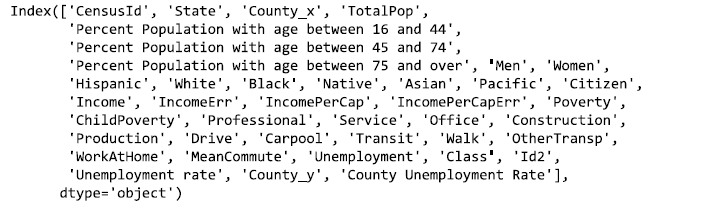
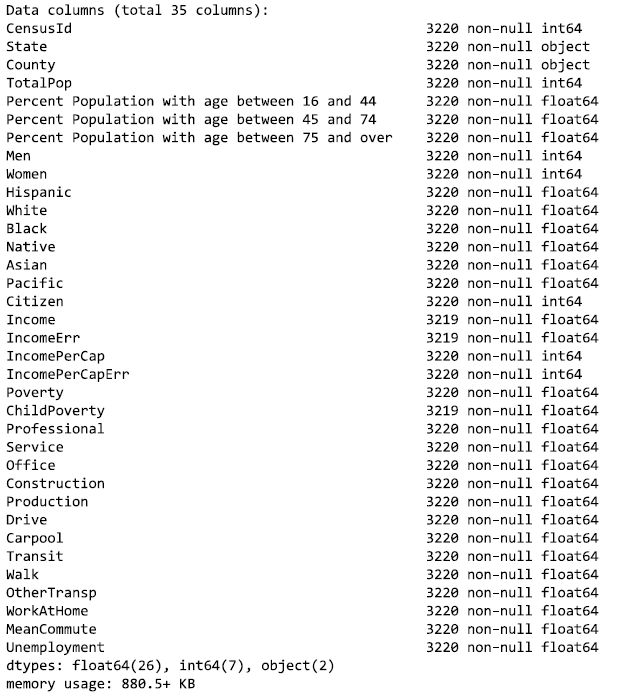
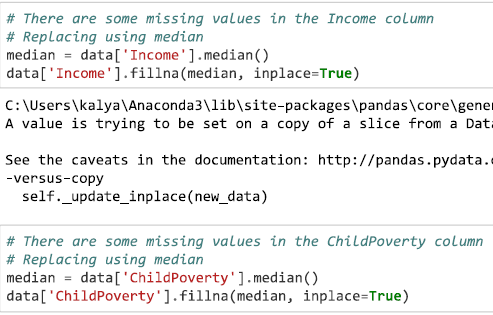
Used OLS module of statsmodels library of python to build linear regression models and pandas library for data analysis.

1. For the final project of my Data science course, I worked on the 2015 County Demographics and unemployment rates dataset.
2. The objective of the project is to predict the unemployment rates of the counties for the year 2017 based on the unemployment rates of counties in the year 2015 using linear regression
3. The dataset has the following columns:

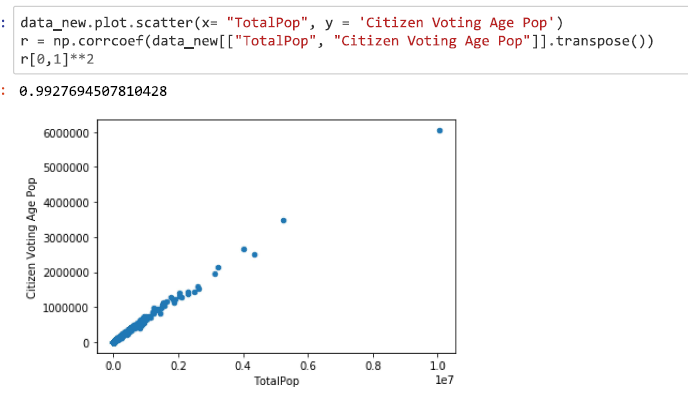


1. Solution Flow Chart:
   * 1. First, I processed the datasets to identify any missing values and replaced them with either mean or median

The columns Income and Child poverty have missing values. I replaced them with the median of their other values:

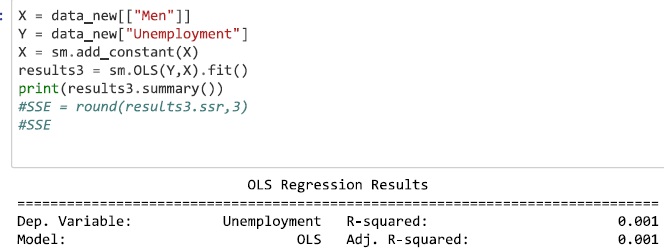
* + 1. I removed attributes that cause multicollinearity (Correlation between the predictors itself).



The variable Citizen Voting age population and total age have a strong correlation(0.99). So, I removed one of them:



* + 1. I explored the data and removed any irrelevant or redundant variables.

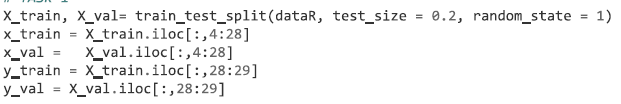


The model with the “Men” variable has a very small R2 value. So, I dropped it:



* + 1. I partitioned the dataset as training dataset and the test dataset using the hold-out method with

80% percent of data as training and 20% as validation data:



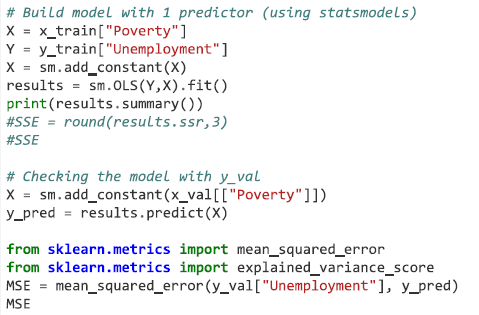
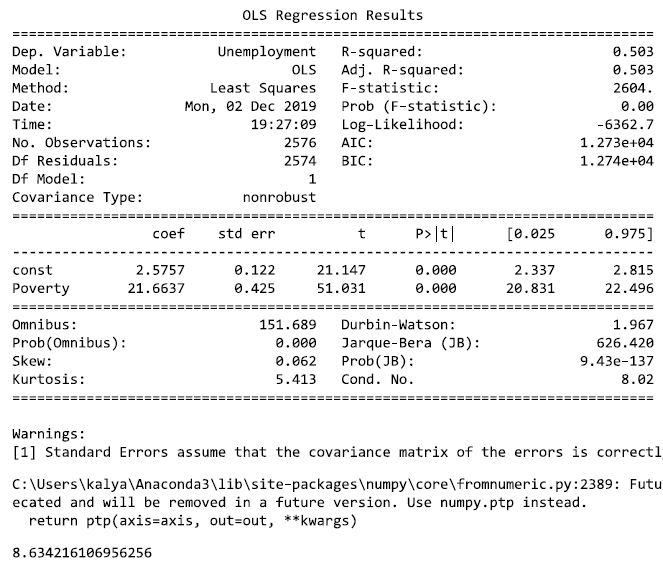
* + 1. I standardized both training and validation datasets:



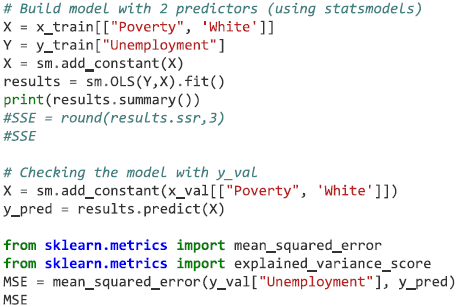
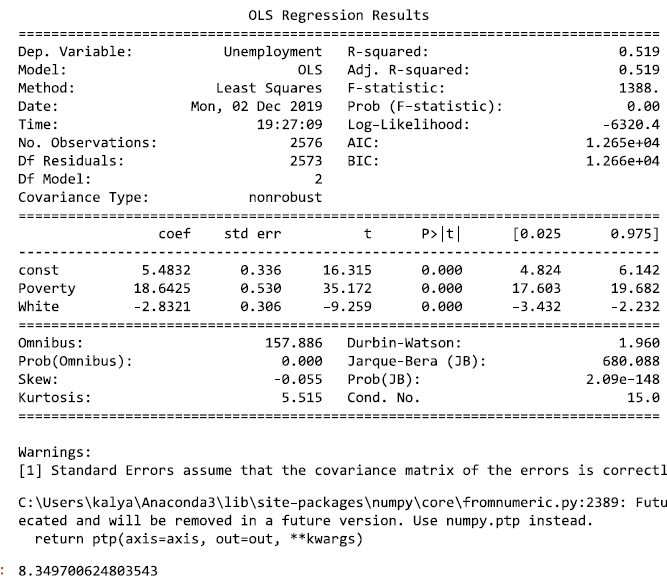
* + 1. I built linear regression models on the 2015 data set using the forward feature selection method and picked the best model based on the root mean squared errors on the test dataset.

Note: In the forward selection method, I first start with the null model (no variable only the intercept which would be either mean or median of the values of the response variable). Then, I build models with a single feature/variable and pick the best model based on the mean squared errors on the validation test dataset. Then, I build models with each of the remaining q-1 variables and select the best two-variable model. I continue like that until I left with no variables.

Theoretically, the best subset selection (checking every possible subset of feature) should give the best model but there is a computational problem associated with it. It is difficult to check all the 2­­p subsets.

The MSE (Mean squared error) on the test dataset of the above model with a single predictor = 8.63

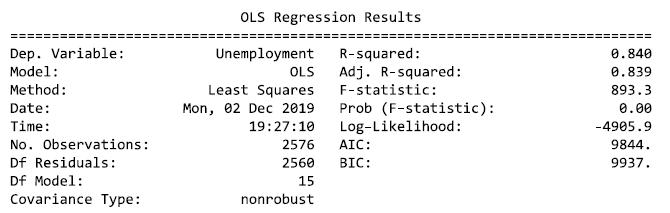
 

The MSE (Mean squared error) on the test dataset of the above model with two predictors = 8.4.

So, I kept the second variable and added the remaining variables one by one and kept the variables whose inclusion reduced the Mse value.

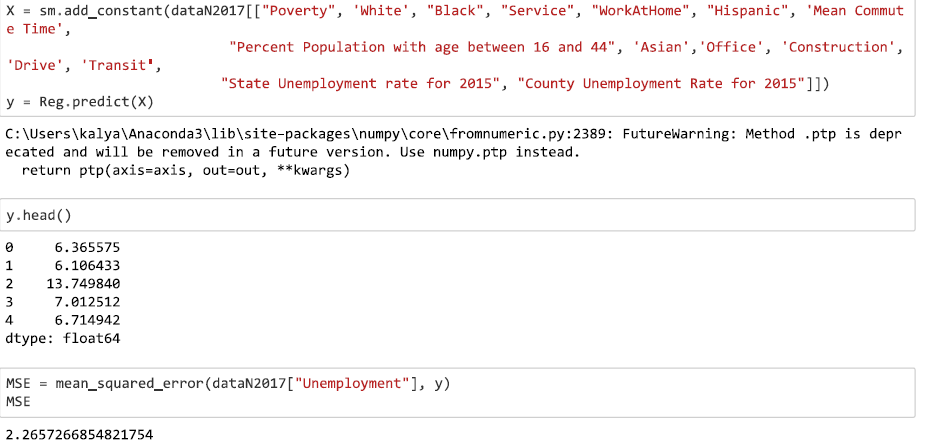
* + 1. I obtained the best model that has 14 predictors out of 40:







* + 1. I applied the above model on the 2017 dataset, predicted the county unemployment rates. I got the mean squared error of 2.26 from the actual unemployment rates.



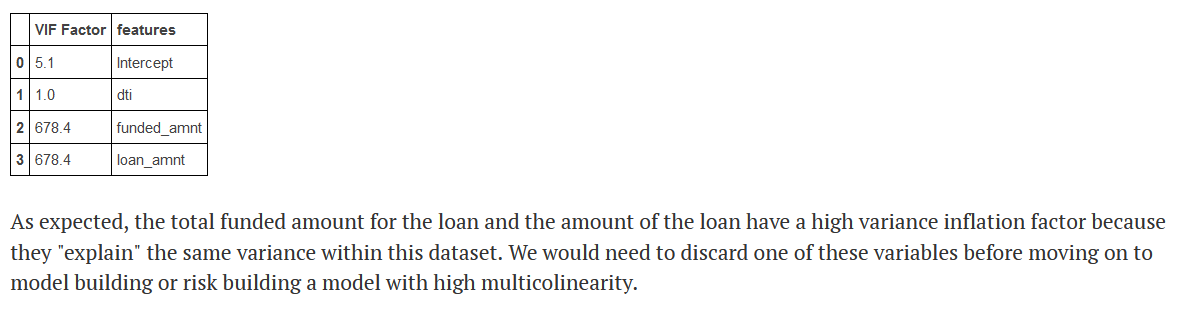
**Better Approaches:**

1. Instead of manually removing correlated predictor variables, we can use the Variable Inflation factor of statsmodels library of python which gives the VIF score for each of the variables.

The Variance Inflation Factor (VIF) is a measure of collinearity among predictor variables within a multiple regression. Steps for Implementing VIF:

1. Run a multiple regression.
2. Calculate the VIF factors.
3. Inspect the factors for each predictor variable, if the VIF is between 5-10, multi-collinearity is likely to present and you should consider dropping the variable.

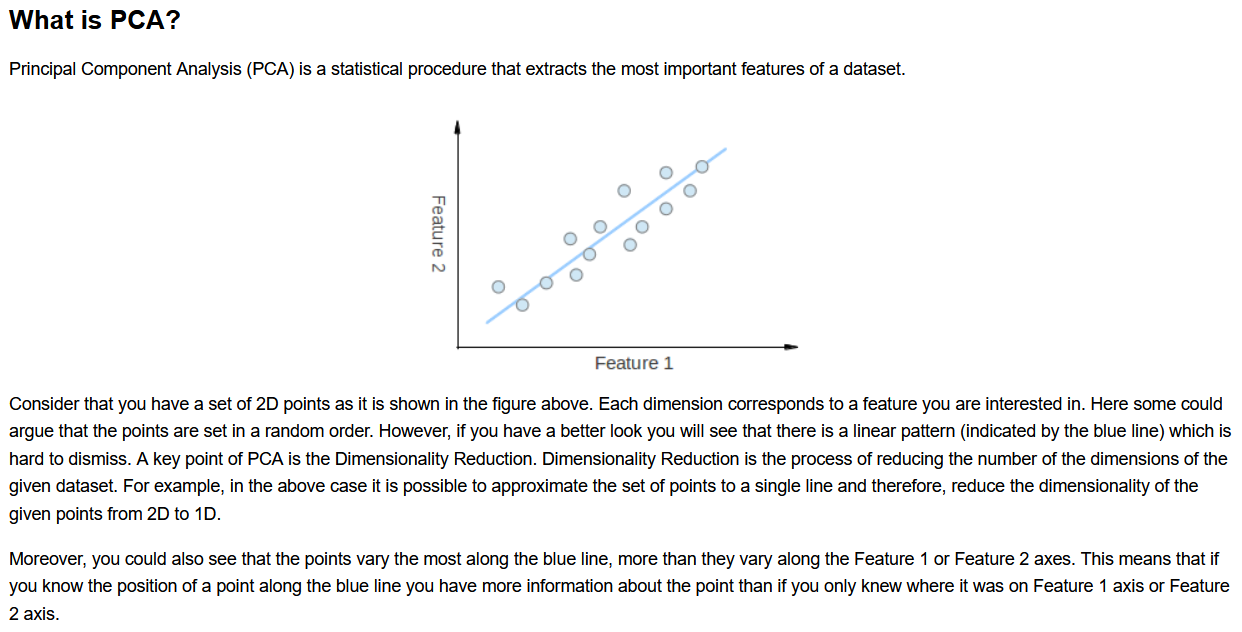
Ex:



Ref: <https://etav.github.io/python/vif_factor_python.html>

1. And instead of using forward selection methods to find the best variables, we can use dimensionality reduction techniques like Principal Component Analysis (PCA). These techniques represent a given dataset in fewer dimensions while retaining the variance of the original dataset.

Principal component analysis (PCA) is a technique for reducing the dimensionality of large datasets, increasing interpretability but at the same time minimizing information loss. It does so by creating new uncorrelated variables that successively maximize variance.



The blue line above is a new artificial feature found by the PCA technique. That line is good enough to represent all the data points. Here, we converted the 2-dimensional dataset into the 1-dimensional dataset (dimensionality reduction)

Ref: <https://towardsdatascience.com/principal-component-analysis-pca-from-scratch-in-python-7f3e2a540c51>

<https://en.wikipedia.org/wiki/Principal_component_analysis>

Video link: <https://www.youtube.com/watch?v=FgakZw6K1QQ>

**Note:** For another project of my data science course, I predicted the votes of Republican and Democratic parties (two response variables) for counties of the US in 2018. I built the Linear regression models on the 2018 demographics of the US counties in a similar fashion.