboost and lightGBM was used as individual regressors with their best parameter combinations with cross validation with 3 kfolds. And random forest was used as a meta-regression method with its best tuned parameter as well. This technique was able to reduced to 31.95 RMSE which is my best RMSE for this capstone competition.

Conclusion

The best prediction model for in this analysis is the Stacking Regression method. it was the model which is able to score the best RMSE score for me in this journey. This technique was able to reduce lots of overfitting by using each individual models as a ensemble and the meta-regression finalized the results of the individual regressors.



The above figure is shown the comparison of test label with individual predicted result based on split test dataset. The stacking regressor plot is more resemble towards the test label.

Both xgboost and lightGBM methods were better predictors than random forest with tuned parameters. The cross-validation technique was also helpful to find out best hyper parameter combinations. And Stacking Regression technique is amazing that it combine multiple regressors to form a ultimate regression. This analysis did not use the full potential of stacking regression method and it has a lot more to do.

In my opinion, If more machine learning models such as SVM, adaboost and other linear and tree based models with individually tuned hyper parameters are added to the individual regressors of stacking regression model, it could be able to score better RMSE value.