House Insurance Renewal Declination

Introduction:

The problem explores some information regarding house insurance renewal in the spirit of identifying factors that affect the renewal declination by the insurance company. Once we identify the candidate features that play important role in declination, we will try to predict the declination percentage in any future neighborhoods.

The features we have are the likelihood of Flood, Minority population %, avg Fire reports% per 100 units, avg crime report per 1000 population, house age, median household income $. In general for a house insurance the most common factors considered by any insurance company are the location, history of insurance claims, security, house age, some information regarding the home owner, pets at home, renovations/ repairs done among others. However, there are some features that would be looked at depending on the location for instance, likelihood of flood if the construction is in a locality susceptible to flood.

It was important to note that the scales of features are very different from each other and had to be treated separately

- Target, declination% given in units 1-100
    - There are problems with a continuous variable with bounded range as target. It had been dealt with accordingly using methods to transform the given labels to fit a model's requirements
- Other features are scaled to match that of the target though this scaling only matters to models utilizing any distance measure to function or the ones like linear regression

This problem can be considered as a classification or regression without further thought but could be proven wrong. All that because of the nature of label i.e, proportion or percentage. A linear regression model would work perfectly since the data at hand is continuous but the limitation would be the strict scale of target. Irrespective of how well a linear regression model or any regression model for that matter (with very few exceptions) would have unbounded range. They can predict a value anywhere on the number like without bounds.

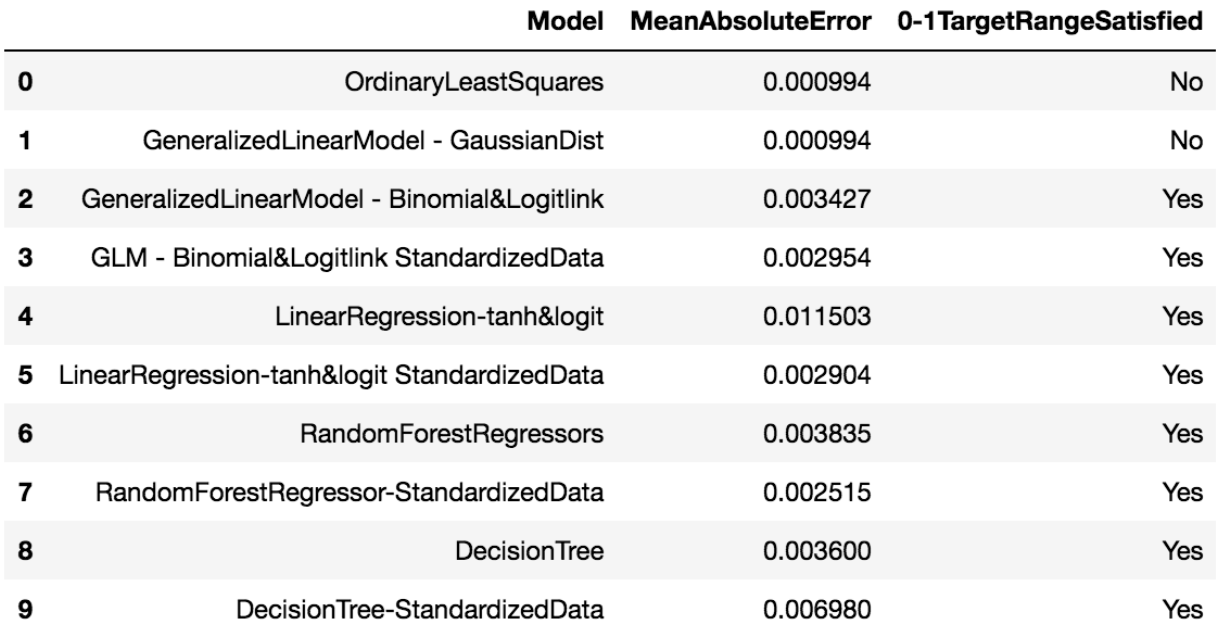
For the same reason specified above can we consider the label a field with multiple classes? We sure can, given there's a strict limit on the possible digits after decimal in a percent value like 1.02, 2.234… the problem arises as the number of digits after decimal increase since the number of possible classes will be one for each possible number. Let's say only 2 digits are allowed after decimal 10.25, 0.43 then the number of classes for a classification problem would be 100 * 100 since each interval of 1 has 100 possible values which would be too many.

Another option would be to bin the percentages into buckets but that would result in information loss and hence not been adapted.

So, the problem had been solved as a regression problem in a bounded region. For this reason, the metric selected to measure performance of any model is mean_absolute_error. We are interested in identifying how far the predictions are from actual target and to maintain the same scale without scaling up or down.

House Insurance Renewal Declination

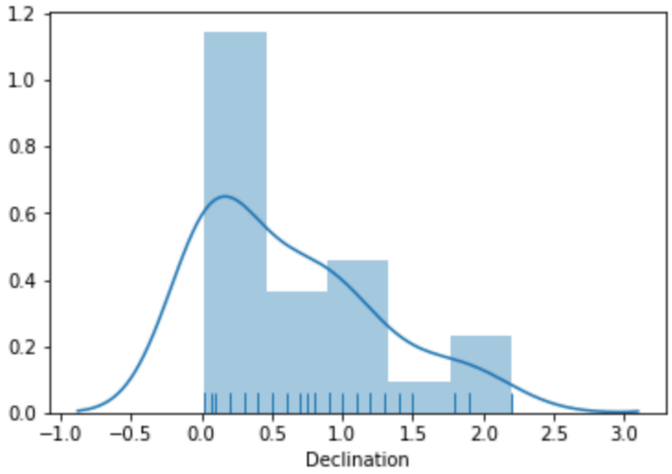
Summary results:

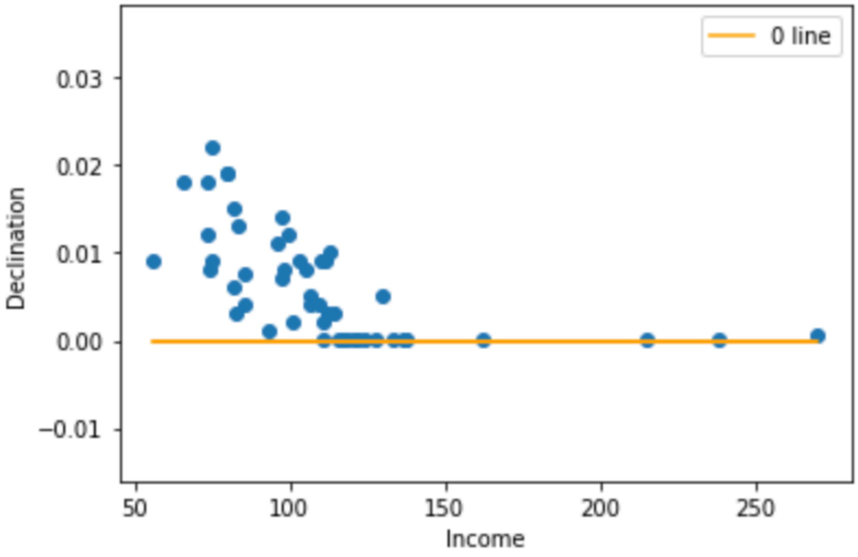


	Model	MeanAbsoluteError	0-1TargetRangeSatisfied
0	OrdinaryLeastSquares	0.000994	No
1	GeneralizedLinearModel - GaussianDist	0.000994	No
2	GeneralizedLinearModel - Binomial&Logitlink	0.003427	Yes
3	GLM - Binomial&Logitlink StandardizedData	0.002954	Yes
4	LinearRegression-tanh&logit	0.011503	Yes
5	LinearRegression-tanh&logit StandardizedData	0.002904	Yes
6	RandomForestRegressors	0.003835	Yes
7	RandomForestRegressor-StandardizedData	0.002515	Yes
8	DecisionTree	0.003600	Yes
9	DecisionTree-StandardizedData	0.006980	Yes

House Insurance Renewal Declination

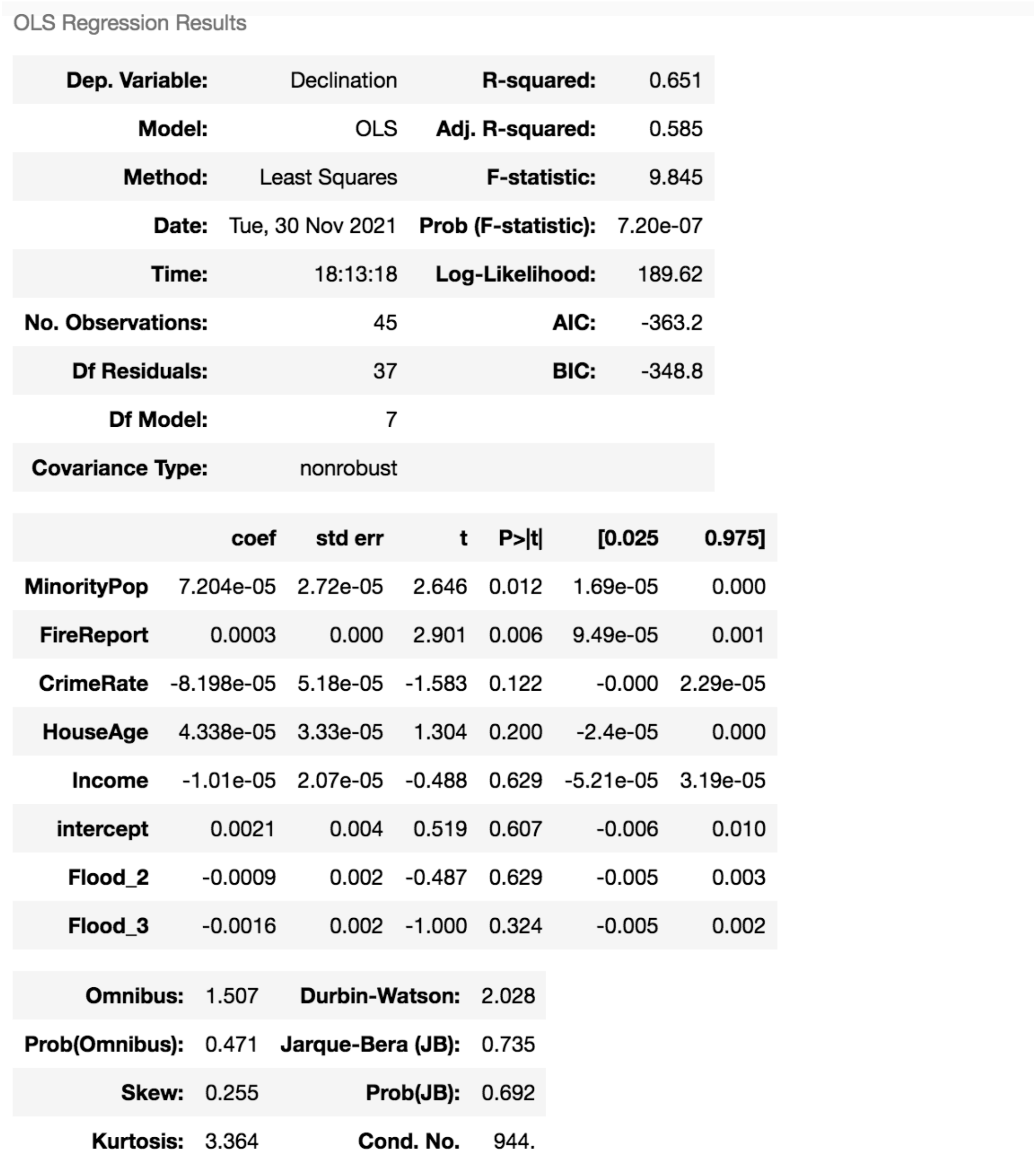
Model Development:
A lot of this problem has to do with the type of target variable and identifying what models work best with this variable. For this reason, the distribution of the target has been looked at, though it looks like a right skewed tail of normal distribution, the bound of 0 and 100 makes it different.
Since it was continuous, Linear regression model had been tried first.
The target column was divided by 100 to obtain all values between 0 and 1





House Insurance Renewal Declination

I modeled a linear regression problem whose summary is below



OLS Regression Results

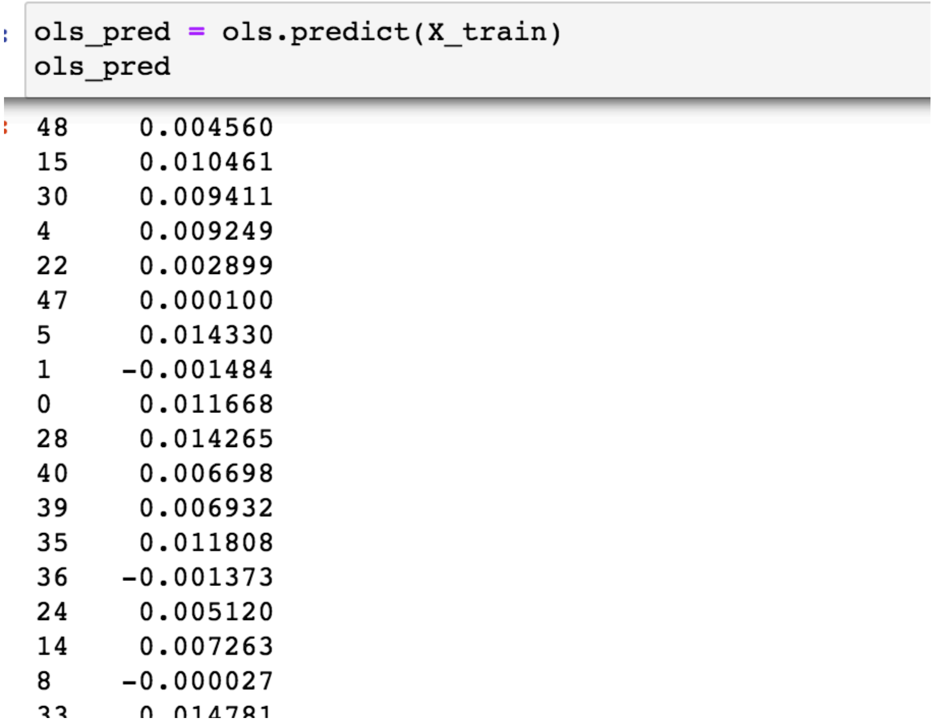
Dep. Variable:	Declination	**R-squared:**	0.651
Model:	OLS	**Adj. R-squared:**	0.585
Method:	Least Squares	**F-statistic:**	9.845
Date:	Tue, 30 Nov 2021	**Prob (F-statistic):**	7.20e-07
Time:	18:13:18	**Log-Likelihood:**	189.62
No. Observations:	45	**AIC:**	-363.2
Df Residuals:	37	**BIC:**	-348.8
Df Model:	7		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
MinorityPop	7.204e-05	2.72e-05	2.646	0.012	1.69e-05	0.000
FireReport	0.0003	0.000	2.901	0.006	9.49e-05	0.001
CrimeRate	-8.198e-05	5.18e-05	-1.583	0.122	-0.000	2.29e-05
HouseAge	4.338e-05	3.33e-05	1.304	0.200	-2.4e-05	0.000
Income	-1.01e-05	2.07e-05	-0.488	0.629	-5.21e-05	3.19e-05
intercept	0.0021	0.004	0.519	0.607	-0.006	0.010
Flood_2	-0.0009	0.002	-0.487	0.629	-0.005	0.003
Flood_3	-0.0016	0.002	-1.000	0.324	-0.005	0.002

Omnibus:	1.507	**Durbin-Watson:**	2.028
Prob(Omnibus):	0.471	**Jarque-Bera (JB):**	0.735
Skew:	0.255	**Prob(JB):**	0.692
Kurtosis:	3.364	**Cond. No.**	944.

House Insurance Renewal Declination

The summary looks like it has no problem, but the actual issue reveals itself after I made predictions. As shown below the predictions



```
ols_pred = ols.predict(X_train)
ols_pred
```

```
48    0.004560
15    0.010461
30    0.009411
4     0.009249
22    0.002899
47    0.000100
5     0.014330
1    -0.001484
0     0.011668
28    0.014265
40    0.006698
39    0.006932
35    0.011808
36   -0.001373
24    0.005120
14    0.007263
8    -0.000027
33    0.014781
```

Our predictions shouldn't be negative, but we clearly see them. So OLS is not a good option though it has very low mean_squared_error.

Then a GLM was tried, with a Gaussian distribution which again worked exactly like an OLS model since ofcourse we worked with a Gaussian distribution because the data is continuous in range 0-1.

Now, we can still use a model like GLM but the only catch is we have to make sure that there is a transformation that would give us the output in the range expected. One such functions is the Logit. It works well with the Binomial family with a link Logit. It works because the target has only the probability of declination which for 1 data point is a Bernauli distribution and for a series of data points is a Binomial distribution and hence Family Binomial with Link Logit.

This worked like a charm and even performed well enough to give a mean absolute error of `0.003427252359408609`

- Linear Regression with Logit
    - This is very similar to GLM but the application of Logit had been done manually
    - To ensure no infinite values, I had used the *tanh* function to ensure that there aren't extremes to the inputs of Logit
    - This wasn't the best of all performers even with the rest of data standardized

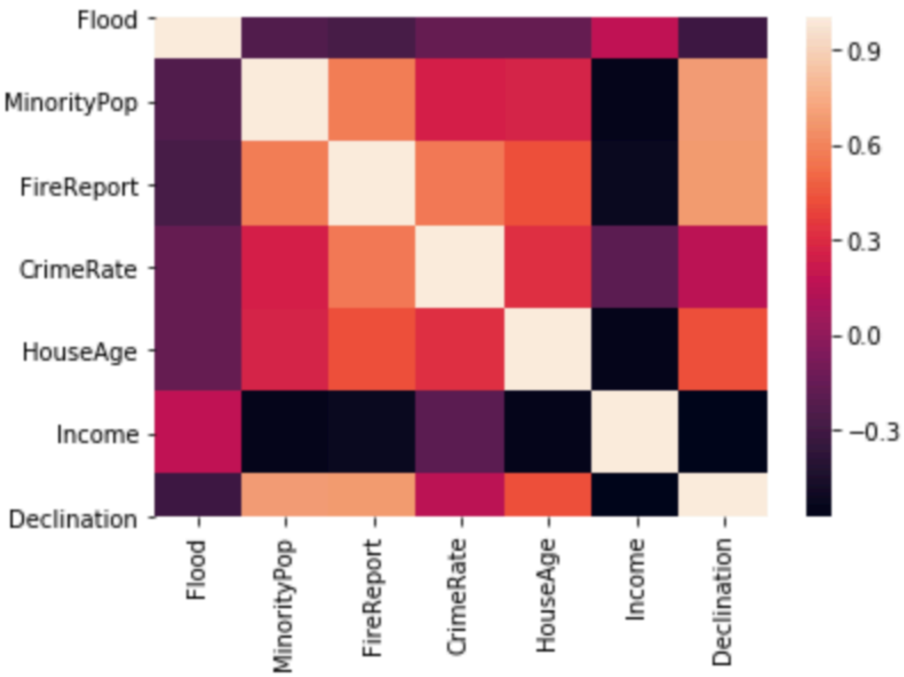
House Insurance Renewal Declination

Tree based models also work well with bounded target variables since they work based on rules and they don't build a line or plane across which predictions occur and hence the next models tried were Decision Trees and Random Forest for regression.

Both of these algorithms were run with and without standardized input data. The best results were for Random Forest Regressor with standardized input. It had a mean absolute error of around 0.0025 which was also the best of all algorithms tried out.

Decision trees performed better without standardized data than with it. The performance worsened a little which wasn't ideal.

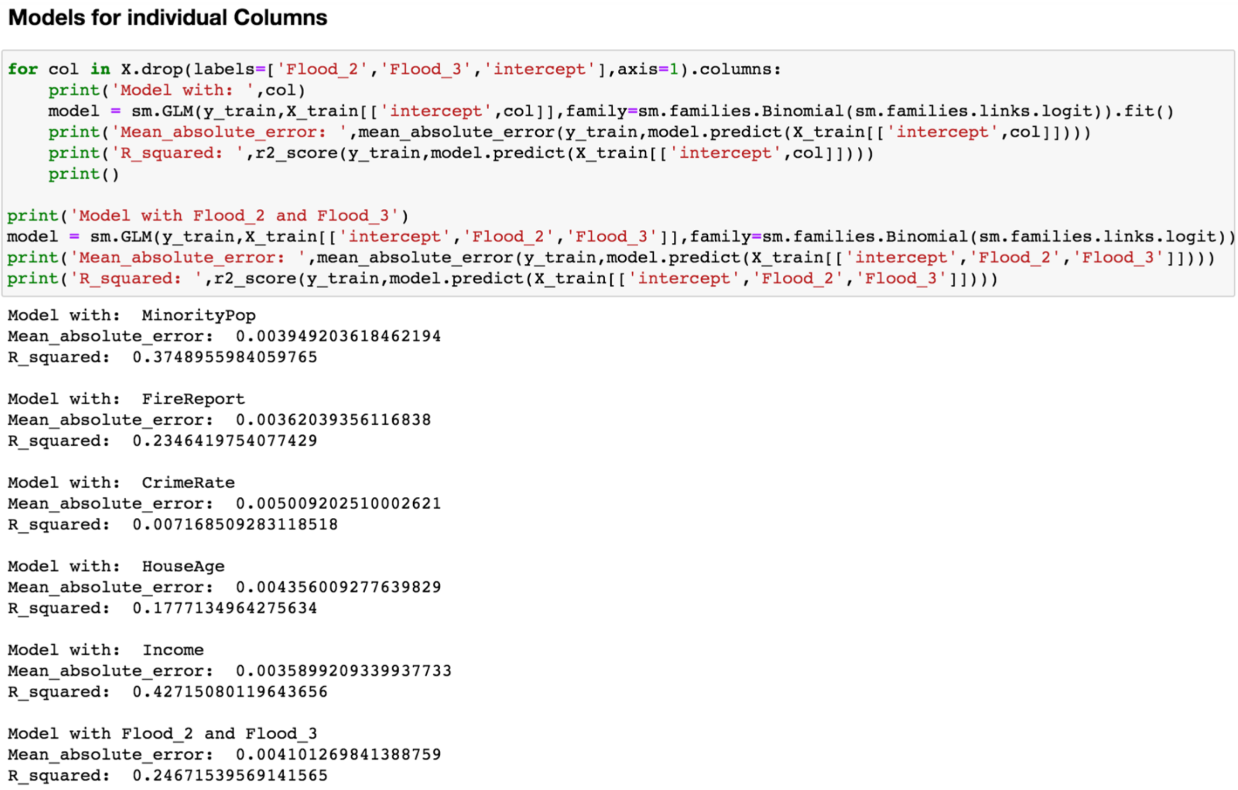
Now, the main reason for this analysis was to identify factors that played an important role in identifying declination percent in a given neighborhood. So, it started with correlations. Declinations had high correlation with almost all variables except for the crime rate which is counter intuitive since security of a house is also a factor considered almost always to determine if a house is worthy of insurance.



Another thing to note here is that there is a considerable negative correlation between Declination and Salary(-57.5%). This might not be the most obvious relation, there is a strong possibility of a confounding variable involved here. Higher salary – in most cases – results in a better house in a better locality and a better locality in return will have lesser negative aspects allowing for a lower declination in insurance renewal.

One more aspect that's worthy of discussion is the negative correlation of declination with Flood. Once again the relation might not be obvious but after some search regarding the insurance and affecting factors an understanding that seemed sensible was that, though there is an area that has some chance of flood insurance companies allow them to renew insurances at a higher price which could also be reasonable since the companies might have to pay out huge sums, in case. Since the house is anyway in a region of possible risk the home owners are also inclined to buy insurances at higher prices.

Then each independent feature was used to build it’s own model to predict the declination percentage.



Models for individual Columns

```
for col in X.drop(labels=['Flood_2','Flood_3','intercept'],axis=1).columns:
    print('Model with: ',col)
    model = sm.GLM(y_train,X_train[['intercept',col]],family=sm.families.Binomial(sm.families.links.logit)).fit()
    print('Mean_absolute_error: ',mean_absolute_error(y_train,model.predict(X_train[['intercept',col]])))
    print('R_squared: ',r2_score(y_train,model.predict(X_train[['intercept',col]])))
    print()

print('Model with Flood_2 and Flood_3')
model = sm.GLM(y_train,X_train[['intercept','Flood_2','Flood_3']],family=sm.families.Binomial(sm.families.links.logit))
print('Mean_absolute_error: ',mean_absolute_error(y_train,model.predict(X_train[['intercept','Flood_2','Flood_3']])))
print('R_squared: ',r2_score(y_train,model.predict(X_train[['intercept','Flood_2','Flood_3']])))
```

```
Model with:  MinorityPop
Mean_absolute_error:  0.003949203618462194
R_squared:  0.3748955984059765

Model with:  FireReport
Mean_absolute_error:  0.00362039356116838
R_squared:  0.2346419754077429

Model with:  CrimeRate
Mean_absolute_error:  0.005009202510002621
R_squared:  0.007168509283118518

Model with:  HouseAge
Mean_absolute_error:  0.004356009277639829
R_squared:  0.1777134964275634

Model with:  Income
Mean_absolute_error:  0.0035899209339937733
R_squared:  0.42715080119643656

Model with Flood_2 and Flood_3
Mean_absolute_error:  0.004101269841388759
R_squared:  0.24671539569141565
```

FireReport had the highest mean_absolute_error followed by CrimeRate and HouseAge. The least mean_absolute_error or the best performing feature not surprisingly is the Income. Moreover, the amount of variance captured is also highest for the Income field and like we saw in the correlation plot crime rate has least amount of variance captured.

RandomForest and Decision Trees are also used to determine the features with promise. On that note, RandomForest regressor in it’s best performing model had not used Minoritypopulation % at all. The weight for that feature in that model was 0. Decision tree on the other hand for it’s best performing model at 0.0038 mean absolute error had not made use of the Flood column.

After all this analysis, we were left with a randomforest model having minor changes in the parameters and standardized input data as the best performing regressor standing at a mean absolute error of 0.0025.

House Insurance Renewal Declination

Conclusion and Limitations:

- Linear and Logistic models work fine with percentage/proportions in target as long as they have a proper transformation and link functions
- Though the performance is not the best in class, linear and logistic still played their part in recognizing potential features for a final model
- We could also have modeled the target variable with a beta distribution with parameters alpha and beta changes in which would cause the distribution to take a myriad of shapes including bell, skewed bell, exponential, parabolic in within the bounds of 0 and 1
- A beta regression is a potential option to work with proportions in target and remains an avenue to explore as far as this exercise is concerned
- Tree based models are the best performing ones since they aren’t concerned about the range or bounds of target and work with their own set of rules to predict a target
- Even out of the Tree based models, Random Forest in particular had the best performance and further tuning it’s parameters could lead to potential improvements
- An extension for this work could be to explore methods for parameter tuning including but not limited to grid search
- Finally, irrespective of how well our models perform on this data it has to be noted that there’s only 50 data points and the result could at best serve as a direction for future analysis