Milestone: - Final Project Report

Uncovering Crime Patterns in Communities: A Data-Driven Approach

Group 87

Student 1: Vishesh Gupta
Student 2: Aryan Fernandes

gupta.vishe@northeastern.edu
fernandes.ar@northeastern.edu

Percentage of Effort Contributed by Student 1: 50%

Percentage of Effort Contributed by Student 2: 50%

Signature of Student 1: Vishesh Gupta

Signature of Student 2: Aryan Fernandes

Submission Date: **April 21st 2023**

Table of Contents

Problem Setting: Urban communities are facing an alarming rise in crime rates, which poses a significant threat to the safety and well-being of residents. The increasing crime not only affects the physical security of individuals but also leads to decreased property values, slowed economic development, and a negative impact on the overall quality of life in a community. Despite these negative impacts, many urban communities struggle to effectively address crime and make their neighborhoods safer due to lack of relevant data and proper analysis. So, each records of the data involve demographic information, economic information, and crime statistics to identify patterns and trends that may be associated with higher crime rates.

Problem Definition: The problem of increasing crime rates in urban communities is a complex issue that requires a multifaceted approach. To effectively address this problem, a thorough analysis is needed to identify patterns and trends that may be associated with higher crime rates. This information can then be used to develop targeted strategies to reduce crime and improve safety in these communities. Additionally, this project aims to provide insights into the root causes of crime in urban communities, which can inform policies and programs to prevent crime and promote community safety.

Key objectives of the project:

- Collect and clean crime data from various sources, including local law enforcement agencies and government databases.
- Analyze crime data at the neighborhood level to identify patterns and trends in crime incidents.
- Investigate how various factors, such as population density, socio-economic status, and the presence of amenities or infrastructure, are related to crime rates in different communities.
- Develop a model to classify crimes or predict crime rates in neighborhoods based on the identified factors.
- Use the findings from the analysis to develop recommendations for reducing crime and improving community safety.

Key Deliverables:

- A detailed analysis of crime patterns and trends in the target communities.

Data Sources:

Website link: https://archive.ics.uci.edu/ml/datasets/communities+and+crime

Data agency: University of California, Irvine (UCI)

Dataset owner: Center for Machine Learning and Intelligent Systems (CMLIS)

Type of data: Community demographic, economic, and crime

statistics Number of records: 2100

Number of attributes: 128

Additional information: The data was collected by the FBI's Uniform Crime Reporting Program and the U.S. Census Bureau's. It includes information on various community characteristics such as population, race, education level, and income, as well as crime statistics such as number of reported violent crimes and property crimes. We will be utilizing dimension reduction techniques to enhance the effectiveness of the dataset.

Sample Dataset Representation:

8	?	?	Lakewood	1	0.19	0.33	0.02	0.9	0.12	0.17	0.34	0.47	0.29	0.32	0.2	1	0.37	0.72	0.34
53	?	?	Tukwilacit	1	0	0.16	0.12	0.74	0.45	0.07	0.26	0.59	0.35	0.27	0.02	1	0.31	0.72	0.11
24	?	?	Aberdeent	1	0	0.42	0.49	0.56	0.17	0.04	0.39	0.47	0.28	0.32	0	0	0.3	0.58	0.19
34	5	81440	Willingbor	1	0.04	0.77	1	0.08	0.12	0.1	0.51	0.5	0.34	0.21	0.06	1	0.58	0.89	0.21
42	95	6096	Bethlehem	1	0.01	0.55	0.02	0.95	0.09	0.05	0.38	0.38	0.23	0.36	0.02	0.9	0.5	0.72	0.16
6	?	?	SouthPasa	1	0.02	0.28	0.06	0.54	1	0.25	0.31	0.48	0.27	0.37	0.04	1	0.52	0.68	0.2
44	7	41500	LincolnTow	1	0.01	0.39	0	0.98	0.06	0.02	0.3	0.37	0.23	0.6	0.02	0.81	0.42	0.5	0.23
6	?	?	Selmacity	1	0.01	0.74	0.03	0.46	0.2	1	0.52	0.55	0.36	0.35	0	0	0.16	0.44	1
21	?	?	Hendersor	1	0.03	0.34	0.2	0.84	0.02	0	0.38	0.45	0.28	0.48	0.04	1	0.17	0.47	0.36
29	?	?	Claytoncit	1	0.01	0.4	0.06	0.87	0.3	0.03	0.9	0.82	0.8	0.39	0.02	1	0.54	0.59	0.22
6	?	?	DalyCitycit	1	0.13	0.71	0.15	0.07	1	0.41	0.4	0.52	0.35	0.33	0.15	1	0.49	0.71	0.16
36	?	?	RockvilleC	1	0.02	0.46	0.08	0.91	0.07	0.1	0.34	0.36	0.22	0.57	0.04	1	0.72	0.53	0.23
25	21	44105	Needhamt	1	0.03	0.47	0.01	0.96	0.13	0.02	0.29	0.32	0.2	0.52	0.04	1	0.8	0.55	0.18
55	87	30075	GrandChut	1	0.01	0.44	0	0.98	0.04	0.01	0.35	0.53	0.32	0.23	0.02	0.77	0.46	0.77	0.41
6	?	?	DanaPoint	1	0.04	0.36	0.01	0.85	0.14	0.26	0.32	0.46	0.3	0.31	0.05	1	0.71	0.67	0.42
19	187	91370	FortDodge	1	0.03	0.34	0.06	0.93	0.03	0.03	0.39	0.41	0.28	0.58	0	0	0.18	0.42	0.81
36	1	1000	Albanycity	1	0.15	0.31	0.4	0.63	0.14	0.06	0.58	0.72	0.65	0.47	0.16	1	0.22	0.52	0.1
34	27	17650	Denvilleto	1	0.01	0.53	0.01	0.94	0.2	0.03	0.34	0.39	0.27	0.36	0.02	0.76	0.79	0.77	0.13
18	?	?	Valparaiso	1	0.02	0.47	0.01	0.97	0.07	0.02	0.7	0.67	0.63	0.37	0	0	0.33	0.56	0.28
42	129	66376	Rostravert	1	0	0.41	0.05	0.96	0.01	0.01	0.37	0.37	0.24	0.55	0.01	0.58	0.23	0.34	0.33
6	?	?	Modestoci	1	0.25	0.54	0.05	0.71	0.48	0.3	0.42	0.48	0.28	0.32	0.26	1	0.33	0.55	0.37
12	31	?	Jacksonvill	1	1	0.42	0.47	0.59	0.12	0.05	0.41	0.53	0.34	0.33	1	0.99	0.28	0.62	0.16
41	?	?	KlamathFa	1	0.01	0.34	0.02	0.87	0.07	0.11	0.49	0.56	0.43	0.47	0	0	0.13	0.4	0.26
19	193	93926	SiouxCityc	1	0.11	0.43	0.04	0.89	0.09	0.06	0.45	0.48	0.31	0.46	0.13	1	0.22	0.52	0.44
6	?	?	Delanocity	1	0.02	0.96	0.05	0	1	1	0.54	0.58	0.39	0.33	0	0	0.16	0.61	0.41

Fig 1. Dataset

The dataset will include the most important attributes as shown in the sample below, as well as a few other attributes that will be countered later on as per analysis:

Attributes	Attributes Type	Illustration
Community_Name	Str/Text	Community name for information only
State_Name/Code	Str/Text	Code for State_Name
Race_Asian_Pct	Numeral-Decimal	Percentage of population that is of asian heritage
Race_White_Pct	Numeral-Decimal	Percentage of population that is caucasian
Auto_Theft_Per_Pop	Numeral-Decimal	Theft cases per 100K population
House_Hold_Size	Numeral-Decimal	Mean people per household
Violent_Crimes_Per_Pop	Numeral-Decimal	Total number of violent crimes per 100K population GOAL attribute (to be predicted)

Fig 2. Attributes and dtypes

Data Exploration

Data exploration is the first and critical step in the data analysis process of our dataset "Communities_Crime.csv." By using various methods like quantitative and qualitative data exploration, analysts can identify hidden patterns, relationships, and biases in the raw data. Visualization techniques, such as scatter plots and Boxplots, can help them to interpret the data quickly and efficiently. Data exploration is not a one-time process, and analysts may need to revisit their assumptions and hypotheses as they gain a deeper understanding of the data. Through data exploration, organizations can gain a competitive advantage and make data-driven decisions that lead to better outcomes. It's clear that data exploration is an iterative and essential process that helps us extract valuable insights from data and generate new hypotheses for further analysis.

Data Visualization and Insights

To better understand the complex relationships between the variables in our dataset, we have decided to create a Heat-Map visualization. By analyzing the correlations between our variables, we hope to uncover hidden patterns and gain deeper insights into the underlying structure of our data. To accomplish this, we utilized our original DataFrame (community_df) and performed the corr() function, which allowed us to calculate the correlation coefficients between each pair of variables.

In order to create a powerful and visually engaging Heat-Map, we turned to the plotly Python Data Visualization Library, which is built on top of Matplotlib. Using plotly, we were able to customize our Heat-Map and generate a stunning visual representation of our data. Each square in the Heat-Map represents the correlation coefficient between two variables, with a color gradient that reflects the strength of the correlation.

Through our Heat-Map visualization, we were able to identify key relationships between our variables and gain deeper insights into the underlying structure of our data. This powerful tool allowed us to draw important connections between different factors in our community and paved the way for more effective decision-making and problem-solving in the future.

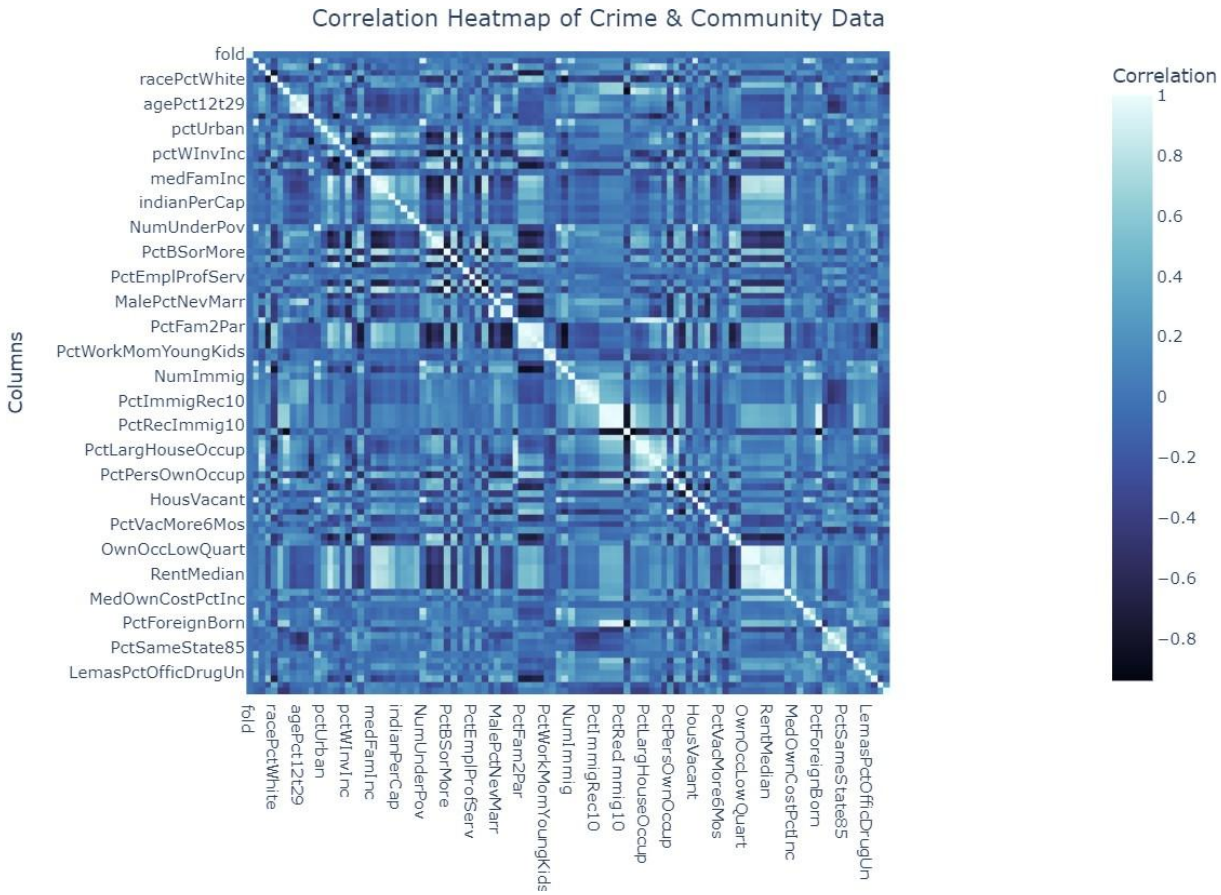


Fig -3 Heat_Map

Correlation values range from -1 to 1, with values close to 1 being highly correlated and values close to -1 being negatively correlated. [‘Household-size’, ‘pct12-21’, ‘pct age 65’, ‘pct whiterace’, ‘pct blackrace’, ‘urban population’ - median income, ‘white-percap income’, ‘hisp-income percap income’, ‘pct 9thgrade’, ‘pct poverty’] are some of the columns with a high degree of correlation.

Standardization

To prepare our dataset for analysis of principal components, we decided to apply standardization using the StandardScaler() function to our communities_df DataFrame. This transformation helps to put our data in a proper format and ensure that all values are in the same range, making it easier to analyze and visualize. By standardizing our data, we were able to remove any potential biases or inconsistencies that may have been present in the raw data, and get a clearer understanding of the patterns and relationships within our dataset. This crucial step paved the way for smoother and more effective analysis and helped us to draw more accurate conclusions about our community data.

```
## Performing scaling on our input dataset

from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_scale = sc.fit_transform(prec_df)

#standardization data after performing standardization
standardized_final_data=pd.DataFrame(data=X_scale, columns = prec_df.columns)
standardized_final_data
```

	population	householdsize	racepctblack	racePctWhite	racePctAsian	racePctHisp	agePct12t21	agePct12t29	agePct16t24	agePct65up	...
0	1.043612	-0.814997	-0.630002	0.599578	-0.161288	0.111765	-0.542791	-0.166286	-0.277923	-0.575886	...
1	-0.453937	-1.853636	-0.235335	-0.056219	1.418982	-0.318466	-1.058398	0.669791	0.082518	-0.854996	...
2	-0.453937	-0.265129	1.224931	-0.793990	0.078147	-0.447535	-0.220536	-0.166286	-0.337996	-0.575886	...
3	-0.138663	1.873246	3.237730	-2.761379	-0.161288	-0.189397	0.552875	0.042733	0.022445	-1.189929	...
4	-0.375118	0.529125	-0.630002	0.804514	-0.304949	-0.404512	-0.284987	-0.793343	-0.638364	-0.352597	...
...	...	...	...	...	...	...	...	...	...	...	...
1989	-0.375118	-0.387322	-0.314269	0.476616	-0.161288	0.068742	0.037268	0.112406	0.082518	-0.687530	...
1990	-0.059845	3.034078	1.106531	-1.941633	3.238686	0.757111	1.712992	2.550963	2.365311	-1.580683	...
1991	0.807157	-0.570611	0.277731	-0.261155	-0.544384	0.455950	-0.478340	0.042733	-0.157776	0.652200	...
1992	0.176610	0.284739	-0.472135	0.476616	0.317581	-0.189397	1.004032	1.714887	1.764576	-0.073487	...
1993	1.122430	1.934343	-0.156402	-1.203862	0.413355	2.693151	0.488424	0.878810	0.382886	-1.413217	...

Fig -4 Standardized Data

Dimension Reduction

Principal Component Analysis

To further analyze our standardized dataset, we decided to apply principal component analysis (PCA) to extract the most important components that capture the highest variance in our data. By using PCA, we were able to reduce the dimensionality of our data and focus on the most relevant variables for our analysis. After applying PCA, we were able to compute 15 principal components out of 104 attributes in our dataset, which captured 85 percent of the total variance.

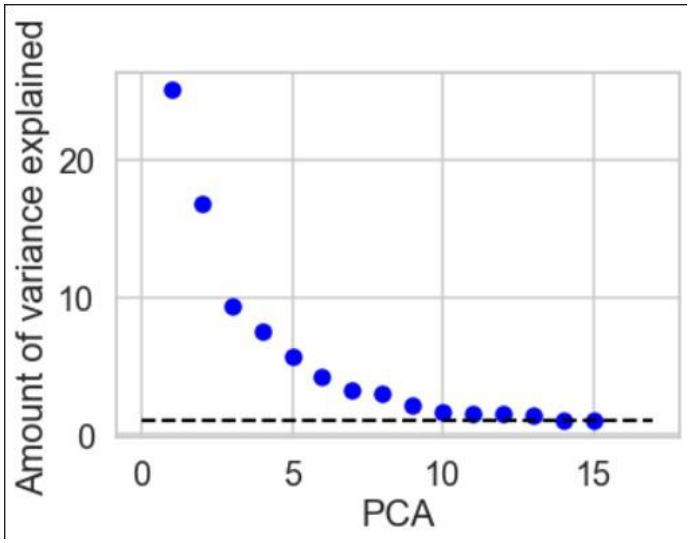


Fig -5 PCA Components

This allowed us to focus on the most significant factors in our data and ignore any extraneous variables that may have been contributing noise or irrelevant information. To better understand the relationship between these 15 principal components, we decided to use the pairplot function, which allowed us to visualize the relationships between variables in our data. By plotting the pairwise relationships between our principal components [PCA_1, PCA_2, PCA_3, PCA_4, PCA_5, PCA_6, PCA_7, PCA_8, PCA_9, PCA_10, PCA_11, PCA_12, PCA_13, PCA_14, PCA_15], we were able to gain deeper insights into the underlying structure of our dataset and identify any significant patterns or trends.

Overall, the combination of PCA and pairplot proved to be a powerful tool for analyzing and visualizing our data, allowing us to focus on the most important variables and uncover key insights about our community.

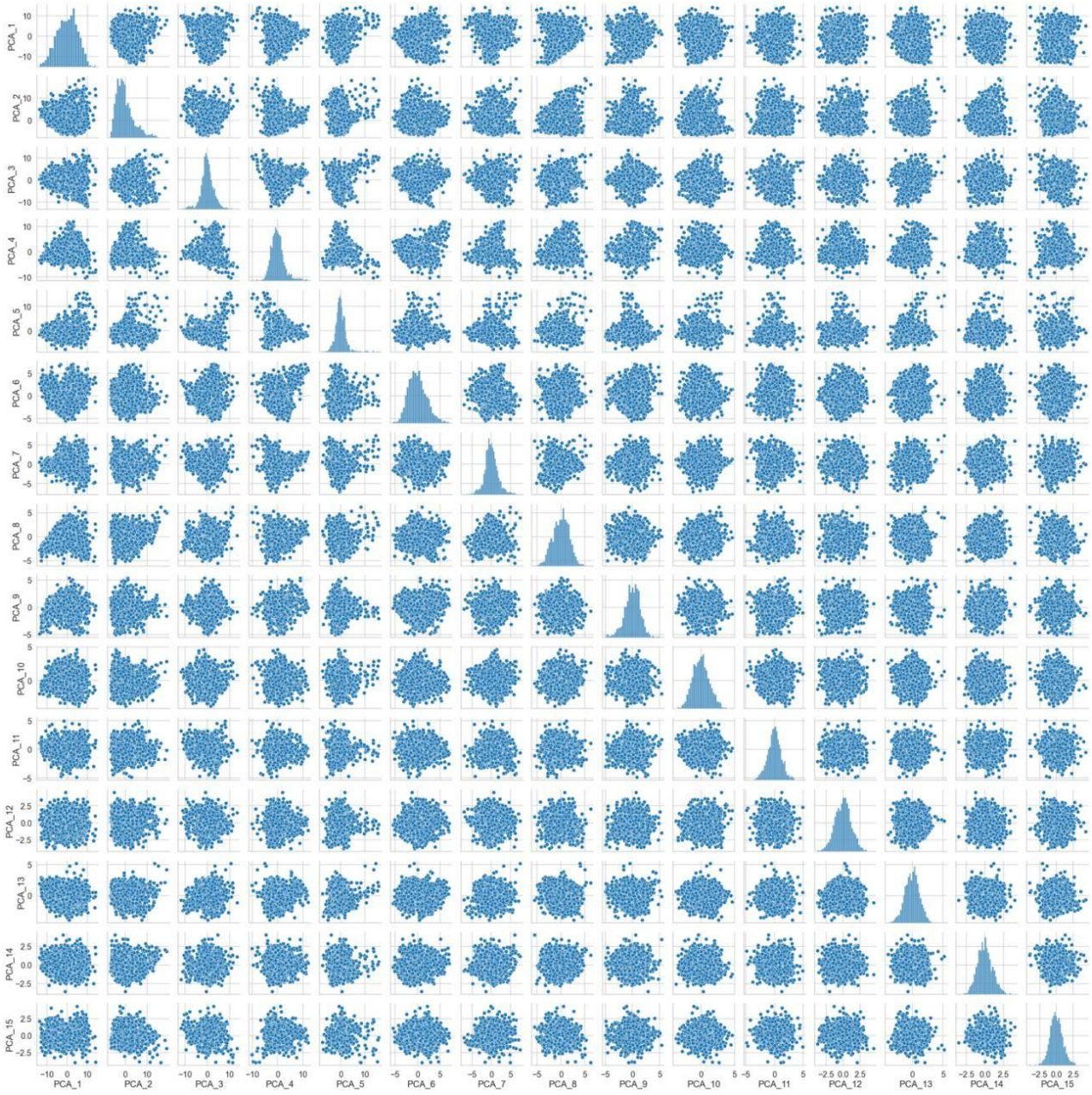


Fig -6 Pair Plot

Distribution of Data using Box Plot

The next step is to examine the distribution of the predictors after performing principal component analysis (PCA) on a set of relevant variables. A box plot has been added to visually display the distribution and any outliers in order to accomplish this. The box plot is a quick and easy way to identify any unusual data points in the distribution, such as extreme values or skewness. We can gain a better understanding of the characteristics of the attributes and determine how much distribution each attribute carries by analyzing the box plot. Overall, the box plot is an effective tool for exploratory data analysis after PCA.

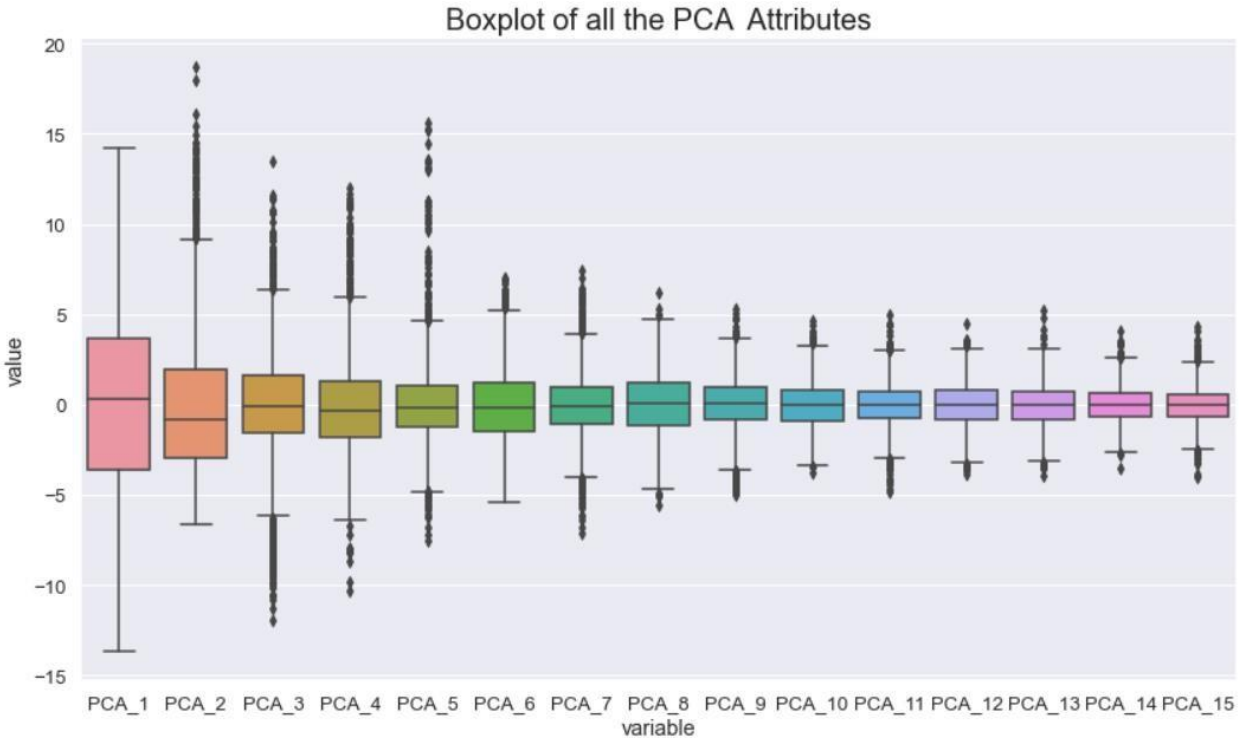


Fig -7 Box Plot

Folium Map plot

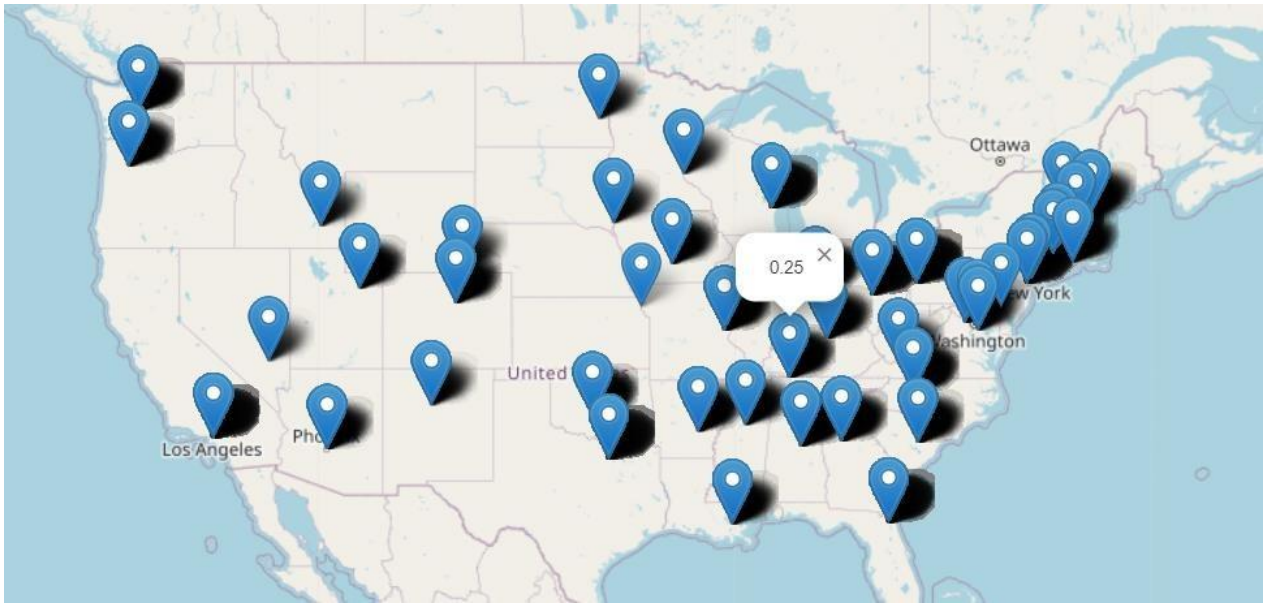


Fig -8 Folium Map

We used the Python library Folium to create a map that displays latitude and longitude data for communities in different states. We also included a hover feature that shows the level of violent crime prediction for each state on the map across United States. This allows for easy visualization of the location and potential safety risks in different areas. Overall, the map provides a useful tool for exploring and understanding geographic data related to violent crime.

Model Exploration

Model exploration is the process of understanding how a trained machine learning model makes predictions and how it represents the relationships between the input features and the target variable

In the context of model exploration, we are performing supervised learning which can be useful because it provides a clear target variable to predict and evaluate the performance of the model. In this project of Communities and Crime, we have a target variable called violent_crime_per_population, and our goal is to perform supervised learning on this variable using regression techniques. Regression is a type of supervised learning that aims to predict a continuous numerical value, such as the crime rate in this case.

To explore and understand the relationships between the input features and the target variable, we have used techniques such as visualization, Correlation, model interpretation, and analysis. For example, we have created pair plots and heat maps to visualize the relationships between different features and the target variable and used Dimension Reduction Techniques such as Principal component analysis to understand which features are capturing the highest number of variance which is really most important for the regression model.

To perform model exploration and selection on the Communities and Crime dataset, we first checked the values in our PCA dataframe after standardizing it. We then added the target variable, violent_crime_rate_per_population, to the PCA dataframe to perform model exploration. To do this, we removed the same variable from our standardized dataset to obtain separate input and output sets. For model performance, we split our data into training and testing sets in a ratio of 80:20 for both the input predictor variables and the output target variable.

Here, X represents the standardized input predictor variables (after PCA), and y represents the target variable (violent_crime_per_population). We split the data into a training set (80%), a testing set (20%) using the train_test_split() function from scikit-learn

We can then use the training set to fit different regression models. Finally, we can evaluate the performance of the selected model on the testing set to estimate its generalization performance

Model Selection

In terms of Machine Learning: Model selection is an important step in machine learning and involves choosing the best model or algorithm to use for a given problem. In the case of the Communities and Crime dataset, since we are performing regression, we need to choose a regression model that can accurately predict crime rates in different communities based on a set of input features.

As an illustration, we could aim to identify the suitable hyperparameters for a specific machine learning algorithm. These hyperparameters are the parameters of the learning method that must be set prior to the model fitting.

There are many different regression models to choose from, including linear regression, decision trees, random forests, support vector regression, and neural networks, among others. The best model for the task depends on a number of factors, including the size and complexity of the dataset, the number of input features, the quality of the data, and the desired level of accuracy.

In this milestone, our primary goal is to evaluate the performance of different regression models on our training, validation and testing data. We will achieve this by applying six different regression models to the standardized input data and comparing their performance metrics. The models we will be using include linear regression, Lasso regression, Ridge regression, ElasticNet, Decision Tree regression, and Random Forest regression.

We will first fit each model to the training data, using the training set split that was created in the previous step. We will then use the trained models to make predictions on the testing set and evaluate their performance using several metrics such as mean squared error (MSE), root mean squared error (RMSE), and coefficient of determination (R^2).

The goal of this evaluation process is to identify which regression model performs the best on our dataset and select it as our final model. By comparing the performance of different models, we can gain insights into the strengths and weaknesses of each model and select the one that provides the best balance of accuracy and interpretability

Step1) The first regression model we have used in this project is linear regression. Linear regression is a simple yet powerful technique that is widely used in machine learning for predicting numerical values, such as crime rates in different communities.

It is important because it is simple, interpretable, and computationally efficient, serving as a baseline for comparing more complex models. It assumes a linear relationship between input features and the target variable.

Step2) The second regression model we have used in this project is Lasso regression. Lasso regression is a type of linear regression that uses regularization to prevent overfitting and improve the generalization performance of the model. Furthermore, Lasso regression can be used to perform feature selection and reduce the dimensionality of the input data, which can improve the computational efficiency of the model and reduce the risk of overfitting.

Step-3) The third regression model we have used in this project is Ridge regression. Ridge regression is a type of linear regression that also uses regularization to prevent overfitting and improve the generalization performance of the model. Furthermore, Ridge regression can also be used to perform feature selection and reduce the dimensionality of the input data, similar to Lasso regression.

Ridge regression is important to use because it can help us reduce the impact of input features that are not relevant for predicting the target variable. This is done by penalizing the input coefficients that have large magnitudes, which can lead to a more robust and accurate model.
In summary, Ridge regression is an important model selection technique that can help us reduce the impact of irrelevant input features and prevent overfitting.

Step-4) The fourth regression model used in this project is Random Forest regression. Random Forest is an ensemble learning method that combines multiple Decision Trees to improve the prediction accuracy and reduce overfitting. Random Forest regression is important here to use because it can capture complex nonlinear relationships and interactions between input features, similar to Decision Trees. However, by combining multiple trees, it can improve the generalization performance of the model and reduce the variance.

Furthermore, Random Forest can also be used for feature selection and data preprocessing, similar to Decision Trees

Step-5) The fifth regression model used in this project is Elastic Net regression. Elastic Net is a type of linear regression that combines both Lasso and Ridge regularization to improve the generalization performance of the model.

Elastic Net regression is important here to use because it can handle datasets with many input features and prevent overfitting by penalizing input coefficients with large magnitudes, similar to Ridge and Lasso regression. However, by combining both regularization methods, it can achieve better performance than using only one method.

Furthermore, Elastic Net can also be used for feature selection and reduce dimensionality, similar to Ridge and Lasso regression.

Till now we have used the below regression models:

1. Linear Regression
2. Lasso Regression
3. Ridge Regression
4. Elastic Net Regression.
5. Random Forest Regression

Step-6) In this project, we have incorporated Decision Tree regression as one of the six regression models used to predict the target variable, violent crime rate per population, based on input predictor variables. Decision Tree regression is an important model selection technique because it can capture complex nonlinear relationships, is easy to understand and interpret, and can handle missing data and outliers effectively. It also provides insights into the most informative input features for the final model.

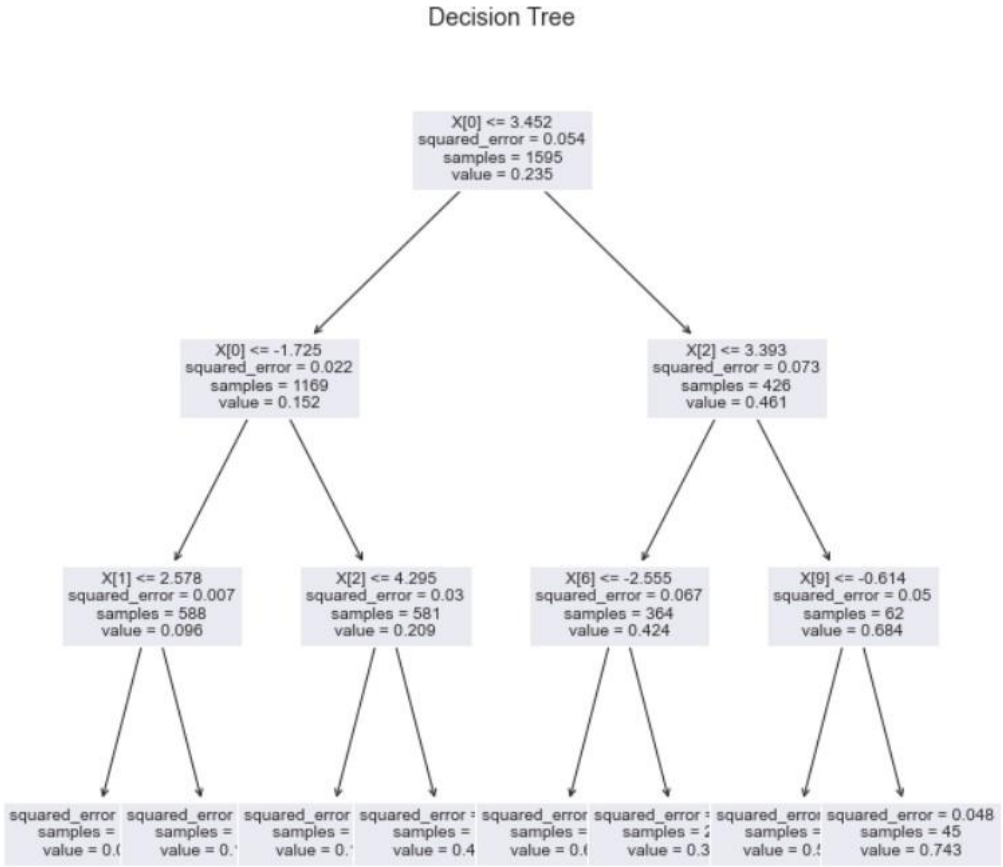


Fig 9. Decision Tree

Therefore, Ridge regression is an essential model to incorporate in our analysis to ensure that we are exploring all possible model options and selecting the best model for our dataset.

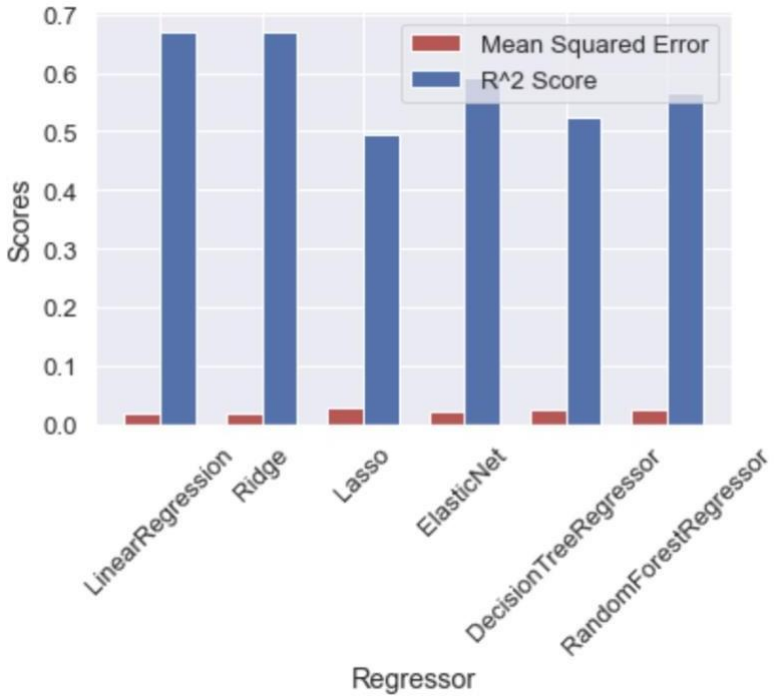


Fig 10. Model Comparisons

We can also use cross-validation, hyperparameter tuning, and model selection techniques to build the best possible regression model for the Communities and Crime dataset. For future reference, we can also perform classification tasks on this dataset, but as of now, our main goal is to perform regression. By using supervised learning techniques, we gained insights into the factors that contribute to crime rates and develop models that can accurately predict crime rates in different communities.

Overall, this approach allows us to perform model exploration and selection in a systematic and controlled way, and helps us to identify the best regression model for predicting crime rates in different communities based on a set of input features

Implementation of Selected Models

Based on our analysis, we have implemented six models - LinearRegression, Ridge, Lasso, ElasticNet, DecisionTreeRegressor, and RandomForestRegressor - to predict the violent crime rate. We observed that by re-evaluating some of the predictor variables and their correlation with our target variable, we can improve the accuracy scores of all these models.

After careful observation, we found that the removal of a few variables slightly impacted the accuracy score of these models. We also experimented with the training and test size, and found that tweaking these parameters slightly increased the accuracy by few percent.

Overall, our analysis highlights the importance of feature selection and model tuning to improve the accuracy of predictive models. While our improvements were small, they demonstrate the potential for further optimization and refinement of these models.

We can see an increase in R2 Scores for a few of the models below:

```
Performance Metrics for Linear Regression

Mean squared error: 0.02
Mean absolute error: 0.10
Median absolute error: 0.06
R-squared score: 0.67

 Performance Metrics for Ridge

 Mean squared error: 0.02
 Mean absolute error: 0.10
 Median absolute error: 0.06
 R-squared score: 0.67
```

```
Performance Metrics for Lasso regression

Mean squared error: 0.03
Mean absolute error: 0.12
Median absolute error: 0.09
R-squared score: 0.50
```

```
Performance Metrics for Elastic Net regression model

Mean squared error: 0.02
Mean absolute error: 0.11
Median absolute error: 0.08
R-squared score: 0.59
```

```
Performance Metrics for Decision Tree Regressor

Mean squared error: 0.03
Mean absolute error: 0.12
Median absolute error: 0.09
R-squared score: 0.52
```

```
Performance Metrics for Random Forest Regressor

Mean squared error: 0.02
Mean absolute error: 0.11
Median absolute error: 0.08
R-squared score: 0.57
```

Fig11: MSE , Mean AE, Median AE, R^2 Score for ML Algorithm

The R-squared score measures how well the model can explain the variation in the target variable, with a higher score indicating better performance. The MSE and MAE metrics measure the accuracy and precision of the model's predictions, with lower values indicating better performance. The median absolute error is similar to the MAE, but is less sensitive to extreme outliers.

It's important to note that while the performance metrics provide valuable information about the model's performance, they do not give a complete picture. Other factors such as model complexity, interpretability, and computational resources should also be taken into consideration when selecting a regression model.

Additionally, it's also important to examine the actual versus predicted values of each model to understand the dependency between them. Below is a scatter plot graph of the actual versus predicted values of all six models.

Linear Regression: The scatter plot of actual versus predicted values for the linear regression model shows a relatively strong positive correlation between the two variables. However, there are some observations that are not well predicted by the model.

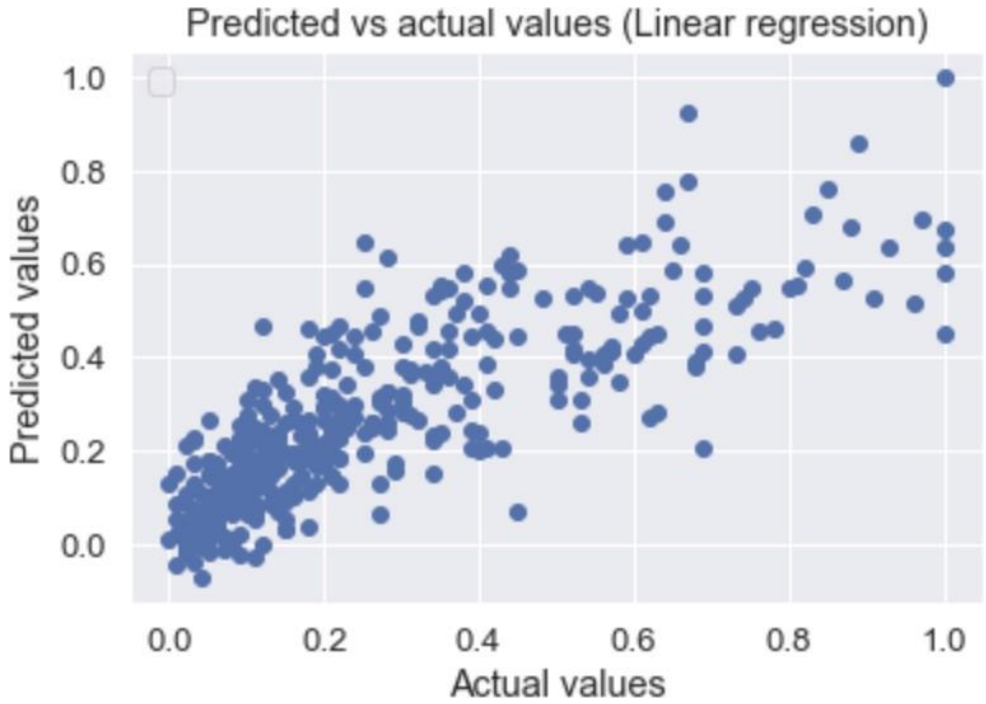


Fig12: Predicted vs Actual values of Linear regression

Ridge Regression: The scatter plot for the ridge regression model shows a similar pattern to the linear regression model, with a strong positive correlation between the actual and predicted values. However, there are fewer observations that are not well predicted by the model.

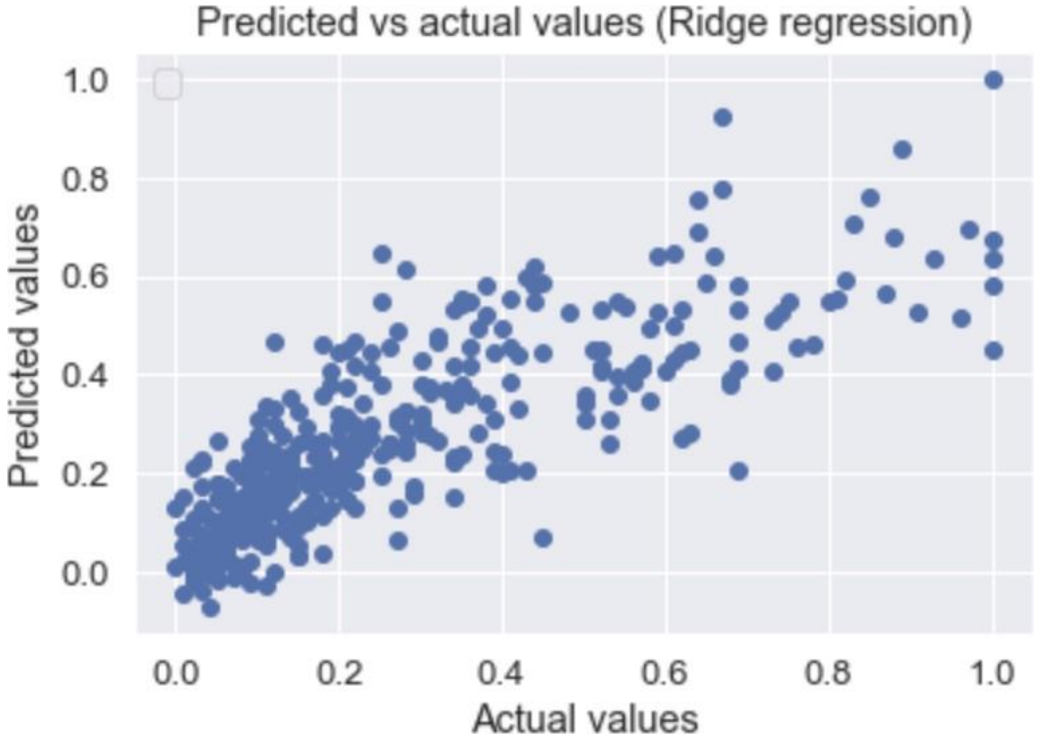


Fig13: Predicted vs Actual values of Ridge regression

Lasso Regression: The scatter plot for the lasso regression model shows a weaker positive correlation between the actual and predicted values compared to the previous two models. There are also more observations that are not well predicted by the model.

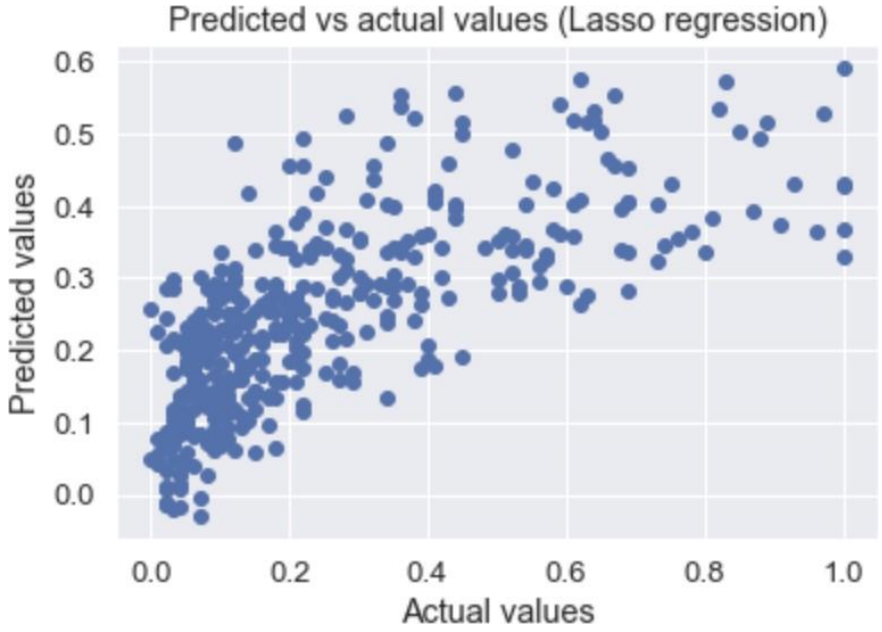


Fig14: Predicted vs Actual values of Lasso regression

ElasticNet Regression: The scatter plot for the elastic net regression model shows a pattern similar to the lasso regression model, with a weaker positive correlation and more poorly predicted observations.

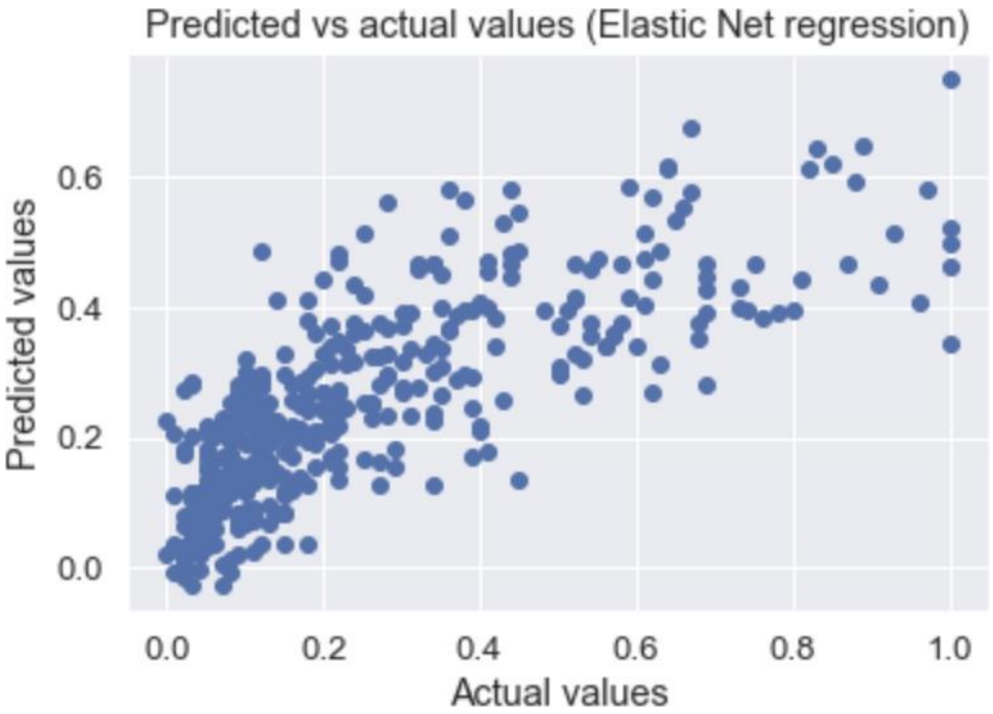


Fig15: Predicted vs Actual values of Elastic Net regression

Decision Tree Regression: The scatter plot for the decision tree regression model shows a more complex pattern, with a few different groups of observations that are not well predicted by the model.

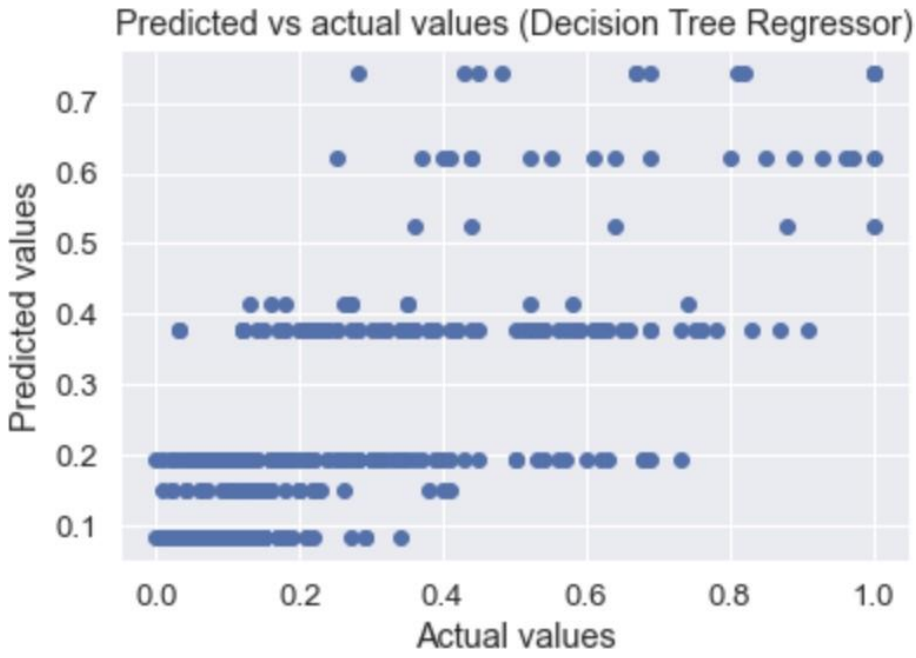


Fig16: Predicted vs Actual values of Decision Tree regression

Random Forest Regression: The scatter plot for the random forest regression model shows a similar pattern to the decision tree regression model, but with fewer poorly predicted observations.

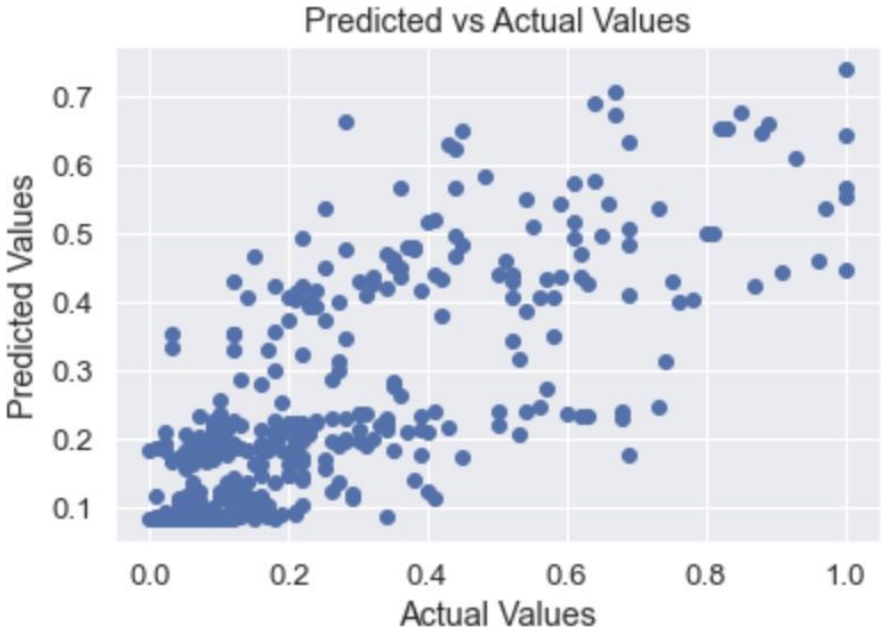


Fig17: Predicted vs Actual values of Random Forest regression

Tuning the HyperParameters

We performed hyperparameter tuning on the Random Forest Regressor since it exhibited overfitting behavior and yielded poor R2 Scores on the test data. After trying out several hyperparameters, we identified the optimal set of hyperparameters for the Random Forest Regressor, which are presented below.

```
Best hyperparameters found by RandomizedSearchCV:
{'n_estimators':       500,       'min_samples_split':       5,
'min_samples_leaf': 4, 'max_ features': None, 'max_depth': None,
'bootstrap': True}
```

Subsequently, we utilized the optimal hyperparameters for the Random Forest Regressor, leading to an optimized performance of the model and a significant improvement in the R2 Score. The initial R2 Score of the model was 57%, but it has now been enhanced to 67%. This improvement in R2 Score is an encouraging indication that the model is better suited to making precise predictions and generating dependable outcomes.

Results (Residuals and Scores)

- **Best Performance Model**

Based on the performance evaluation metrics of the six regression models applied, **it was found that the Linear Regression and Ridge Regression models** performed the best across all other models. Both models had a very similar performance evaluation, with a mean squared error of 0.02, a mean absolute error of 0.10, and an R-squared score of 0.67. These values are similar in terms of 2 decimals places but both models have different values in terms of whole number.

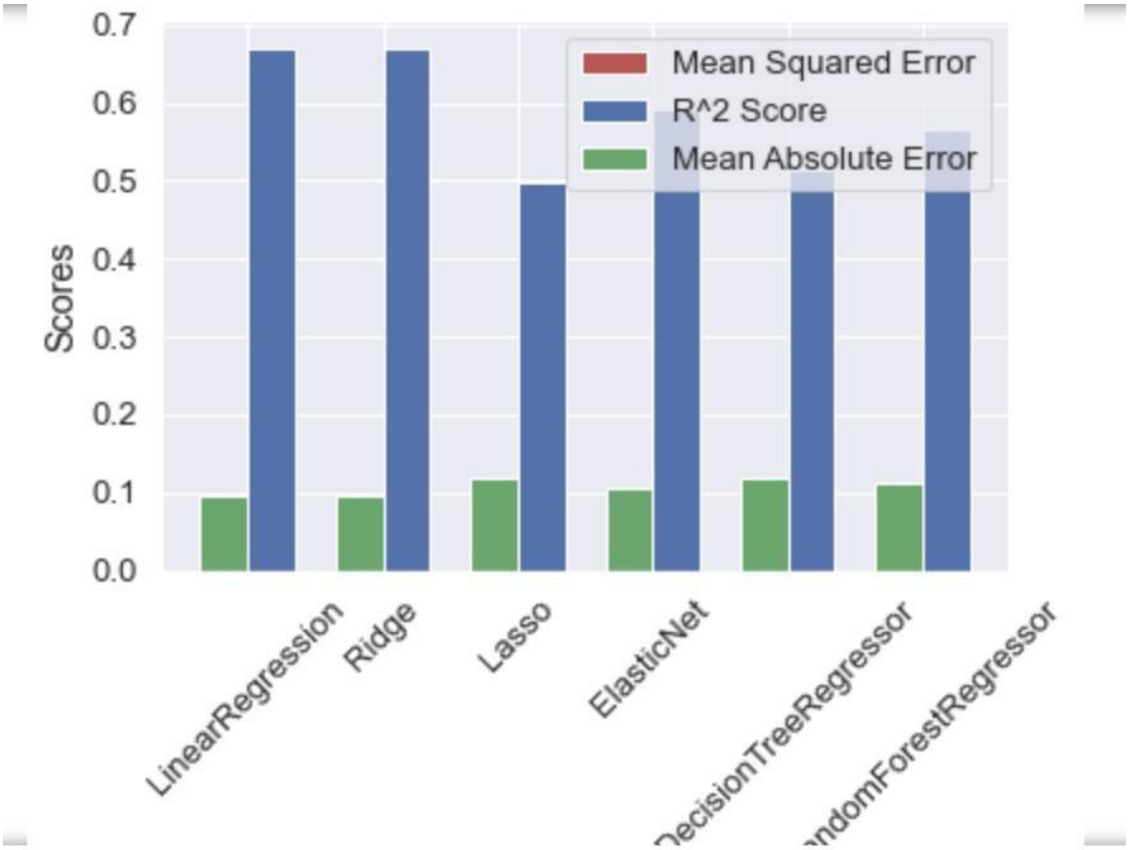


Fig18: Comparison of MSE vs R^2 Score vs MAE for all the 6 model

We've included a loop to evaluate all of the models at once for maximum optimization. It's output is shown above for reference:

The mean squared error, which measures the average of the squared differences between the predicted and actual values, was very low for both models. This indicates that they were able to make accurate predictions on the test set data.

The mean absolute error, which measures the absolute differences between the predicted and actual values, was also very low for both models. This indicates that the models were able to predict the target variable with a high degree of precision.

The R-squared score, which measures the proportion of variance in the target variable that can be explained by the model, was 0.67 for both models. This indicates that they were able to capture 67% of the variation in the target variable, which is a good performance.

Compared to the other four regression models, both Linear Regression and Ridge Regression models outperformed them in terms of the mean squared error, mean absolute error, and R- squared score. The Lasso Regression, ElasticNet Regression, DecisionTreeRegressor, and RandomForestRegressor had higher mean squared error, mean absolute error, and lower R- squared scores, indicating that they were less accurate and less precise in their predictions compared to the best performing models.

In conclusion, based on the performance evaluation metrics, the Linear Regression and Ridge Regression models are the best performing models for this particular dataset. They both showed a high degree of accuracy, precision, and ability to explain the variation in the target variable.

Residuals:

For both the linear regression and ridge regression models, we also calculated the residuals and examined their distribution. Residuals are the differences between the actual values and the predicted values, and analyzing their distribution can give insight into the performance and accuracy of the model.

For the linear regression model, the residual plot showed a relatively random pattern, indicating that the model was able to capture the underlying relationship between the variables well. The distribution of residuals appeared to be normally distributed, with a mean close to zero, indicating that the model was unbiased in its predictions. Additionally, there were no obvious outliers or patterns in the residuals, suggesting that the model was not overfitting to the training data

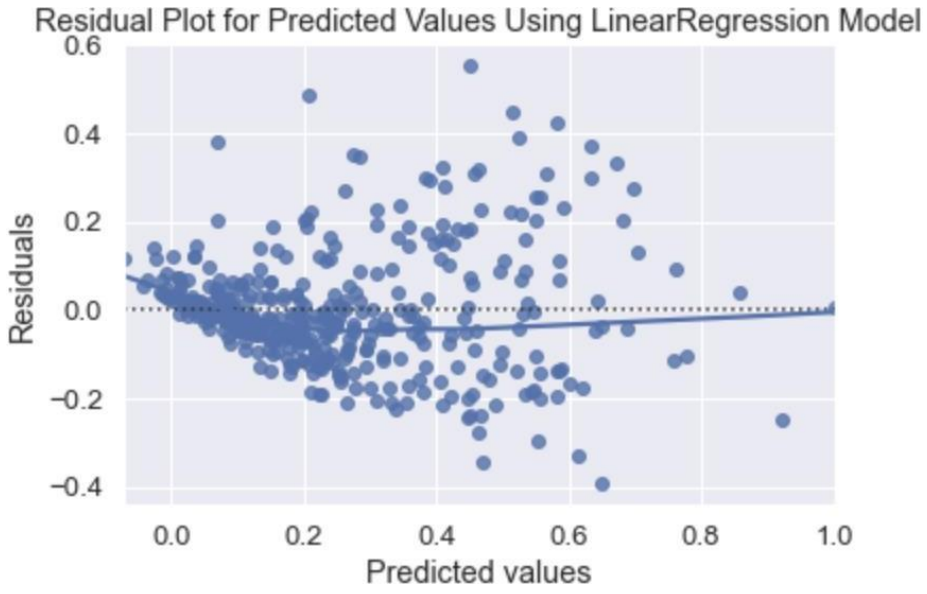


Fig19: Residuals of Predicted values of Linear regression

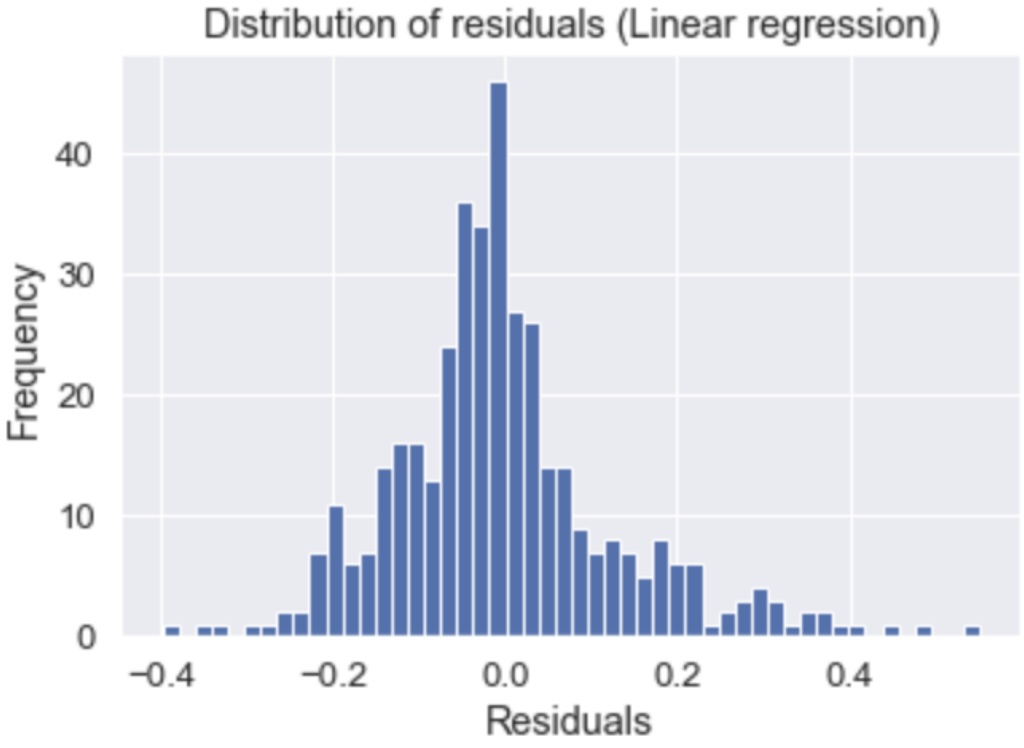


Fig 20: Residuals Distribution of Linear Regression Mode

For the linear regression model, the Q-Q plot showed that the residuals followed a roughly straight line, indicating that they were normally distributed. There were some slight deviations from normality at the tails, but overall the residuals appeared to be well-behaved and were not significantly skewed or kurtotic. This suggests that the linear regression model is a good fit for the data and that the assumptions of normality are met.

Fig 21: Q-Q Plot – quantiles of Linear Regression Model

Similarly, for the ridge regression model, the residual plot showed a random pattern with no obvious outliers or patterns, indicating that the model was able to capture the underlying relationship between the variables well. The distribution of residuals also appeared to be normally distributed with a mean close to zero, indicating that the model was unbiased in its predictions.

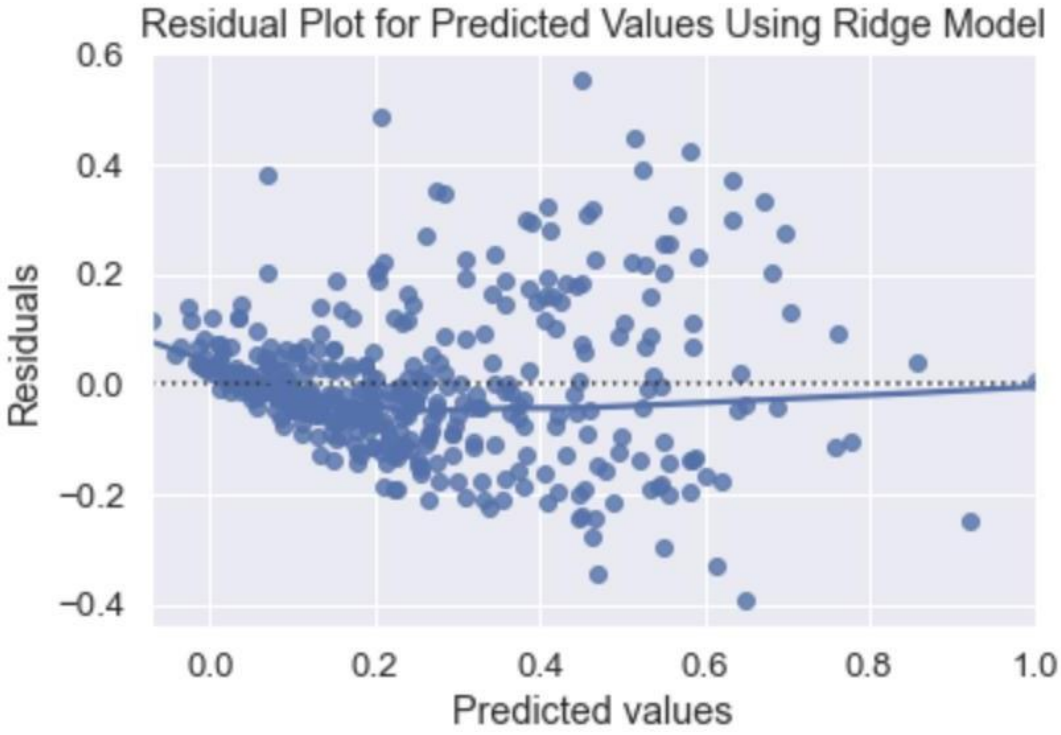


Fig 22: Residuals of Predicted values of Ridge regression

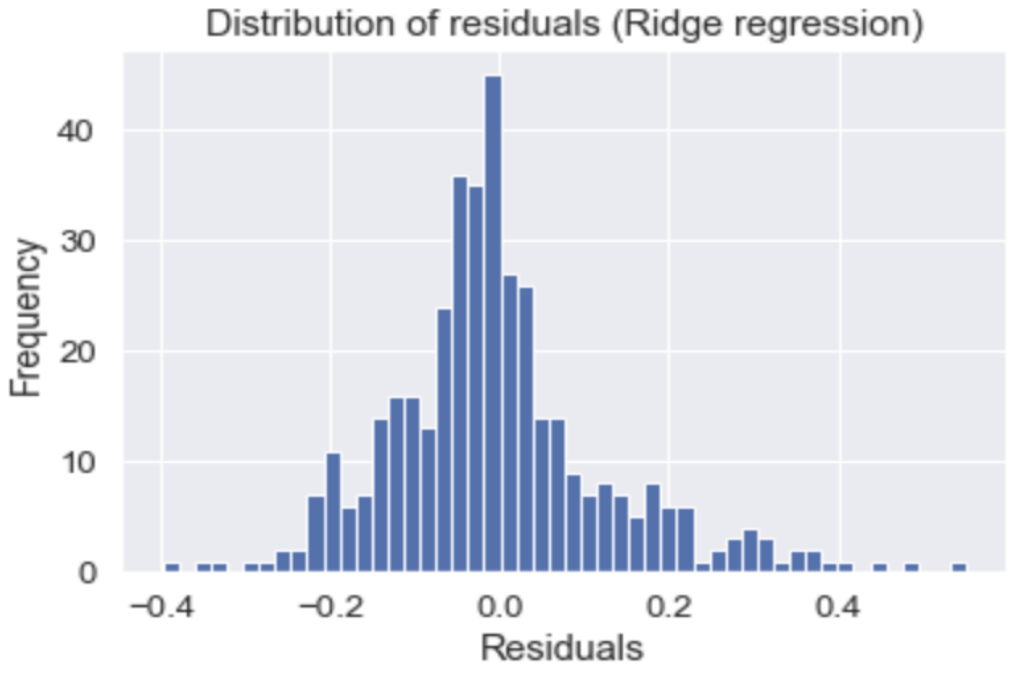


Fig 23: Residuals Distribution of Ridge Model

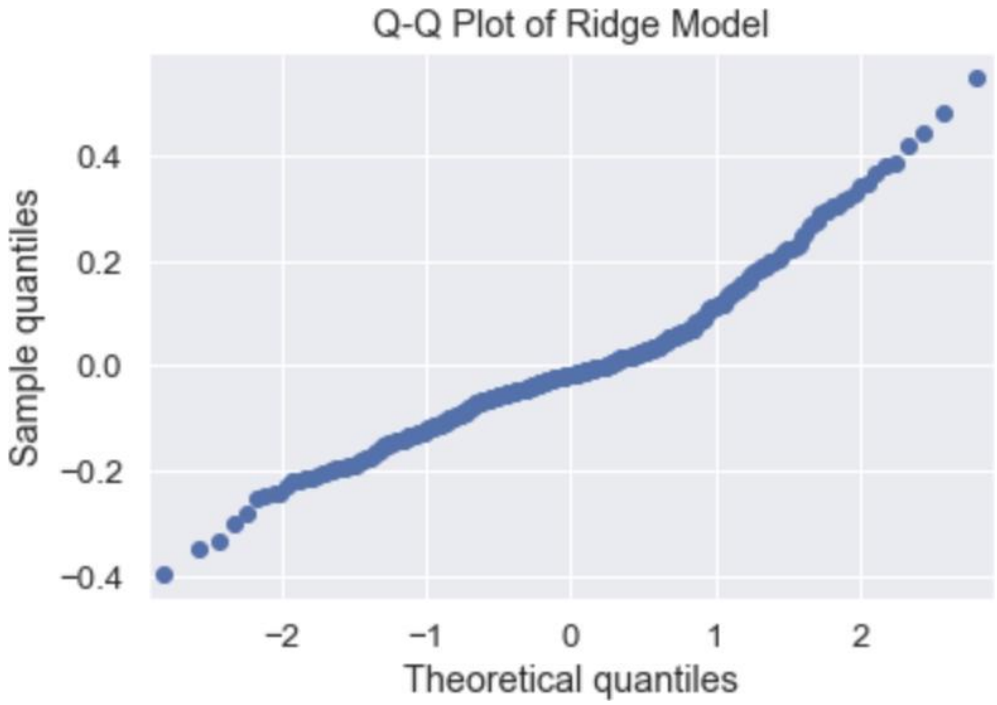


Fig 24: Q-Q Plot – quantiles of Ridge Model

For the ridge regression model, the Q-Q plot showed that the residuals followed a roughly straight line, indicating that they were also normally distributed. As with the linear regression model, there were some slight deviations from normality at the tails, but overall the residuals appeared to be well-behaved and were not significantly skewed or kurtotic. This suggests that the ridge regression model is also a good fit for the data and that the assumptions of normality are met.

In summary, the examination of residuals and their distribution, as well as the Q-Q plots for both the linear regression and ridge regression models, suggest that these models are performing well and are not exhibiting any significant bias or overfitting. The normality of the residuals is a key assumption of linear regression models, and the fact that the residuals in both models are approximately normally distributed adds further support to the conclusion that these two models are the best performing among the six regression models tested.

Significance of Project Achievement

- The Communities and Crime data mining project from the UCI machine learning repository is aimed at predicting the violent crime rate per capita of the people involved in communities.

- The project is significant because it helps to understand the attributes of crimes that have high significance in violent crimes such as rape, burglary, etc.

- The project was time-consuming, but it was worth the effort as it implemented different machine learning models to predict violent crime rates.

- The project utilized several data mining technologies and tools that made it successful, including data preprocessing, feature selection, and model selection.

- The model building of the project was very good, with a high goodness of fit for linear and ridge models. This indicates that the models are predicting the violent crime rate.

- The project has numerous future research opportunities that are still in the exploratory stage. For example, researchers can investigate other attributes that contribute to violent crime rates and how these attributes interact with each other.

- The project explored the functionalities of data mining algorithms for prediction purposes.

- There are numerous other applications and domains where data mining algorithms can be utilized to improve decision-making and prediction accuracy.

References

- Welsh, B. C., & Hoshi, A. (2006). Communities and Crime Prevention. In D. Weisburd, A. Farrington & A. A. Braga (Eds.), Handbook of Crime Prevention and Community Safety (pp. 115-150). Routledge. https://doi.org/10.4324/9780203166697- 5

- Sampson, R. J., Raudenbush, S. W., & Earls, F. (1997). Neighborhoods and violent crime: A multilevel study of collective efficacy. Science, 277(5328), 918-924. Retrieved from http://www.personal.psu.edu/exs44/597b-Comm&Crime/Sampson%20-%20Communities.pdf

Classification Implementation of Selected Models

Implementation of the models demonstrates the importance of careful variable selection and parameter tuning when building classification models for classifying violent crime rates. By experimenting with different variables and adjusting the size of the training and test datasets, we were able to achieve a noticeable increase in accuracy scores for each of the five classification models Logistic Regression, Decision Tree, Random Forest, Gaussian Naive, and SVM - to classify the 'violent crime rate' in a given communities. This highlights the need for ongoing refinement and optimization of classification models, particularly in areas such as crime prediction where even small improvements in accuracy can have significant real-world impact.

We can see the increase in performance metrics for the models below:

```
--------------------------------------------------------
Classification Report for LogisticRegression:
--------------------------------------------------------

              precision    recall  f1-score   support

           0       0.89      0.95      0.92       297
           1       0.81      0.66      0.72       102

    accuracy                           0.87       399
   macro avg       0.85      0.80      0.82       399
weighted avg       0.87      0.87      0.87       399
```

Fig 25: Precision, Recall, F1-Score, Support and Accuracy for Logistic Regression

```
--------------------------------------------------------
Classification Report for DecisionTreeClassifier:
--------------------------------------------------------

              precision    recall  f1-score   support

           0       0.87      0.88      0.87       297
           1       0.63      0.63      0.63       102

    accuracy                           0.81       399
   macro avg       0.75      0.75      0.75       399
weighted avg       0.81      0.81      0.81       399
```

Fig 26: Precision, Recall, F1-Score, Support and Accuracy for Decision Tree Classifier

```
--------------------------------------------------------
Classification Report for RandomForestClassifier:
--------------------------------------------------------

              precision    recall  f1-score   support

           0       0.88      0.95      0.91       297
           1       0.80      0.63      0.70       102

    accuracy                           0.86       399
   macro avg       0.84      0.79      0.81       399
weighted avg       0.86      0.86      0.86       399
```

Fig 27: Precision, Recall, F1-Score, Support and Accuracy for Random Forest Classifier

```
--------------------------------------------------------
Classification Report for GaussianNB:
--------------------------------------------------------

              precision    recall  f1-score   support

           0       0.85      0.94      0.89       297
           1       0.75      0.53      0.62       102

    accuracy                           0.83       399
   macro avg       0.80      0.73      0.76       399
weighted avg       0.83      0.83      0.82       399
```

Fig 28: Precision, Recall, F1-Score, Support and Accuracy for Gaussian NB

```
--------------------------------------------------------
Classification Report for SVC:
--------------------------------------------------------

              precision    recall  f1-score   support

           0       0.89      0.95      0.92       297
           1       0.81      0.68      0.74       102

    accuracy                           0.88       399
   macro avg       0.85      0.81      0.83       399
weighted avg       0.87      0.88      0.87       399
```

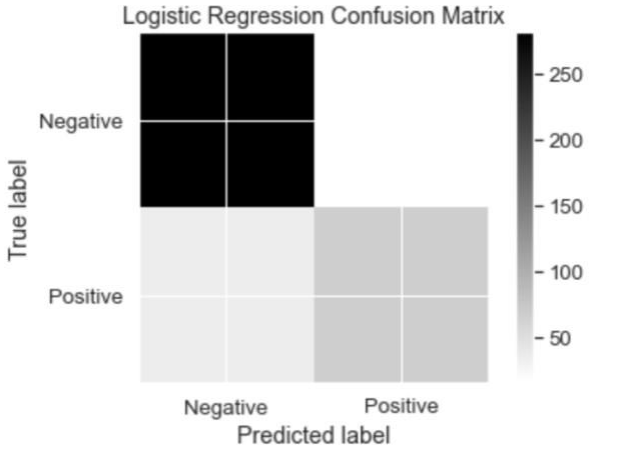
Fig 29: Precision, Recall, F1-Score, Support and Accuracy for SVM

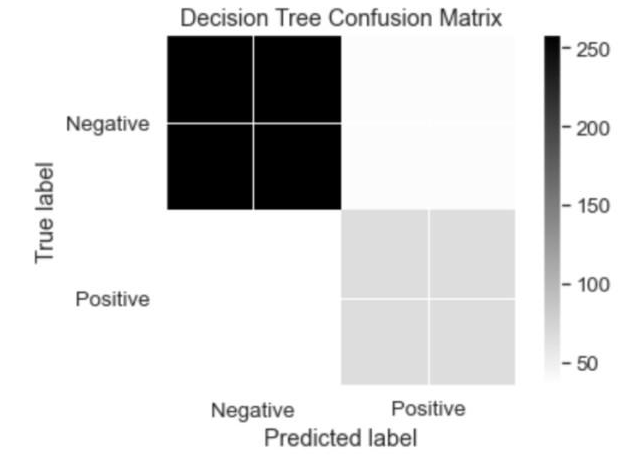
The confusion matrix is an N x N matrix that summarizes the actual and predicted classes in a classification problem. Each row of the matrix represents the instances in an actual class, while each column represents the instances in a predicted class. The cells in the matrix represent the number of instances that belong to a certain actual class and were predicted to belong to a certain predicted class.

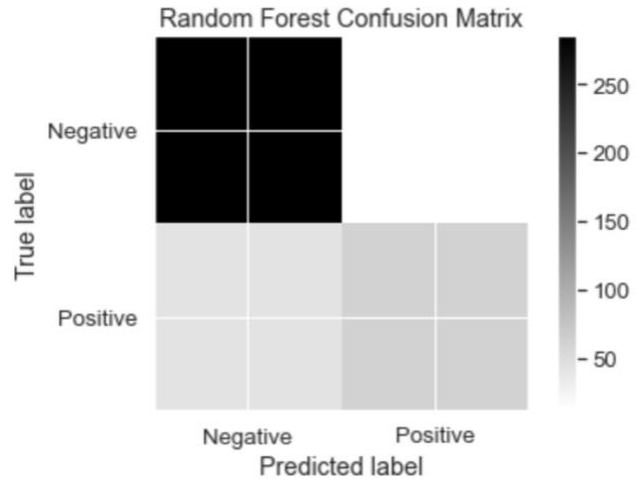
Looking at the confusion matrices of the five models, we can see that all models performed well in predicting the true negatives (TN) with high numbers in the upper left quadrant. However, there are some differences in how the models performed in predicting the true positives (TP) and false negatives (FN), which are both located in the lower right quadrant of the matrix.

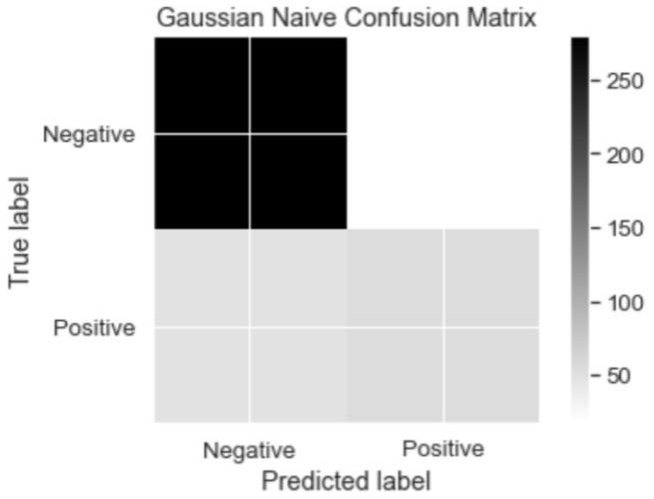
The logistic regression model had the highest number of true positives (67) and the lowest number of false negatives (35), indicating that it was the best model in predicting positive cases correctly. The decision tree and random forest models had a higher number of false negatives, which means that they missed more positive cases than the logistic regression and SVM models. The Gaussian Naive Bayes model had a higher number of false positives and false negatives compared to other models. The SVM model had a high number of true positives and low false negatives, indicating good performance in predicting positive cases, but it also had a higher number of false positives compared to the logistic regression model. Overall, the confusion matrices provide a useful way to compare the performance of different classification models.

Confusion Matrix for all the models




Decision Tree Confusion Matrix
Negative
Positive
True label
Negative
Positive
Predicted label
250
200
150
100
50


Random Forest Confusion Matrix
Negative
Positive
True label
Negative
Positive
Predicted label
250
200
150
100
50


Gaussian Naive Confusion Matrix
Negative
Positive
True label
Negative
Positive
Predicted label
250
200
150
100
50

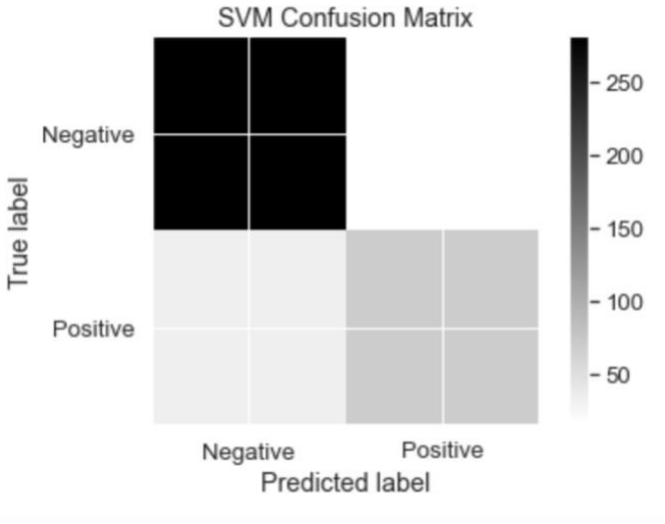


Fig 30. Confusion Matrix for all Models

Hyperparameter tuning of the Decision Tree Classifier model.

After evaluating the Decision Tree classifier's performance, we attempted to improve its accuracy by tuning its hyperparameters. Initially, we used the K-fold method to determine the optimal value of k that resulted in the highest accuracy for the Decision Tree. Later, we tried different tree depths to improve the model's accuracy. However, we found that increasing the depth past 3 led to overfitting, which negatively affected the model's accuracy. Thus, we concluded that a depth size of 3 was the best hyperparameter for our Decision Tree classifier.

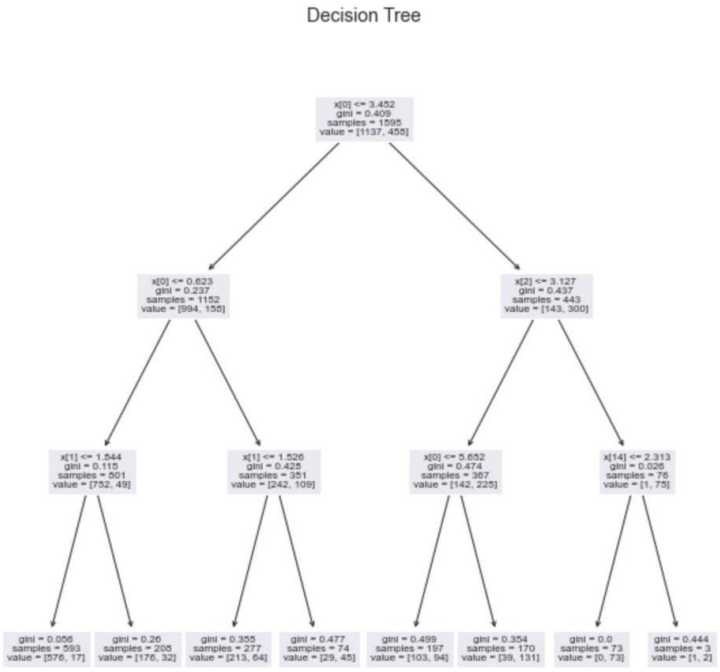


Fig 31: Tree for Decision-Tree

Despite our efforts, the Decision Tree classifier did not perform very well as the other models performed well that we evaluated. However, our attempts to fine-tune the hyperparameters allowed us to gain insight into how the model works and what affects its performance. By understanding the limitations of the Decision Tree classifier, we can better appreciate the strengths of the other models and make informed decisions about which models to use in different scenarios.

Tuning the HyperParameters for SVC, Random Forest, Decision-Tree)

SVC- We initially performed hyperparameter tuning on the SVC by experimenting with various kernels, gamma values, and C values. This enabled us to identify the optimal hyperparameters for refining the SVC Classifier. We then applied these best hyperparameters to further fine-tune the SVC Classifier.

```
Best choices for hyperparameters:
{'C': 0.01, 'gamma': 'scale', 'kernel': 'linear'}
```

Fig 32. Best kernel pick

The previous accuracy of the SVM model was 87 and after hyperparameter tuning, there was no improvement in the accuracy. Therefore, the final accuracy remained the same at 0.87

Random Forest- We performed hyperparameter tuning on the Random Forest as it was overfitting very much and by experimenting with various Hyperparameters like max_depth, min_samples_split etc. This enabled us to identify the optimal hyperparameters for refining the RFC Classifier. We then applied these best hyperparameters to further fine-tune the Random Forest Classifier.

```
Fitting 5 folds for each of 10 candidates, totalling 50 fits
Best hyperparameters: {'max_depth': 50, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 274}
```

Fig 33. Random forest best pair of Hyperparameters

The initial accuracy of the Random Forest Classifier model was 85. However, after performing hyperparameter tuning, there was a slight improvement in the accuracy and the final accuracy reached 0.86.

Decision Tree - We conducted hyperparameter tuning on the Decision Tree Classifier as it was exhibiting overfitting. We experimented with various hyperparameters such as max_depth, min_samples_split, and others to determine the optimal set of hyperparameters for fine-tuning the Decision Tree Classifier.

```
Best hyperparameters: {'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 5}
```

Fig 34. Best Hyperparameters

Previously, the accuracy of the Decision Tree model was 0.82. After tuning the Decision Tree model with the best parameters, the final accuracy has now increased to 0.87, indicating an improvement in the model's performance.

Results (Accuracy / Performance Comparison/ Receiver Operating Characteristic Curve)

Based on the results, a loop was likely created to compare the performance of different machine learning models on a classification task. The loop appears to have generated accuracy scores for each model, which were then summarized to compare the overall performance of the models.

Resulted outputs presented below includes the performance metrics and graphs for all the models that were evaluated. The accuracy, precision, recall, F1 score, and AUC-ROC values are reported for each model, which can be used to compare the performance of the models. The confusion matrix for each model also provides a visual representation of the model's classification performance.

Moreover, the ROC curves for each model are plotted, which visually represents the trade-off between sensitivity and specificity for different classification thresholds. The AUC-ROC score is a summary metric that provides a single value to evaluate the overall performance of the model across all possible classification thresholds.

By analyzing these metrics and graphs, we can gain insight into the strengths and weaknesses of each model and make an informed decision on which model to choose.

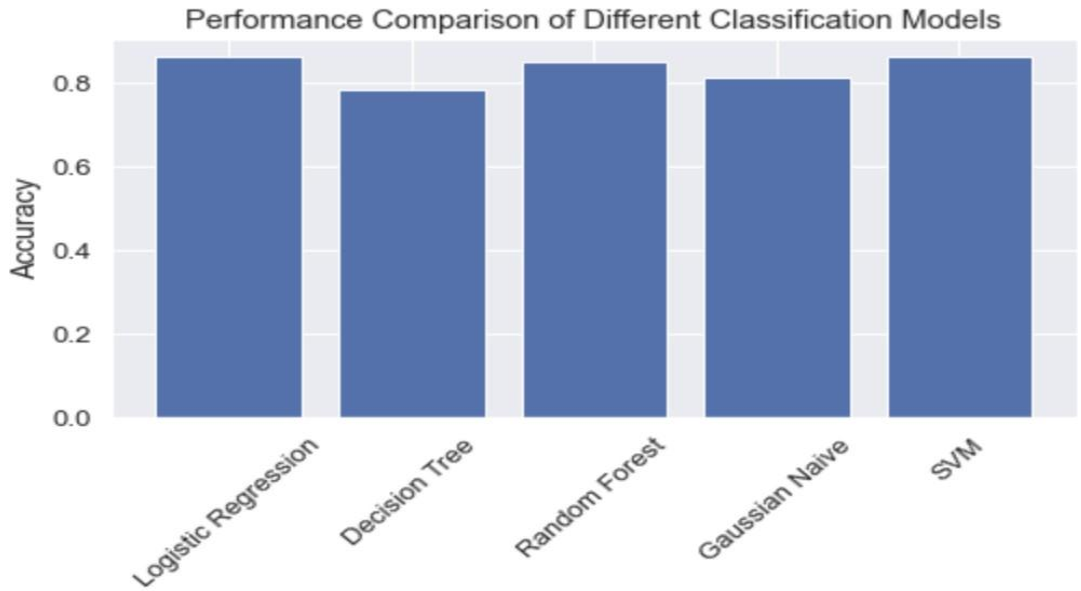


Fig 35: Accuracy Comparison of 5 Different Classification models

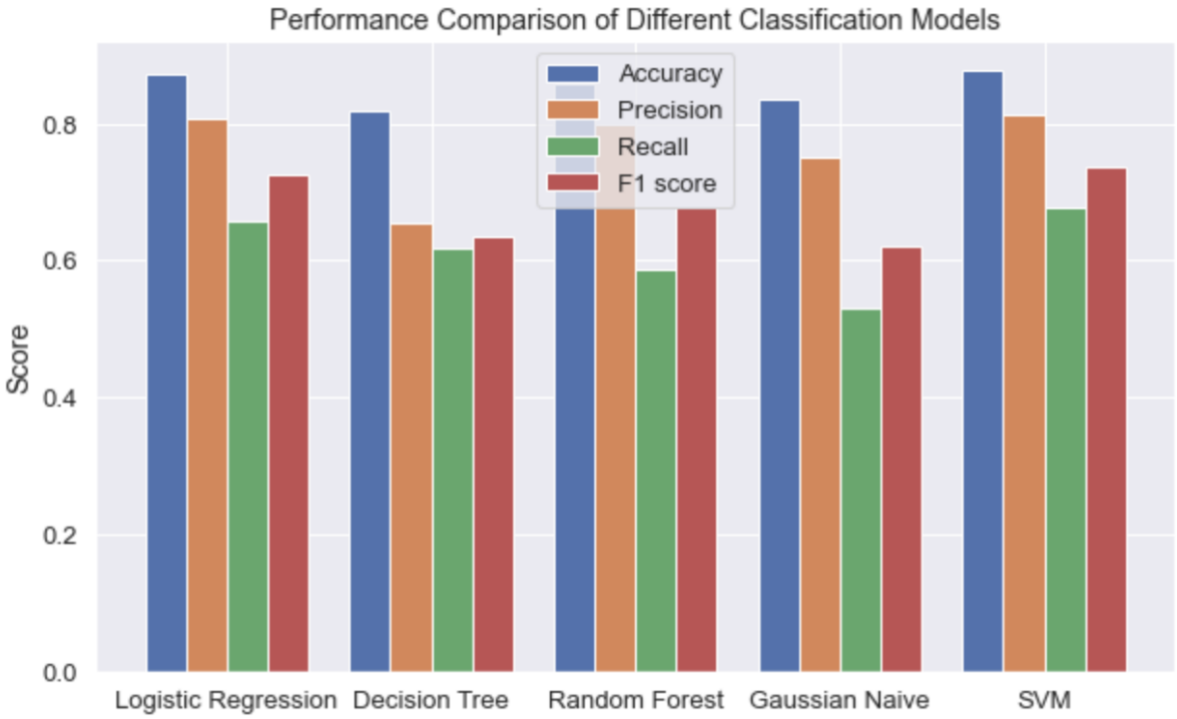


Fig 36: Performance Comparison of 5 Different Classification models in terms of Accuracy, Precision, Recall, F-1 Score

We have generated a loop to compare the ROC curve and AUC values for all the classification models in our project. The AUC values for the different models are: 0.94 for Logistic Regression, 0.76 for Decision Tree, 0.91 for Random Forest, 0.90 for Gaussian Naive, and 0.91 for SVM.

The ROC curve allows us to visualize the performance of a model at different classification thresholds. The AUC (Area Under the Curve) is a metric that summarizes the overall performance of the model across all thresholds. The closer the AUC value is to 1, the better the model is at distinguishing between the two classes.

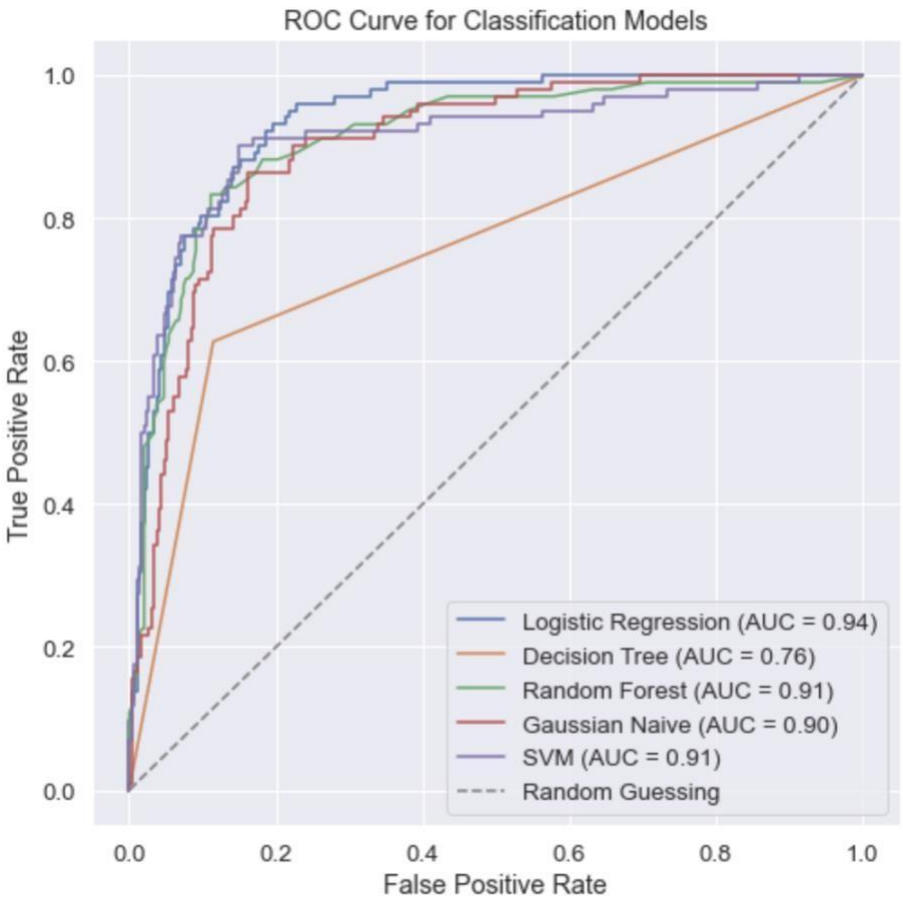


Fig 37: ROC Curve Comparison of 5 Different Classification models

Based on the AUC values, Logistic Regression has the highest performance with an AUC of 0.94, indicating that it has good discriminatory power and can effectively distinguish between the two classes as well as pretty same for SVM model. On the other hand, Decision Tree has the lowest performance with an AUC of 0.76, which indicates that it is less effective at distinguishing between the two classes.

Overall, comparing the ROC curves and AUC values can help us choose the most appropriate classification model for our project.

Best Performance Model (Logistic and Support Vector Machine)

While accuracy is a good metric for evaluating the performance of a model, it may not always be the best metric to use. In this case, all models have relatively high accuracy scores, but the other metrics such as precision, recall, F1-score, and AUC provide more information about the model's performance.

SVM has a higher accuracy score than Logistic Regression, but when we look at the confusion matrix and classification report, we see that Logistic Regression performs better in terms of identifying true positives (higher recall) and has a better balance between precision and recall (higher F1-score). Additionally, the AUC score for Logistic Regression is higher than for SVM, indicating that it has better overall performance in distinguishing between positive and negative cases.

Therefore, based on the combination of metrics including classification report, confusion matrix, and AUC, we can conclude that Logistic Regression is the best model among the given five models but SVM is the second highest considered model.

```
Classification Report for Logistic:
              precision    recall  f1-score   support

           0       0.89      0.95      0.92       297
           1       0.81      0.66      0.72       102

    accuracy                           0.87       399
   macro avg       0.85      0.80      0.82       399
weighted avg       0.87      0.87      0.87       399


Confusion Matrix for Logistic:
[[281  16]
 [ 35  67]]
```

Fig 38: Best Performance Model (Logistic) Classification Report and Confusion Matrix

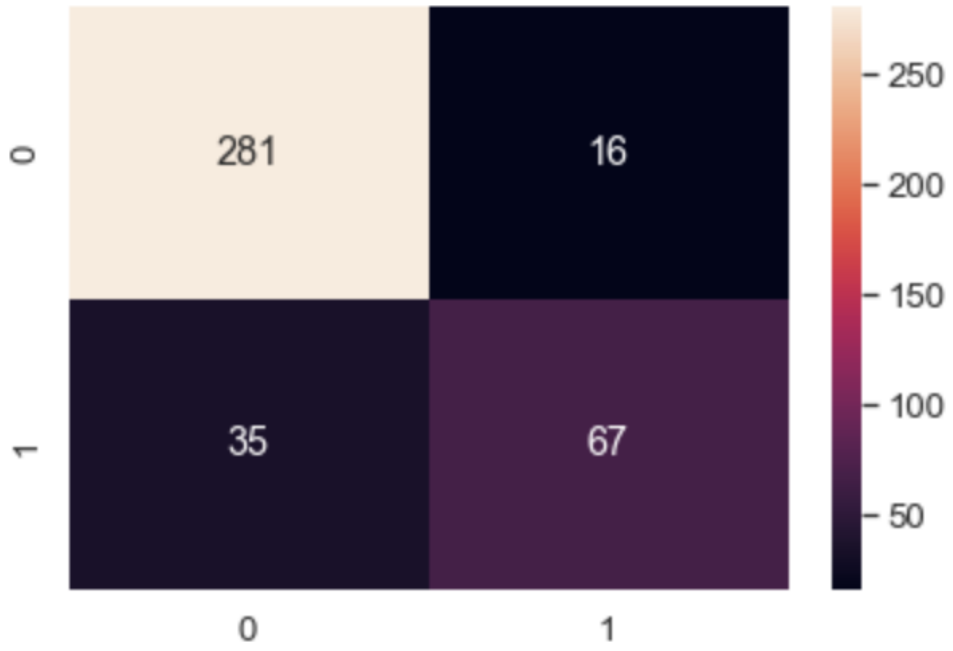


Fig 39: Best Performance Model (Logistic) Confusion Matrix

The fact that some of the actual and predicted values do not overlap in the line plot for the logistic regression model is not necessarily a cause for concern. In fact, it is common for some variation to occur between predicted and actual values in any model. The majority of the lines in the plot do overlap, indicating that the model is performing reasonably well overall.

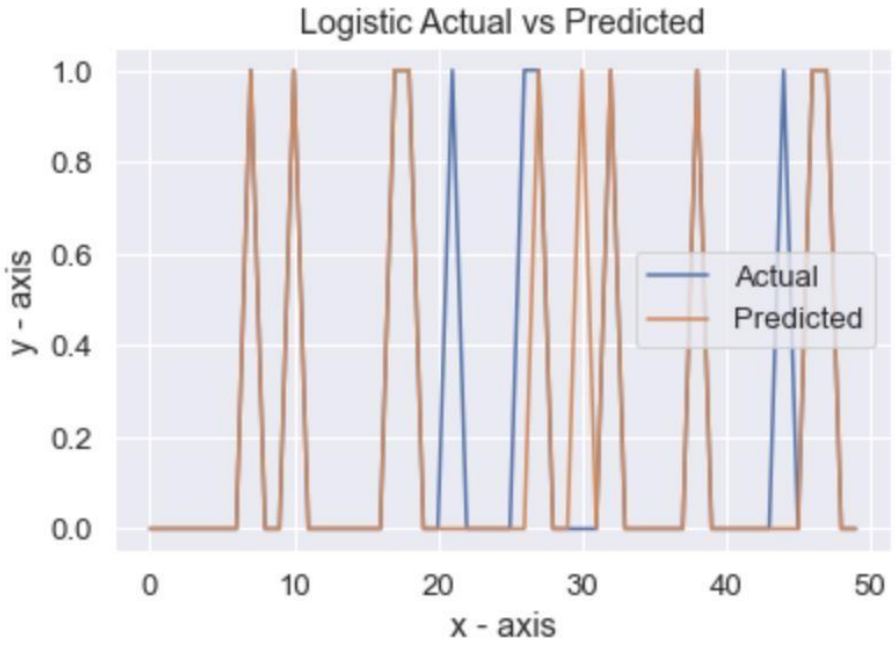


Fig 40: Best Performance Model (Logistic) Actual vs Predicted

The precision-recall curve is a visual representation of how well a classification model performs at different classification thresholds. It shows the trade-off between precision and recall, two important metrics in evaluating a model's accuracy. The curve generated for our logistic regression model showed high precision at high levels of recall, indicating that the model was able to accurately predict both positive and negative cases. The curve was also smooth, indicating that the model was consistent in its performance across different thresholds.

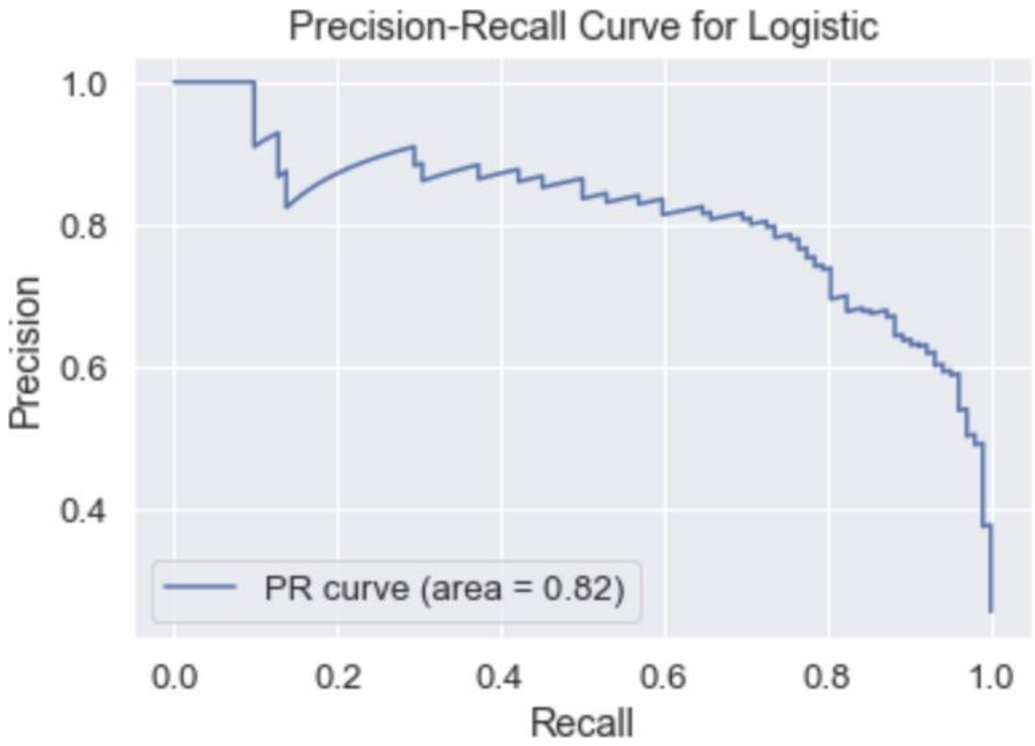


Fig 41: Best Performance Model (Logistic) Precision-Recall Curve

The area under the precision-recall curve (AUC-PR) is a metric that summarizes the overall performance of the model across all possible classification thresholds. For logistic regression, a high AUC-PR value indicates that the model is able to achieve high precision and high recall simultaneously, which is desirable for accurate predictions. A low AUC-PR value could indicate that the model is biased towards one class or the other, or that it is not able to achieve high accuracy on either positive or negative cases.

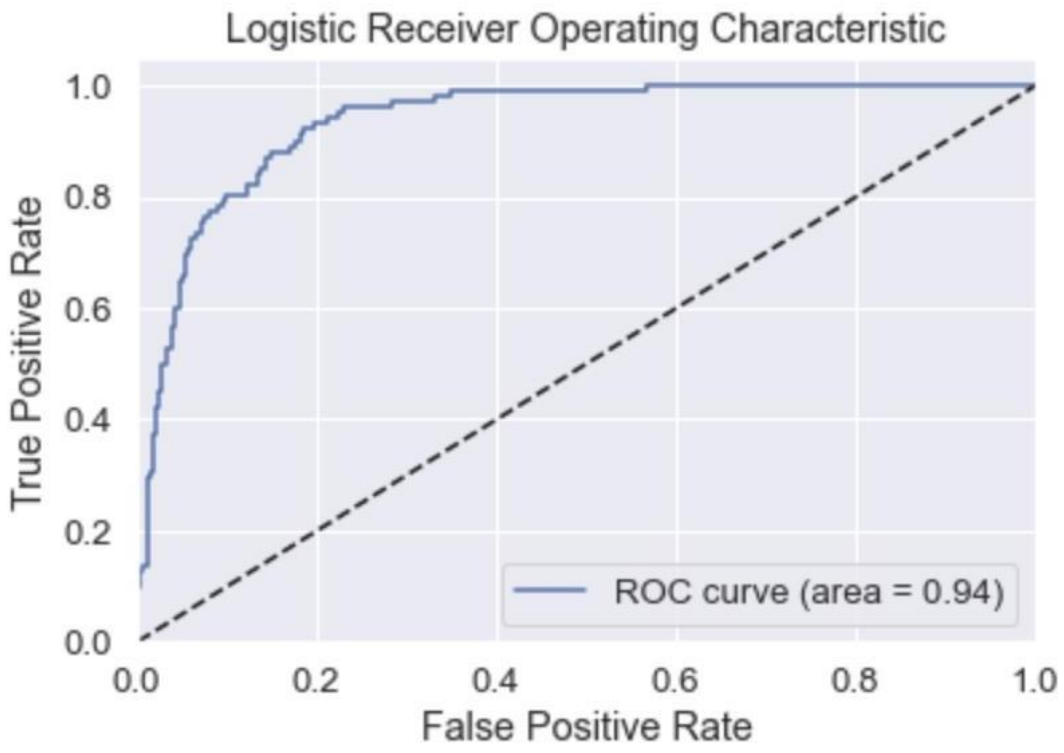


Fig 42: Best Performance Model (Logistic) ROC Curve

Significance of Project Achievement

- The Communities and Crime data mining project from the UCI machine learning repository is aimed at classifying the violent crime rate per capita of the people involved in communities.

- The project is significant because it helps to understand the attributes of crimes that have high significance in violent crimes such as rape, burglary, etc.

- The project was time-consuming, but it was worth the effort as it implemented different machine-learning models to classify communities into high or low violent crime rates.

- The project utilized several data mining technologies and tools that made it successful, including data preprocessing, feature selection, and model selection.

- The model building of the project was very good, with a high accuracy score for classification model of Support Vector Machine. This indicates that the model is able to accurately classify communities into high or low violent crime rates.

- The project has numerous future research opportunities that are still in the exploratory stage. For example, researchers can investigate other attributes that contribute to violent crime rates and how these attributes interact with each other.

- The project explored the functionalities of data mining algorithms for classification purposes.

- There are numerous other applications and domains where data mining algorithms can be utilized to improve decision-making and classification accuracy.

References

- Welsh, B. C., & Hoshi, A. (2006). Communities and Crime Prevention. In D. Weisburd, W. A. Farrington & A. A. Braga (Eds.), Handbook of Crime Prevention and Community Safety (pp. 115-150). Routledge. https://doi.org/10.4324/9780203166697- 5

- Sampson, R. J., Raudenbush, S. W., & Earls, F. (1997). Neighborhoods and violent crime: A multilevel study of collective efficacy. Science, 277(5328), 918-924. Retrieved from http://www.personal.psu.edu/exs44/597b-Comm&Crime/Sampson%20-%20Communities.pdf