Course Project - IME692

Ravi Kumar(180594)

October 22, 2021

Estimating Covid-19 Vaccine Disparity in the US

Acknowledgements

We wish to express our sincere gratitude to our instructor **Dr. Shankar Prawesh** for his valuable advice and guidance in completing this project. The way he presented each and every topic in the class made the topics very interesting and understandable, which helped a lot in making our project possible.

Introduction :

We have been given the dataset collected to examine the relationship between social determinants of health and racial disparities in covid vaccination at the country level in the US. Our task is to predict the variable **CvdVax_DisparityY** using the other features from the dataset. The data points with **test = 1** have been used for testing the models, while with **test = 0** have been used for training purposes.

Visualization of the Dataset:

The **training** dataset has **531** rows, while the **test** dataset comprises **255** data points. We have a total of **18 features,** including the target variable.

Class	Number of Points
Y=1	255
Y=0	276

The above table shows that the data distribution among the two classes is more or less the same, and there is no class imbalance. To get the correlation between any two variables - X1 and X2, we know -

$$Corr(X_1, X_2) = \frac{\sum_N (x_{1i} - \bar{x}_1)(x_{2i} - \bar{x}_2)}{\sqrt{\sum_N (x_{1i} - \bar{x}_1)^2 \sum_N (x_{2i} - \bar{x}_2)^2}}$$

where N=531 and X1 and X2 can be any variable between x1 to x20. After getting the correlation matrix, we observe that variables are highly uncorrelated and all correlation values < 0:3 for any two variables.

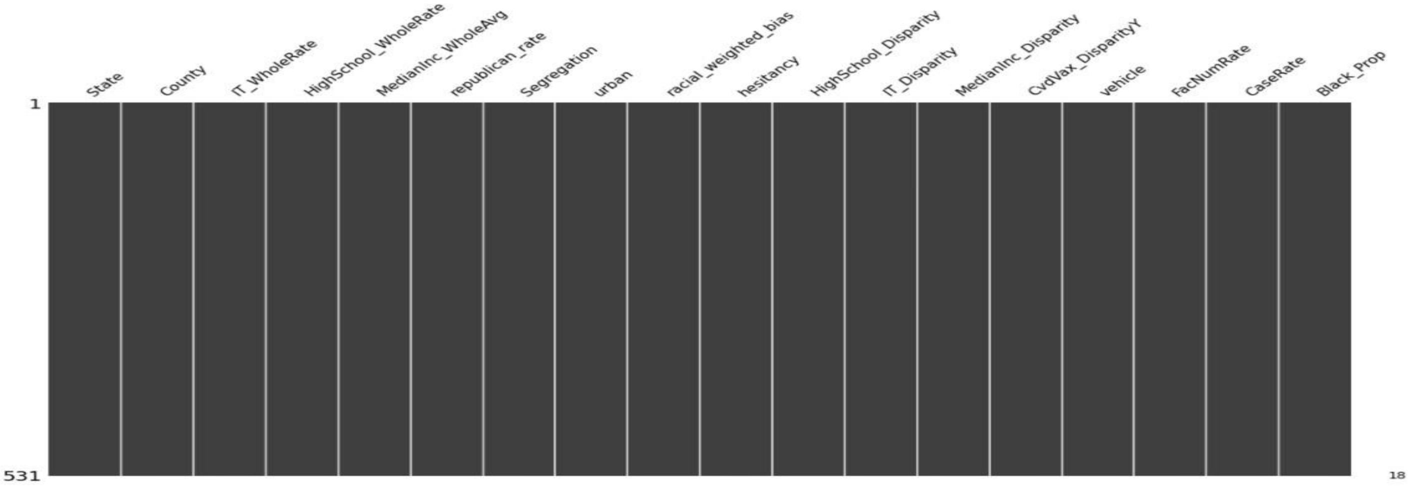


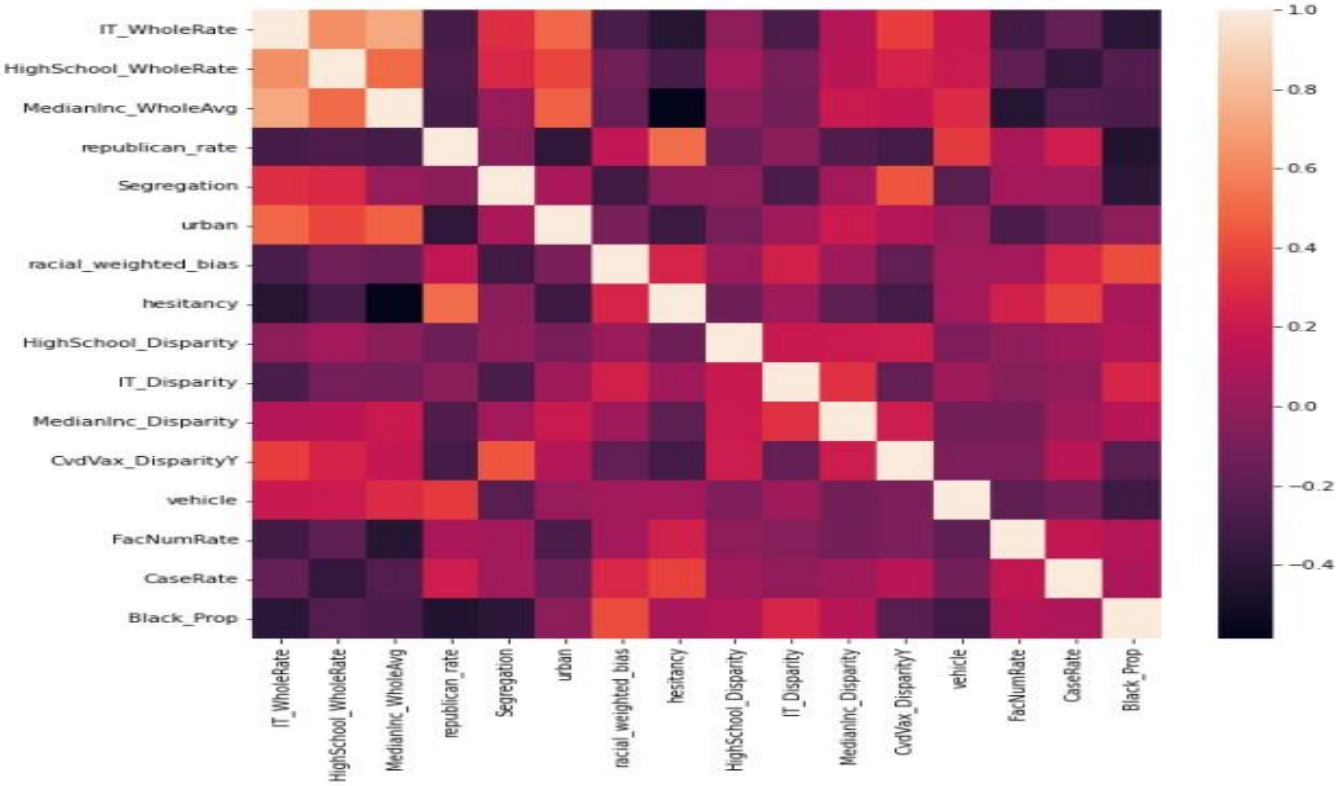
Fig: shows us there is no data missing

Salient Features Of Dataset

The dataset has **18** features

- **State** - State in which the county is located
- **County** - County name
- **IT_WholeRate** - County-level computer ownership and internet subscription data (in percentage)
- **HighSchool_WholeRate** - County-level education data. Percentage of population that qualifies as a high school graduate or higher
- **MedianInc_WholeAvg** - County-level household median income data
- **republican_rate** - The share of votes cast for the Republican candidate in the 2020 election
- **Segregation** - Black-White segregation index measures. This index ranges from 0 (complete integration) to 100 complete segregation
- **urban** - A dummy-coded variable representing whether a county is considered urban or rural.
- **racial_weighted_bias** - Measure indicating implicit racial bias in a county. Larger values indicate greater bias against Blacks.
- **hesitancy** - Survey response for overall vaccine hesitancy in a county
- **HighSchool_Disparity** - Difference in county-level high school education between White and Black population
- **IT_Disparity** - Difference in county-level computer ownership and internet subscription between White and Black population
- **MedianInc_Disparity** Difference in county-level median income between White and Black population
- **CvdVax_DisparityY** Difference in the Covid-19 vaccination rate (first dose) between White and Black residents in a County (in percentage).
- **vehicle** Proportion of vehicle ownership in a county
- **FacNumRate** Number of health care facilities per capita
- **CaseRate** - Covid-19 cases per capita
- **Black_Prop** - Proportion of black residents in a county

Our task is to estimate the variable **CvdVax_DisparityY** using the other predictor variables



Correlation Heat Map

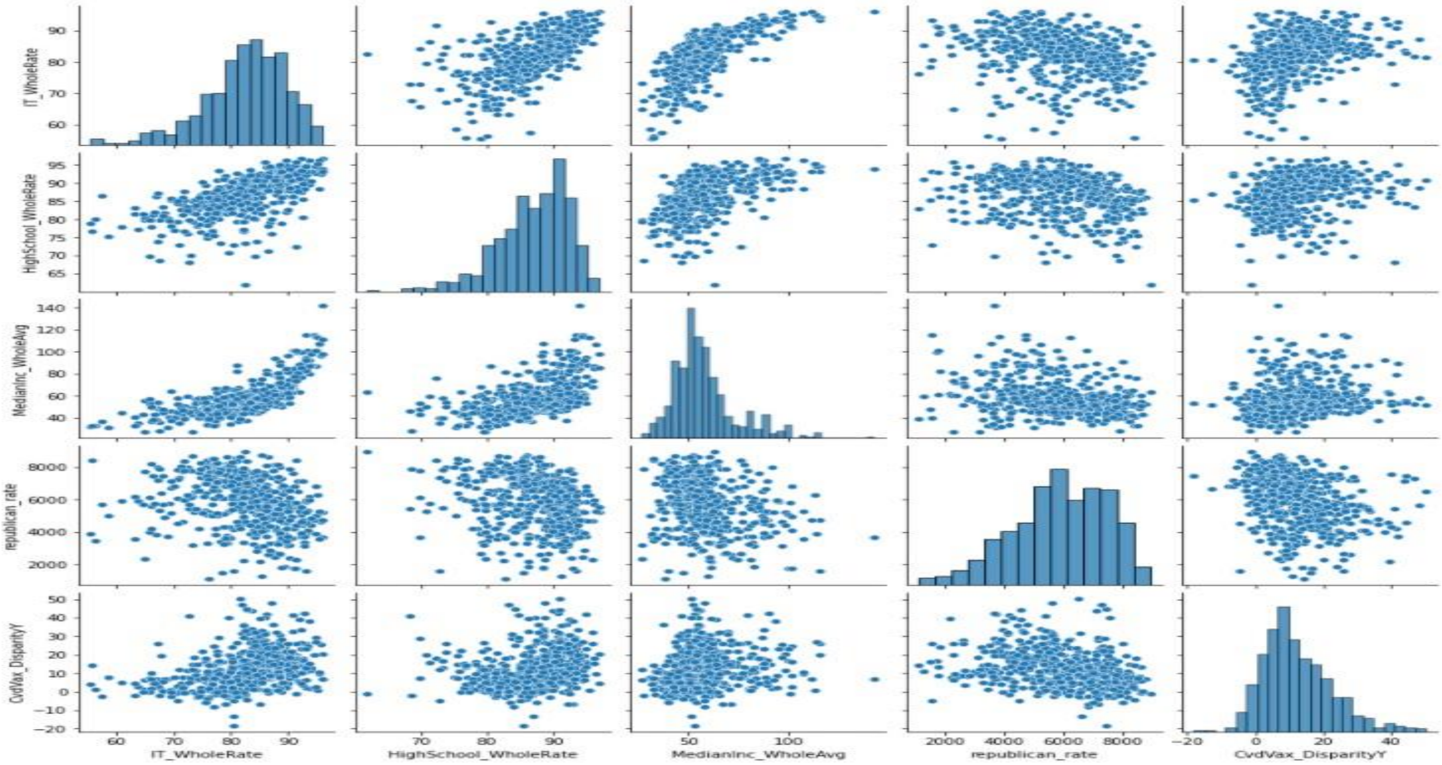


Figure 1: Pair plot of variables - x1 to x5. (Similar trend is observed for other variables)

In the preliminary analysis, we can observe that :

- **IT_WholeRate**, **HighSchool_WholeRate**, **MedianInc_WholeAvg** are highly correlated to each other
- **CvdVax_Disparity** seems to be related to factors of social welfare like **IT_WholeRate** and **MedianInc_WholeAvg**
- It is also related to the measure of **Segregation** and percentage of **Black Population** and the disparity in **High School** and **Income** rates
- Interestingly the covid disparity also seems to be negatively correlated to the **republican_rate**

Data Preprocessing

- There is neither any missing value nor any noisy data in our dataset.
- As we have only one file of dataset so there is no need of data integration.

- We plotted the pair plot of all the possible pairs to visualize the dependency of one attribute over another so that we can decrease the no. of attributes. Also, we plotted the heatmap of the correlation matrix. Most of the pixels are dark pink in colour, which means most of the attributes are independent of each other. Some are of positive correlation and some are negatively correlated. But Pearson's product coefficient of all lies between (-0.5 to +0.5). Hence, we are not removing any attribute in this step.

We have used the train test split function from Scikit-learn library to split the training dataset into 70% train and 30% test set. Sklearn (or Scikit-learn) is a Python library that offers various features for data processing that can be used for classification, clustering, and model selection. Selecting a proper model allows us to generate accurate results when making a prediction. To do that, we need to train our model by using a specific dataset. Then, we validate the model against another dataset. As we cannot use the test dataset to validate our model, we'll need to split the training dataset using the train test split function from Sklearn model selection.

Different Classification models

A classification model attempts to draw some conclusions from observed values. Given one or more inputs, a classification model will predict the value of one or more outcomes. There are two approaches to machine learning: supervised and unsupervised. We will use a supervised learning approach here.

There are many classification algorithms available now, but it is not possible to conclude which one is superior to the other. It depends on the application and nature of the available data set. So we will now discuss different classification models.

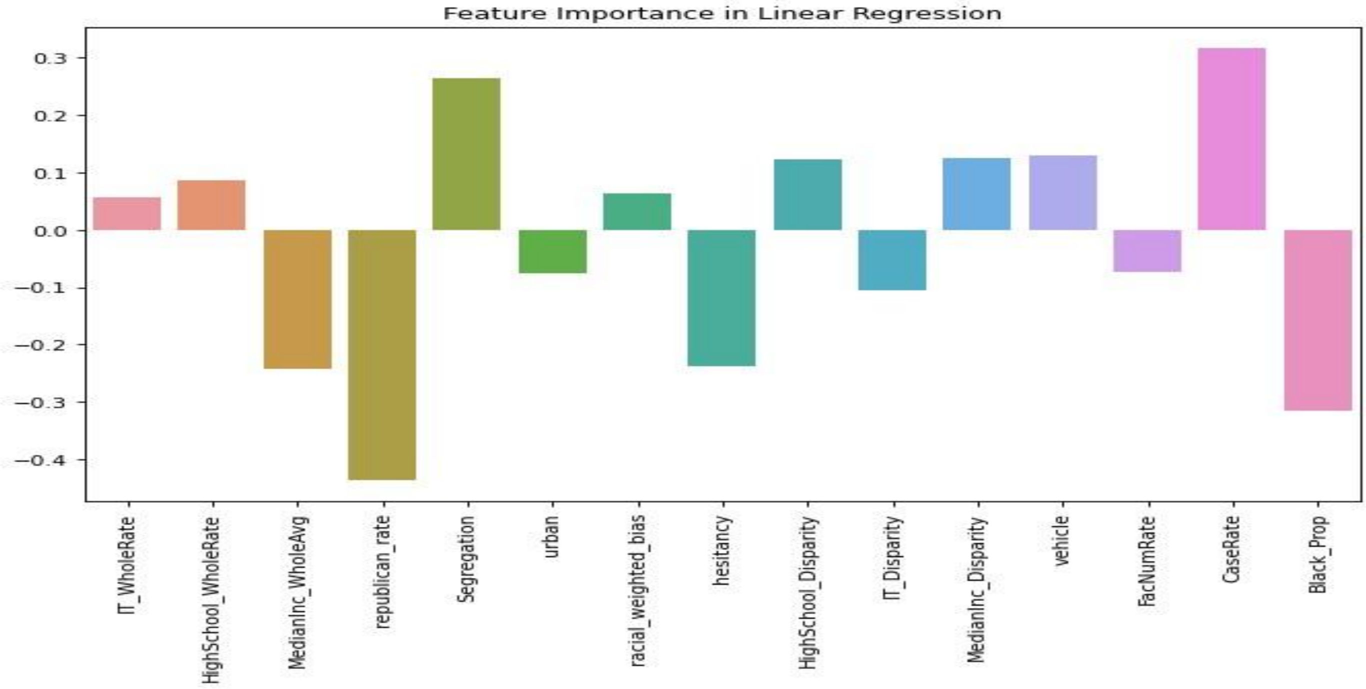
Linear Regression using Sklearn

Linear regression is an attractive model because the representation is so simple.

The representation is a linear equation that combines a specific set of input values (x), the solution is the predicted output for that set of input values (y). As such, both the input values (x) and the output value are numeric. Learning a linear regression model means estimating the importance of the coefficients used in the representation with the data that we have available.

Results: Mean Squared Error : 0.637

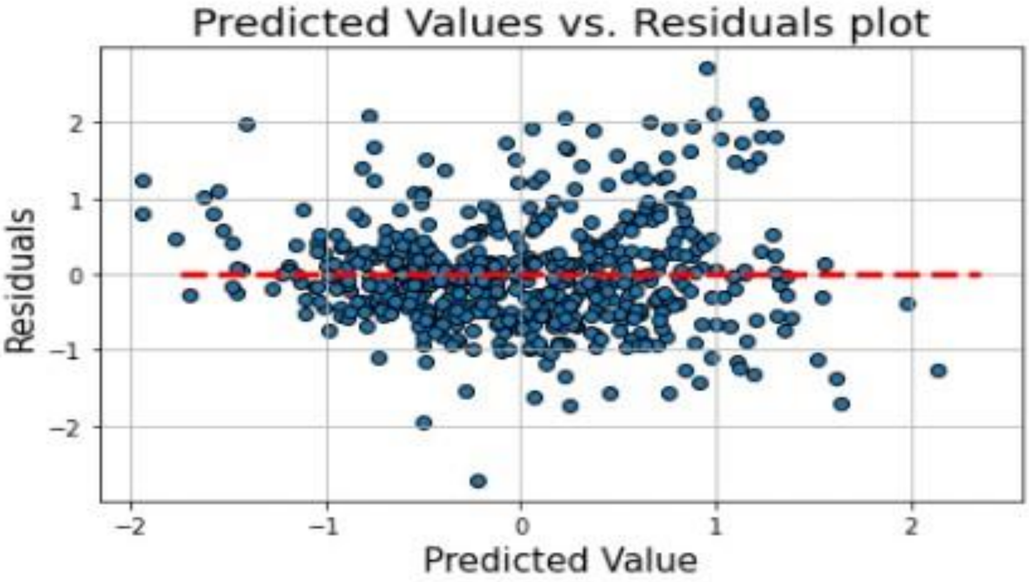
R-squared score : 0.363

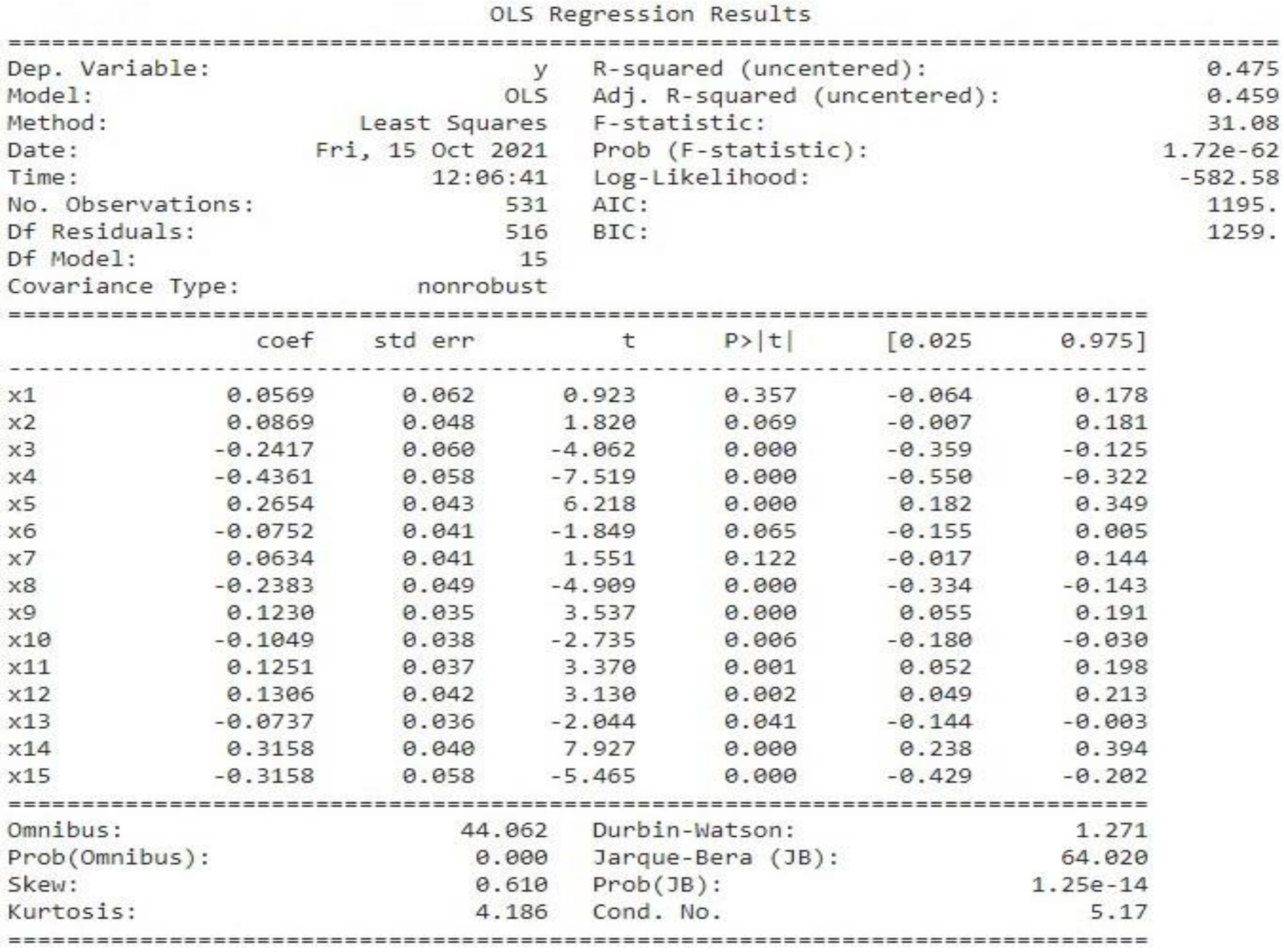


OLS Regression using statsmodels.api

The Ordinary Least Squares procedure seeks to minimize the sum of the squared residuals. This means that given a regression line, we calculate the distance from each data point to the regression line, square it, and sum all of the squared errors together. This is the quantity that ordinary least squares seek to minimize.

This approach treats the data as a matrix and uses linear algebra operations to estimate the optimal values for the coefficients. It means that all of the data must be available, and you must have enough memory to fit the data and perform matrix operations.



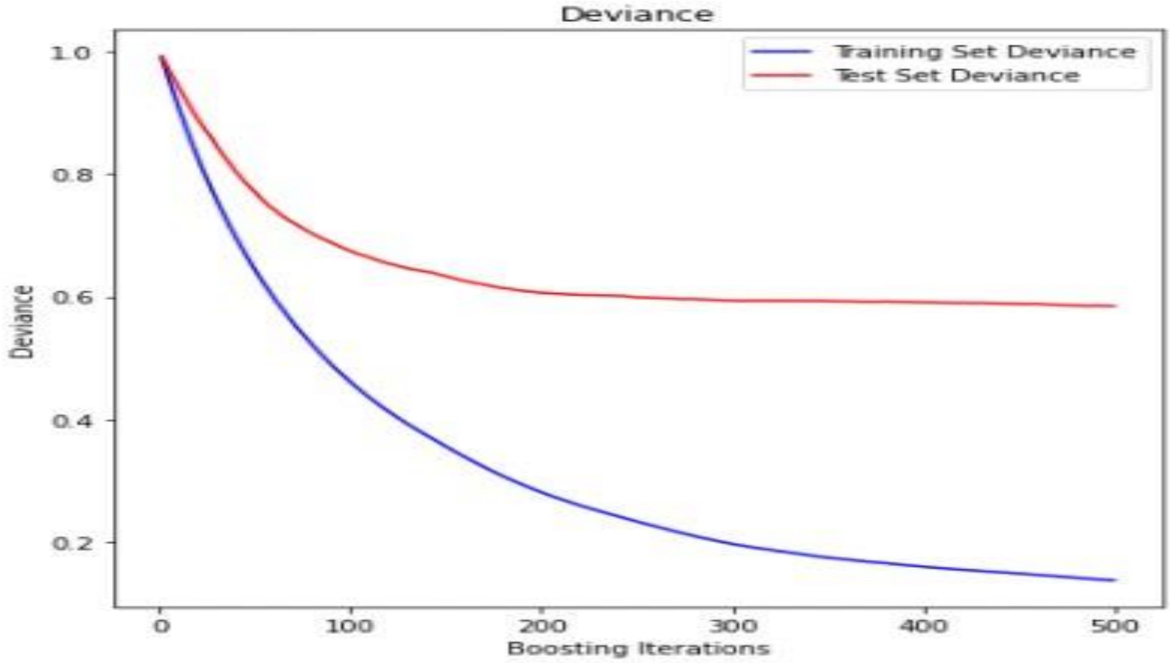


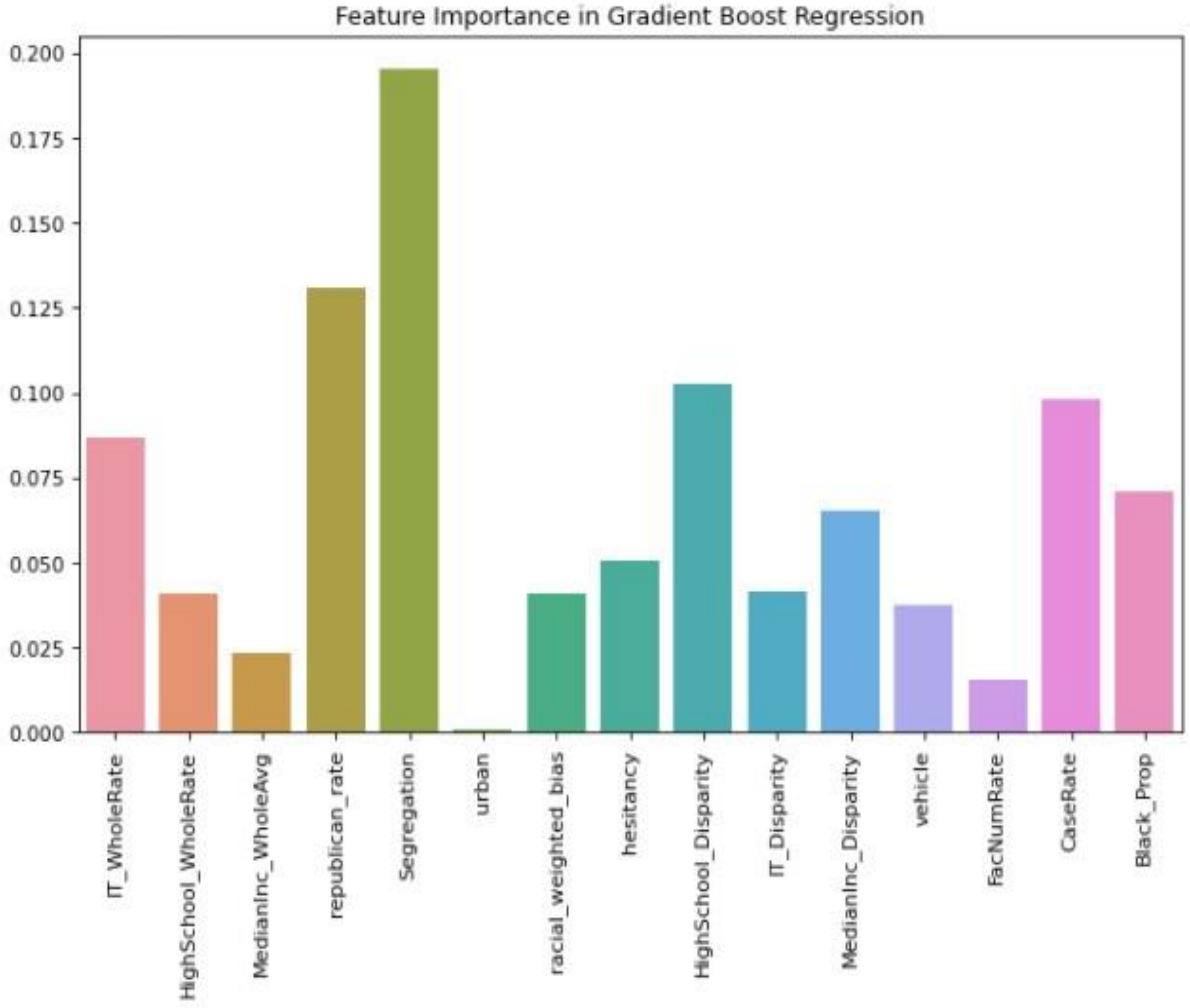
```
                                 OLS Regression Results
=======================================================================================
Dep. Variable:                      y   R-squared (uncentered):                   0.475
Model:                            OLS   Adj. R-squared (uncentered):              0.459
Method:                 Least Squares   F-statistic:                              31.08
Date:                Fri, 15 Oct 2021   Prob (F-statistic):                    1.72e-62
Time:                        12:06:41   Log-Likelihood:                         -582.58
No. Observations:                 531   AIC:                                      1195.
Df Residuals:                     516   BIC:                                      1259.
Df Model:                          15
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1             0.0569      0.062      0.923      0.357      -0.064       0.178
x2             0.0869      0.048      1.820      0.069      -0.007       0.181
x3            -0.2417      0.060     -4.062      0.000      -0.359      -0.125
x4            -0.4361      0.058     -7.519      0.000      -0.550      -0.322
x5             0.2654      0.043      6.218      0.000       0.182       0.349
x6            -0.0752      0.041     -1.849      0.065      -0.155       0.005
x7             0.0634      0.041      1.551      0.122      -0.017       0.144
x8            -0.2383      0.049     -4.909      0.000      -0.334      -0.143
x9             0.1230      0.035      3.537      0.000       0.055       0.191
x10           -0.1049      0.038     -2.735      0.006      -0.180      -0.030
x11            0.1251      0.037      3.370      0.001       0.052       0.198
x12            0.1306      0.042      3.130      0.002       0.049       0.213
x13           -0.0737      0.036     -2.044      0.041      -0.144      -0.003
x14            0.3158      0.040      7.927      0.000       0.238       0.394
x15           -0.3158      0.058     -5.465      0.000      -0.429      -0.202
==============================================================================
Omnibus:                       44.062   Durbin-Watson:                   1.271
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               64.020
Skew:                           0.610   Prob(JB):                     1.25e-14
Kurtosis:                       4.186   Cond. No.                         5.17
==============================================================================
```

Gradient Boost Regression

GB builds an additive model in a forward stage-wise fashion; it optimizes arbitrary differentiable loss functions. In each stage, a regression tree is fit on the negative gradient of the given loss function.

Results: The mean squared error (MSE) on test set: 0.5859
The R- Squared score on the test set: 0.4141

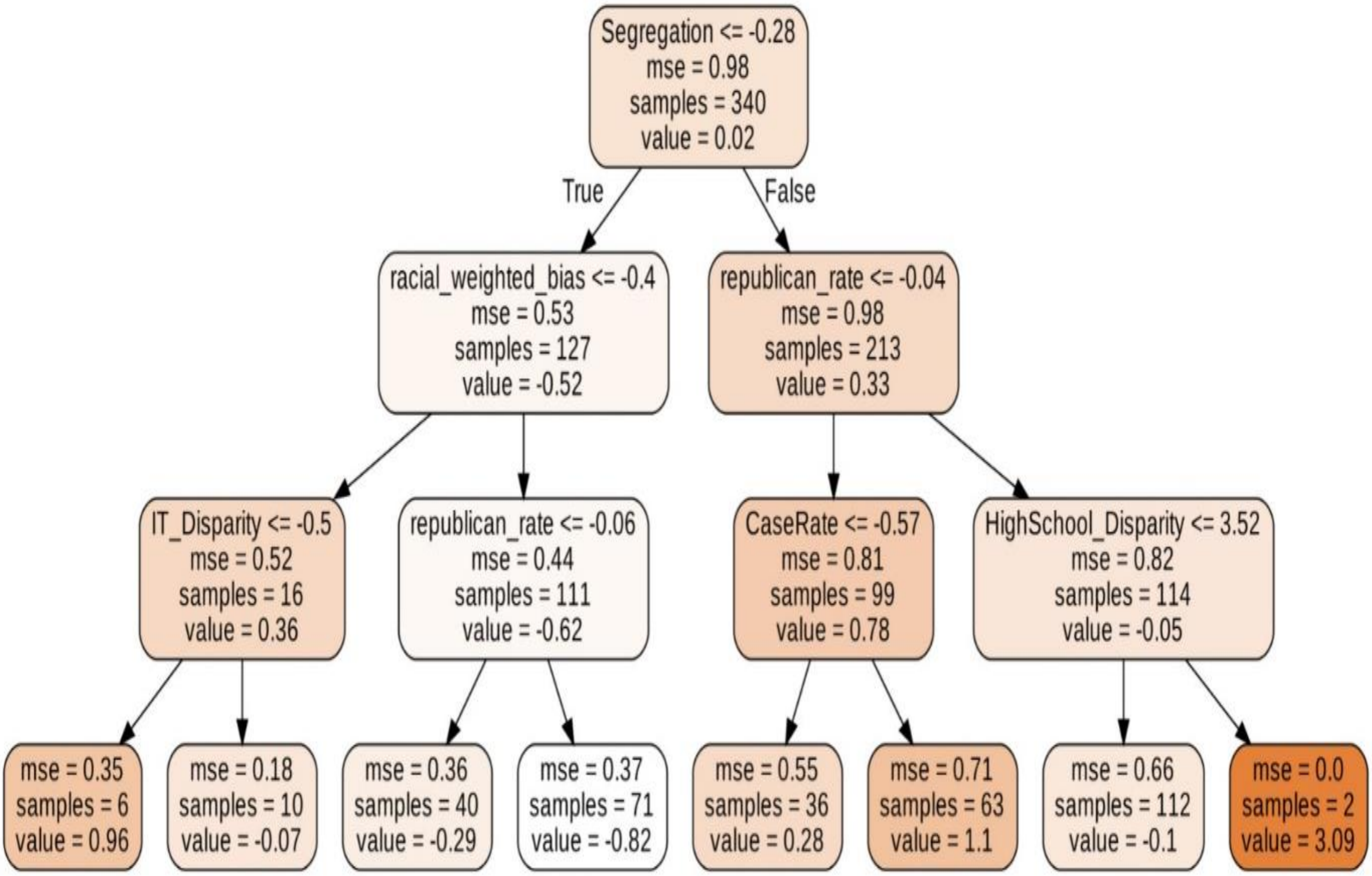

Deviance
Training Set Deviance
Test Set Deviance
Deviance
1.0
0.8
0.6
0.4
0.2
0
100
200
300
400
500
Boosting Iterations


Feature Importance in Gradient Boost Regression
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
IT_WholeRate
HighSchool_WholeRate
MedianInc_WholeAvg
republican_rate
Segregation
urban
racial_weighted_bias
hesitancy
HighSchool_Disparity
IT_Disparity
MedianInc_Disparity
vehicle
FacNumRate
CaseRate
Black_Prop

Random Forest Regression

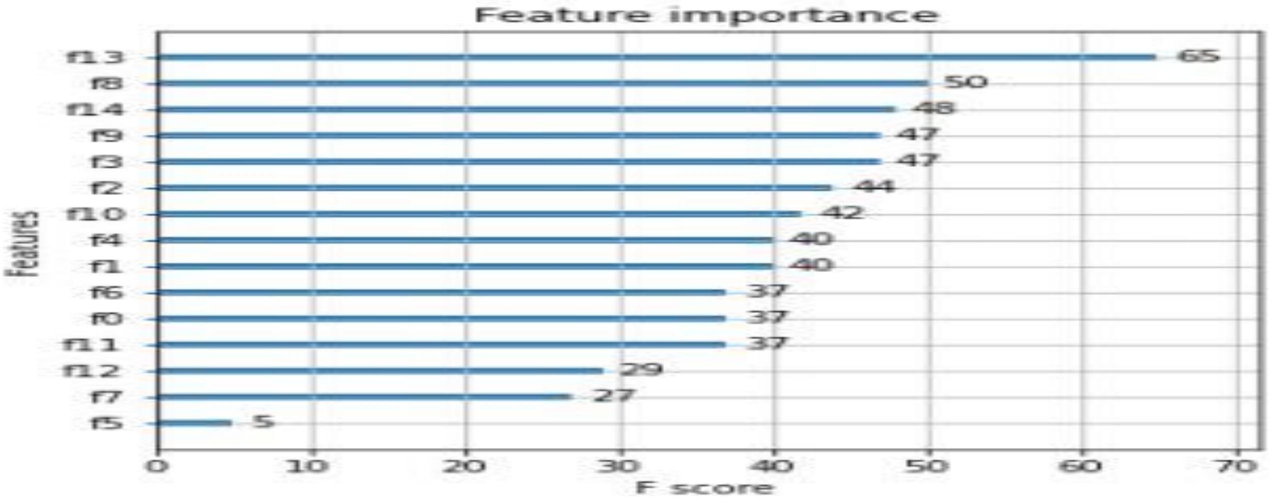
Random Forest Regression is a supervised learning algorithm that uses the **ensemble learning** method for regression. The ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.

Results:The mean squared error (MSE) on test set: 0.6614
The R-squared error on test set: 0.3386



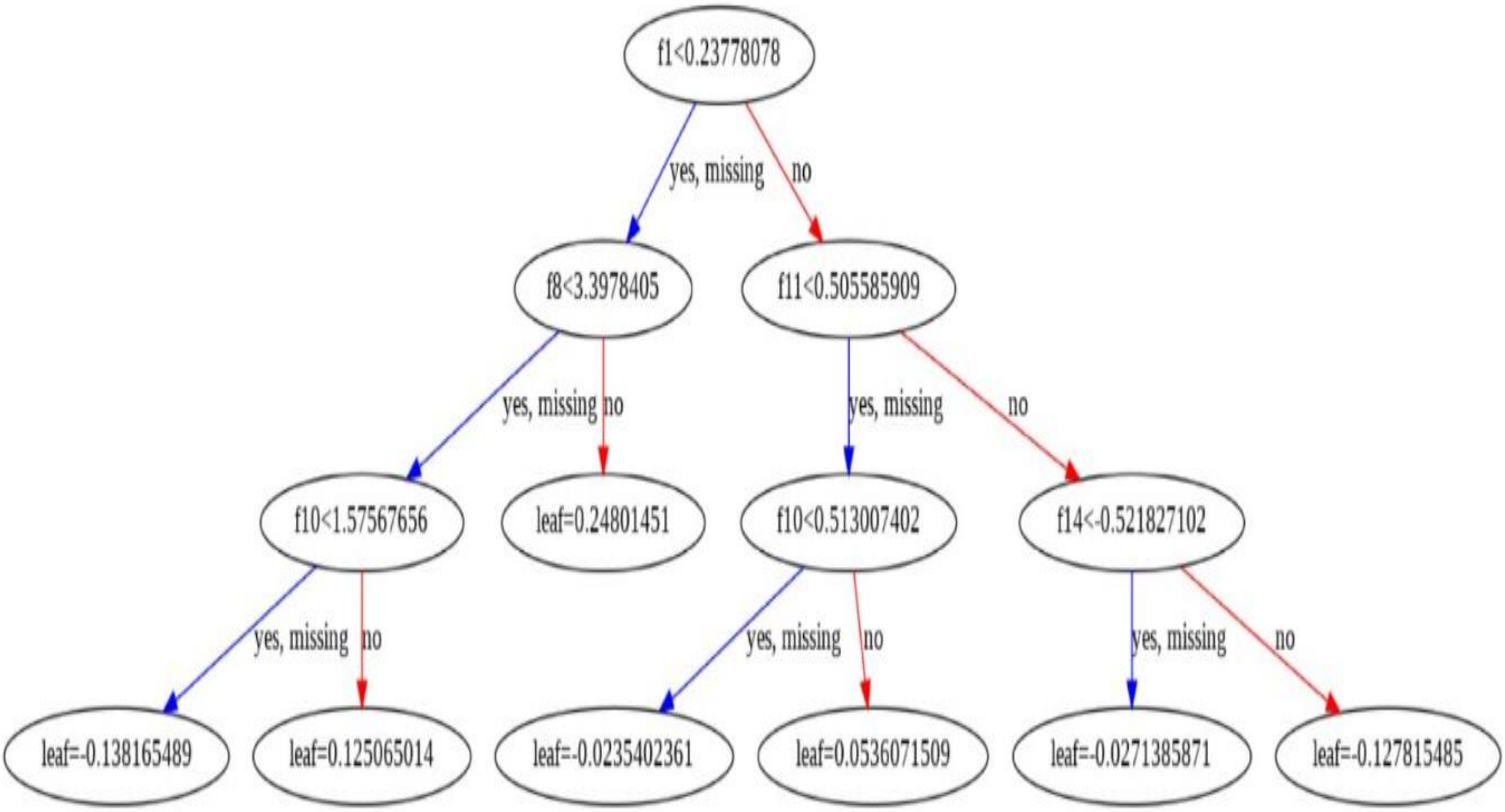
XGBoost Regression

Extreme Gradient Boosting (XGBoost) is an open-source library that provides an efficient and effective implementation of the gradient boosting algorithm.



XGBoost is an efficient implementation of gradient boosting that can be used for predictive regression modeling.

Results: The mean squared error (MSE) on test set: 0.5979
The R-squared error on test set: 0.4021



Support Vector Regression

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by ending the hyper-plane that differentiates the two classes very well.
SVMs can be used on nonlinear data also using something called a Kernel trick. Basically it projects our data into a higher dimensional space such that our data can be separated by a hyperplane in this space.

Results:- The mean squared error (MSE) on test set: 0.6373
The R-squared score on the test set: 0.3627

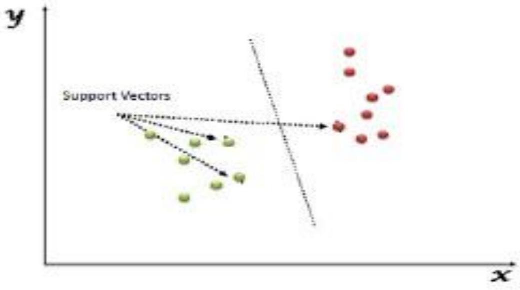


Figure 7: SVM classifier

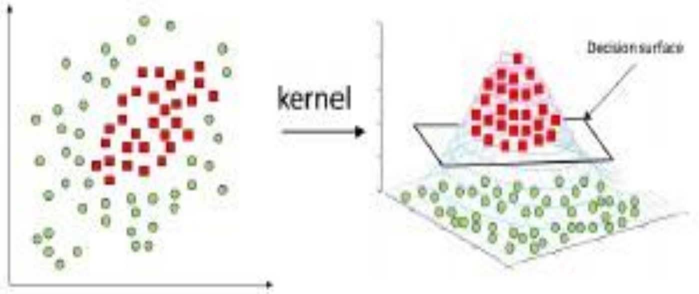
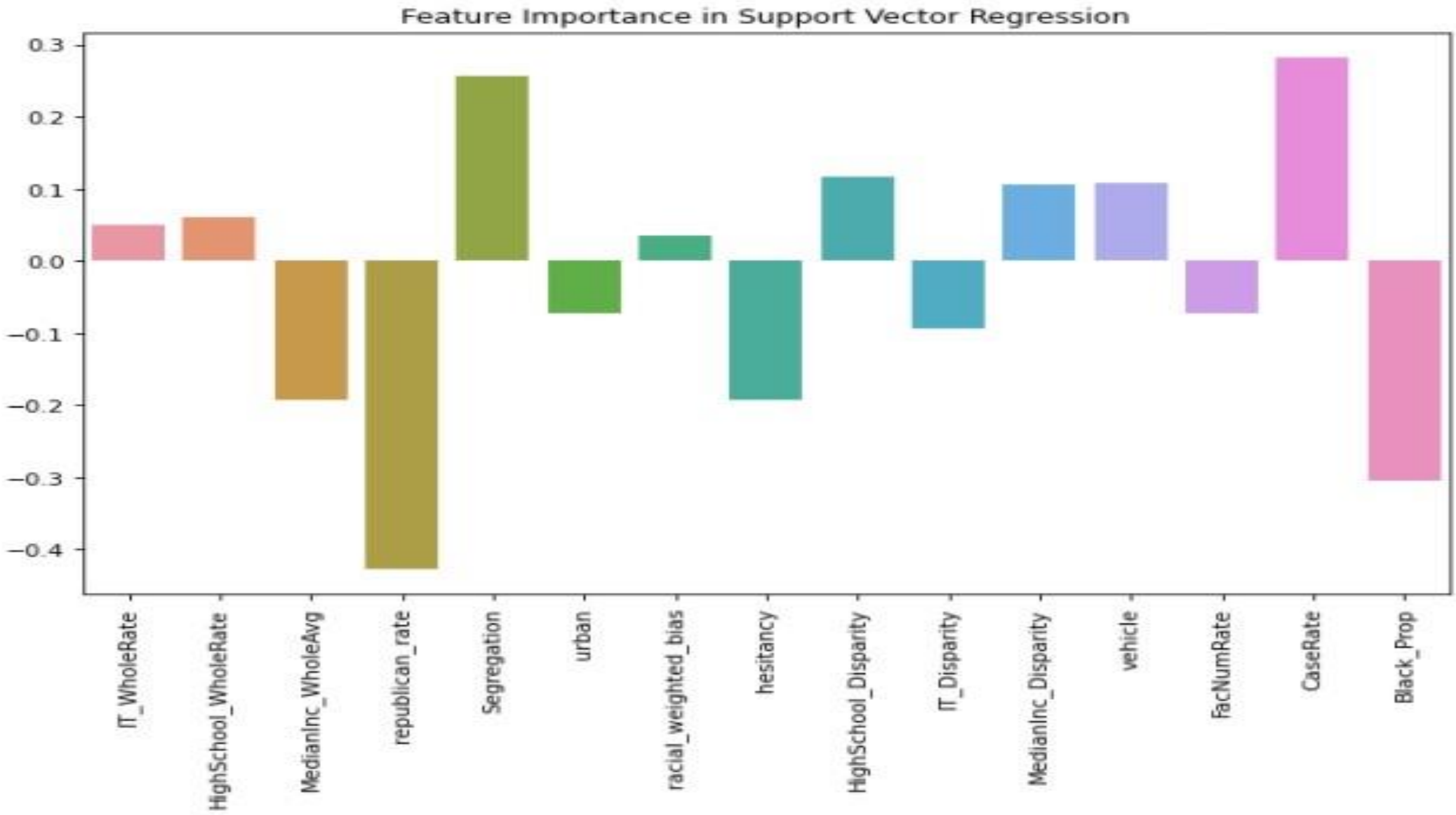


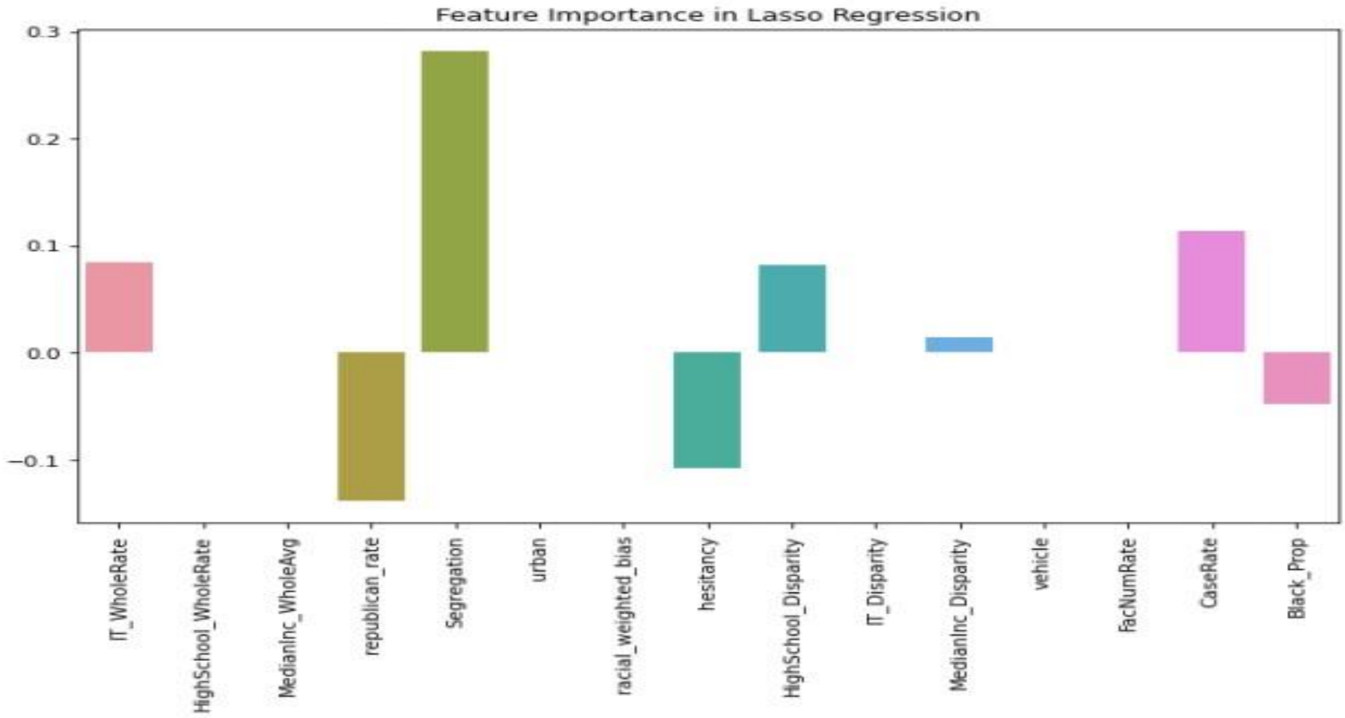
Figure 8: Kernel trick using Gaussian kernel



LASSO Regression

Lasso regression is a type of **linear regression** that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean. The lasso procedure encourages simple, sparse models (i.e., models with fewer parameters). This particular type of regression is well-suited for models showing high levels of multicollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination.

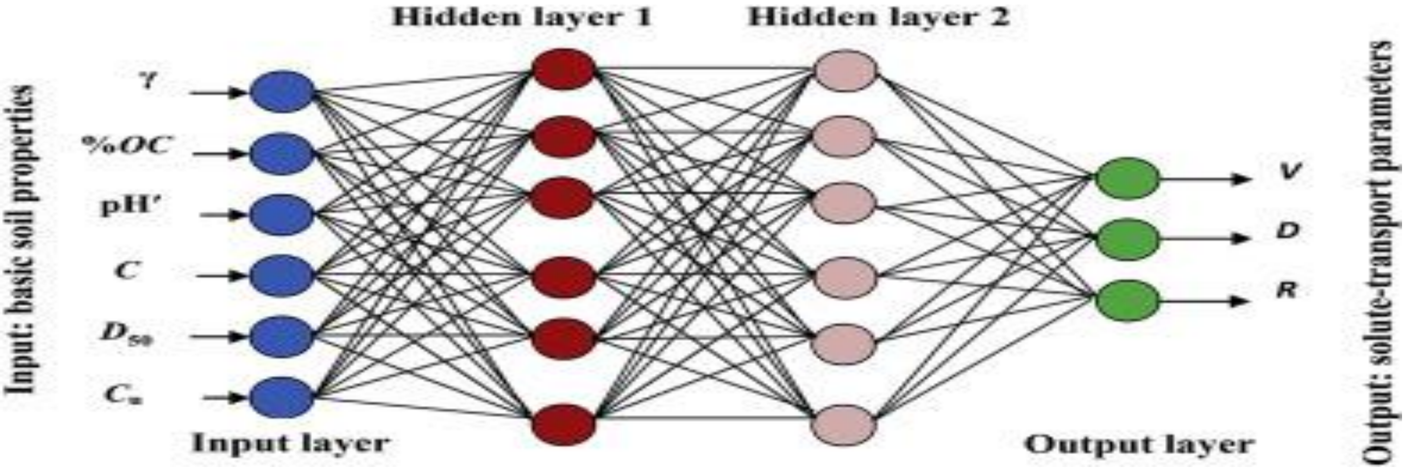
Results:- The mean squared error (MSE) on test set: 0.7107
The R-squared score on the test set: 0.2893



Neural Networks

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data by mimicking how the human brain operates.

Results:- The mean squared error (MSE) on test set: 0.5479
The R-squared score on the test set: 0.4521



Conclusion:-

Resources:-

https://machinelearningmastery.com/linear-regression-for-machine-learning/

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html

https://levelup.gitconnected.com/random-forest-regression-209c0f354c84

https://machinelearningmastery.com/xgboost-for-regression/

https://towardsdatascience.com/unlocking-the-true-power-of-support-vector-regression-.