Communities and Crime

**The Impact of Feature Selection Techniques on Support Vector Classification**

Table of Contents

[Introduction 3](#_Toc153650276)

[Literature Review 3](#_Toc153650277)

[Related Studies 3](#_Toc153650278)

[Feature Selection 4](#_Toc153650279)

[Filter Method 4](#_Toc153650280)

[Wrapper Method 5](#_Toc153650281)

[Embedded Method 5](#_Toc153650282)

[Communities and Crime Dataset 6](#_Toc153650283)

[Research Aims 6](#_Toc153650284)

[Methodology 6](#_Toc153650285)

[Data Preprocessing 6](#_Toc153650286)

[Data Normalization 6](#_Toc153650287)

[Exploratory Data Analysis 7](#_Toc153650288)

[Removal of Features 7](#_Toc153650289)

[Normality Test 8](#_Toc153650290)

[Changing Target Variable to Categorical Variable 8](#_Toc153650291)

[Splitting Data 9](#_Toc153650292)

[Missing/Null Values 9](#_Toc153650293)

[Feature Selection Techniques 9](#_Toc153650294)

[Performance Evaluation 11](#_Toc153650295)

[Accuracy Score of Zero-Folds 11](#_Toc153650296)

[Repeated K-Fold Cross Validation Technique 11](#_Toc153650297)

[Accuracy, Precision, and Recall of Test Data 12](#_Toc153650298)

[Matthew’s Correlation Coefficient 12](#_Toc153650299)

[Speed, Memory, and Repeat Execution Time Measures 13](#_Toc153650300)

[Results 14](#_Toc153650301)

[Zero-Fold Scores 14](#_Toc153650302)

[10-Fold Cross Validation Scores 14](#_Toc153650303)

[Accuracy, Precision, and Recall of Test Data 15](#_Toc153650304)

[Matthew’s Correlation Coefficient 16](#_Toc153650305)

[Execution Time and Memory Use 17](#_Toc153650306)

[Discussion 17](#_Toc153650307)

[Limitations of Feature Selection Techniques 18](#_Toc153650308)

[Ethical Considerations 19](#_Toc153650309)

[Conclusion 20](#_Toc153650310)

[Future Considerations 20](#_Toc153650311)

[Conclusion 20](#_Toc153650312)

[References 21](#_Toc153650313)

# Introduction

Many large US cities across the country have been experiencing rises in violent crimes (USAFacts, 2021). Coinciding with Covid-19 pandemic, homicide crime rates have increased by 25%, with an overall 3.3% increase in violent crimes in American cities (USAFacts, 2021). This demonstrates concerns for public safety and the need for preventative measures. This requires a need to determine what factors contribute to crime and what could possible solutions look like. Knowing these factors can help predict crime and allow communities leaders, law enforcement, and lawmakers make informed decisions through policy and programs to prevent or stop crime.

# Literature Review

## Related Studies

Studies looking at crime data have used feature selection processes to determine the most relevant features to include in their classification models (Kolomoytseva, 2021; Yerpude & Gudur, 2017). By running through these procedures, researchers would be able to estimate based on the models the most relevant features able to predict violent and non-violent crime (Kolomoytseva, 2021; Yerpude & Gudur, 2017). In Kolomoytseva (2021), Stepwise Selection and Least Absolute Shrinkage and Selection Operator (LASSO) were used to reduce the number of features included in the classification analysis. These methods’ performance was evaluated on the basis of mean square error (MSE) (Kolomoytseva, 2021). It was reported that the Stepwise Selection method did improve MSE measures, and the LASSO method saw further improvement (Kolomoytseva, 2021). Afterwards, classification analysis using Linear Discriminate Analysis (LDA), Logistic Regression, Classification Tree, and the three ensemble classifiers: Bagging, Random Forest, and Boosting was completed on the feature selected data subset (Kolomoytseva, 2021). The classifiers were evaluated based on test errors – the accuracy of the classification model (Kolomoytseva, 2021). The results showed that the most accurate model was the Random Forest classification model with a 4.33% test error rate (Kolomoytseva, 2021). Another study, Yerpude & Gudur (2017), used the same dataset to run classification models without applying a feature selection technique. These classification/regression models include decision trees, random forest, naïve bayes, and linear regression (Yerpude & Gudur, 2017). For each of the classification/regression models used in this study, an importance feature measure was used to select the top 10 features (Yerpude & Gudur, 2017). This importance features measures varied for each classifier used, for example, random forest used the Gini index (Yerpude & Gudur, 2017). Each model was evaluated based on cross validation, accuracy, precision, recall, F1 and MSE scores (Yerpude & Gudur, 2017). Similarly, in Kolomoytseva, the random forest classification model was the most balanced measure compared to the other 3 models in Yerpude & Gudur).

## Feature Selection

Feature Selection (FS) is the technique that can be used to remove features from datasets that are statistically uncorrelated with the dependent variables (Elssied et al., 2014; Parimala and Nallaswamy, 2011; Dhanya et al., 2020; Perangin-Angin & Bachtiar, 2020). This reduced set of features is then used in classification (Elssied et al., 2014; Parimala and Nallaswamy, 2011; Dhanya et al., 2020; Perangin-Angin & Bachtiar, 2020). This reduced dimensionality and computational complexity helps improve the efficiency and accuracy of classification of the dataset (Elssied et al., 2014; Kourou et al., 2014; Dhanya et al., 2020; Perangin-Angin & Bachtiar, 2020). Three types of techniques are used in feature selection: Filter, Wrapper, and Embedded methods.

## Filter Method

The filter methods or algorithms are used to select a subset of features before any classification processes are commenced (Elssied et al., 2014; Dhanya et al., 2020). The method uses statistical properties or probabilities of the different features in the subset to filter and remove the less relevant features (Elssied et al., 2014; Dhanya et al., 2020). This relevancy is based on comparative computational efficiency (Elssied et al., 2014; Dhanya et al., 2020). These models are typically faster than wrapper and ensemble methods but less accurate (Jovic et al., 2015; DeepaLakshmi &Velmurugan, 2016).

#### Kendall Tau Correlation Coefficient

Previous studies have used the Kendall Tau correlation coefficient to determine the select the most important features in predictive classification models (Van Hulse et al., 2012; Filus et al., 2021). The feature selection Kendall Tau correlation coefficient method can be used to reduce the high dimensionality or the number of attributes (predictors) before running a classification process (Van Hulse et al., 2012; Filus et al., 2021). The Kendall Tau correlation coefficient measures correspondence between two rankings (ranked features) the strength and direction of non-linear relationships between two variables (target variable being categorical) (Van Hulse et al., 2012; Filus et al., 2021). It provides an outcome measure between -1 and 1, whereby -1 indicates the strong disagreement and 1 indicates the strong agreement between two variables (Van Hulse et al., 2012; Filus et al., 2021). A value of 0 would indicate that there is no relationship between the two variables (Van Hulse et al., 2012; Filus et al., 2021).

Previously a study used a combination of various feature selection methods, including Spearman’s rank correlation coefficient (a similar method to Kendall Tau, but for output numerical data types) to a dataset about phishing websites in order to create a machine learning model that best detects which websites are phishing websites (Bhowmik et al., 2022). The process of the study first applied the filter methods that included Pearson correlation, Chi2, Information Gain, and Spearman Rank that would select the best features based on a determined threshold (Bhowmik et al., 2022). Afterwards, the 4 filter methods combined their selected features through a perturbation ensemble function which resulted in 89 features selected (Bhowmik et al., 2022). These features would then be feed into the wrapper method which would reduce the features to 51 (Bhowmik et al., 2022). A similar approach in this study, using the Kendall Tau feature selection method to reduce the dimensionality of the dataset.

### Wrapper Method

The wrapper method trains a model with a subset of features that only increase the model’s performance (Dhanya et al., 2020; Jovic et al., 2015). This is done by exploring every possible choice through this feature selection technique (Dhanya et al., 2020; Jovic et al., 2015). This usually makes the method computationally expensive, meaning more memory or time needed to complete the technique (Dhanya et al., 2020; Jovic et al., 2015). This means that wrapper methods tend to be slower than filter methods, however, wrapper methods obtain subsets that perform better than filter methods (Jovic et al., 2015).

#### Forward Selection

Forward selection begins by training a model with no features selected (Reif & Shafait, 2016). Afterwards, the technique will select the best feature to add (Reif & Shafait, 2016). This is repeated until a decided threshold is reached (Reif & Shafait, 2016). This technique has been used before for multiple datasets from the UCI repository to improve the performance the performance (e.g., accuracy) of a classifier (Mani & Kalpana, 2016).

A previous study looked to at 59 real-world datasets and used the SVM classification algorithm to predict the classification target feature (Reif & Shafait, 2016). To improve the accuracy and the runtime of the SVM algorithm, a forward selection technique with a set limit of feature evaluations was integrated (Reif & Shafait, 2016). The technique limited the features to only those with the highest predictive quality (Reif & Shafait, 2016). This reduced the computational requirements (Reif & Shafait, 2016). The results of the study found that this maintained the accuracy of the traditional forward selection method, but greatly improved the runtime (faster) (Reif & Shafait, 2016). Furthermore, when compared with filter and embedded methods, this approach was able to achieve higher accuracy (Reif & Shafait, 2016).

### Embedded Method

The Embedded methods use algorithms that are a hybrid of both the filter and wrapper methods (Jovic & Bogunovic, 2015; Dhanya et al., 2020; Fonti & Belister, 2017; Ghosh et al., 2020). These techniques have been reported to be faster than wrapper methods and more accurate than filter methods (Jovic & Bogunovic, 2015; Dhanya et al., 2020; Fonti & Belister, 2017; Ghosh et al., 2020).

#### Gradient Boosting

The Gradient Boosting method performs by creating multiple “weak” predictive models (decision trees) (Lamari et al., 2020, Xu et al., 2014, Kahn et al., 2022, Jung et al., 2023). The subsequent model tries to predict the error left by the previous model (Lamari et al., 2020, Xu et al., 2014, Kahn et al., 2022, Jung et al., 2023). Due to this sequential process, gradient boosting is usual slow but highly accurate. The collective building of these models will create one accurate predictor (Lamari et al., 2020, Xu et al., 2014, Kahn et al., 2022, Jung et al., 2023). This method can be adapted to select the top features that predict the target variable (Lamari et al., 2020, Xu et al., 2014, Kahn et al., 2022, Jung et al., 2023). Studies have shown that Gradient Boosting is highly accurate (e.g., higher accuracy, precision and recall performance) when compared to other models such as generalized linear models, neural networks and other ensemble models (Lamari et al., 2020, Xu et al., 2014, Kahn et al., 2022, Jung et al., 2023).

A previous study looked at three models to select the top features that predict violent and nonviolent crime in San Francisco, California (Kahn et al., 2022). The study compared Naïve Bayes, Random Forest and Gradient Boosting Decision Tree for their accuracy (Kahn et al., 2022). The Gradient Boosting technique achieved 98.5%, 96.96%, and 100% for accuracy, precision, and recall, respectively (Kahn et al., 2022). Whereas, Random Forest achieved, 63.43%, 62.80%, and 63.29% for accuracy, precision, and recall, respectively (Kahn et al., 2022). Lastly, Naïve Bayes achieved 63.33%, 63.88%, and 64.67% for accuracy, precision, and recall, respectively (Kahn et al., 2022). This study demonstrated that it was the most accurately performing model out of the three (Kahn et al., 2022).

## Communities and Crime Dataset

The dataset, “Communities and Crime”, was taken from the UC Irvine Machine Learning Repository (Redmond, 2009). The dataset combines socioeconomic and law enforcement data from the 1990 US Census and US LEMAS survey, respectively (Redmond, 2009). It also includes crime data from the 1995 FBI UCR (Redmond, 2009). The dataset consists of 1994 instances with 128 features (Redmond, 2009). Of the 128 features, 122 are predictive, 5 are non-predictor, and 1 is the target (Redmond, 2009). The obtained dataset came with all the numeric data normalized through an unsupervised, equal-interval binning method (Redmond, 2009). The numeric data is presented through decimal ranges 0.00 – 1.00. The features in the dataset retain their distributions and skewness (Redmond, 2009).

## Research Aims

The aim of the study will be to 1) determine which combination of 10 features are influential predictors that can classify five quantile levels of crime the best using the three feature selection techniques; 2) determine the effectiveness of the classifier model using subset data from the three feature selection techniques used; and 3) determine which of the three models used are stable. The link to GitHub repository is: <https://github.com/mbitel/CIND820_CommunitiesCrimeProject.git>

# Methodology

## Data Preprocessing

### Data Normalization

The public dataset provided from the UC Irvine Machine Learning Repository (Redmond, 2009) were already normalized/standardized. Meaning the values were between 0 and 1 for all numeric features, apart from the non-predictor features.

### Exploratory Data Analysis

An exploratory data analysis was conducted to review the different distributions and relationships (i.e., correlations) between the features and the target variable. The syntax used is shown below (YData Labs Inc, 2023):

*from ydata\_profiling import ProfileReport*

*EDA = ProfileReport(crime2, title = "Cleaned Crime Data")  
EDA.to\_file("CleanCrimeDataFull.html")*

### Removal of Features

The dataset contained a total of 128 features. Of the 128 features, one feature was the target feature, “ViolentCrimesPerPop”. The dataset was reviewed through a CSV file for missing and null values. Features that had numerous missing or null values were removed from the dataset. The features that were removed due to numerous missing/null values were (n=25): NumInShelters, NumStreet, LemasSwornFT, LemasSwFTPerPop, LemasSwFTFieldOps, LemasSwFTFieldPerPop, LemasTotalReq, LemasTotReqPerPop, PolicReqPerOffic, PolicPerPop, RacialMatchCommPol, PctPolicWhite, PctPolicBlack, PctPolicHisp, PctPolicAsian, PctPolicMinor, OfficAssgnDrugUnits, NumKindsDrugsSeiz, PolicAveOTWorked, PolicCars, PolicOperBudg, LemasPctPolicOnPatr, LemasGangUnitDeploy, LemasPctOfficDrugUn, PolicBudgPerPop. In addition, labelled non-predictive features were also removed. This included (n=5): “state”, “county”, “community”, “communityname”, “fold”. Furthermore, features that were measuring the same variable, but used a different type of measure (e.g., NumUnderPov vs PctUnderPov; count vs percentage). The features removed because of duplicate measures were (n=3): “numbUrban”, “NumUnderPov”, “NumIlleg”. The syntax used to upload the communities and crime dataset and remove features identified above is (NumFOCUS, Inc, 2023):

*import pandas as pd*

*crime = pd.read\_csv('communities.data', sep = ",", names= ['state', 'county', 'community', 'communityname', 'fold', 'population', 'householdsize', 'racepctblack', 'racePctWhite', 'racePctAsian', 'racePctHisp', 'agePct12t21', 'agePct12t29', 'agePct16t24', 'agePct65up', 'numbUrban', 'pctUrban', 'medIncome', 'pctWWage', 'pctWFarmSelf','pctWInvInc', 'pctWSocSec', 'pctWPubAsst', 'pctWRetire', 'medFamInc', 'perCapInc', 'whitePerCap', 'blackPerCap', 'indianPerCap', 'AsianPerCap','OtherPerCap', 'HispPerCap', 'NumUnderPov', 'PctPopUnderPov', 'PctLess9thGrade', 'PctNotHSGrad', 'PctBSorMore', 'PctUnemployed', 'PctEmploy', 'PctEmplManu', 'PctEmplProfServ', 'PctOccupManu', 'PctOccupMgmtProf', 'MalePctDivorce', 'MalePctNevMarr', 'FemalePctDiv', 'TotalPctDiv', 'PersPerFam', 'PctFam2Par', 'PctKids2Par', 'PctYoungKids2Par', 'PctTeen2Par', 'PctWorkMomYoungKids', 'PctWorkMom', 'NumIlleg', 'PctIlleg', 'NumImmig', 'PctImmigRecent', 'PctImmigRec5', 'PctImmigRec8', 'PctImmigRec10', 'PctRecentImmig', 'PctRecImmig5', 'PctRecImmig8', 'PctRecImmig10', 'PctSpeakEnglOnly', 'PctNotSpeakEnglWell', 'PctLargHouseFam', 'PctLargHouseOccup', 'PersPerOccupHous', 'PersPerOwnOccHous', 'PersPerRentOccHous', 'PctPersOwnOccup', 'PctPersDenseHous', 'PctHousLess3BR', 'MedNumBR', 'HousVacant', 'PctHousOccup', 'PctHousOwnOcc', 'PctVacantBoarded', 'PctVacMore6Mos', 'MedYrHousBuilt', 'PctHousNoPhone', 'PctWOFullPlumb', 'OwnOccLowQuart', 'OwnOccMedVal', 'OwnOccHiQuart', 'RentLowQ', 'RentMedian', 'RentHighQ', 'MedRent', 'MedRentPctHousInc', 'MedOwnCostPctInc', 'MedOwnCostPctIncNoMtg', 'NumInShelters', 'NumStreet', 'PctForeignBorn', 'PctBornSameState', 'PctSameHouse85', 'PctSameCity85', 'PctSameState85', 'LemasSwornFT', 'LemasSwFTPerPop', 'LemasSwFTFieldOps', 'LemasSwFTFieldPerPop', 'LemasTotalReq', 'LemasTotReqPerPop', 'PolicReqPerOffic', 'PolicPerPop', 'RacialMatchCommPol', 'PctPolicWhite', 'PctPolicBlack', 'PctPolicHisp', 'PctPolicAsian', 'PctPolicMinor', 'OfficAssgnDrugUnits', 'NumKindsDrugsSeiz', 'PolicAveOTWorked', 'LandArea', 'PopDens', 'PctUsePubTrans', 'PolicCars', 'PolicOperBudg', 'LemasPctPolicOnPatr','LemasGangUnitDeploy','LemasPctOfficDrugUn', 'PolicBudgPerPop', 'ViolentCrimesPerPop'])  
crime2 = crime.drop(crime.columns[[0, 1, 2, 3, 4, 15, 32, 54, 94, 95, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 121, 122, 123, 124, 125, 126]], axis = 1)  
print(crime2)*

### Normality Test

A Shapiro – Wilk test was performed to determine the normality distribution of the dataset (Gonzalez-Estrada & Cosmes, 2019). The results of these test determined what feature selection test would be most suitable to run against the dataset. Initially, a regression analysis was chosen to develop the model that predicts Violent crime because both the predictive variables and the target variable were continuous. After completing a normality test, a large number of the features were not normally distributed (right and left skewed distributions). This violates one of the assumptions for linear regression analysis. Meaning that using linear models could produce inaccurate results. The following syntax was used to apply the Shapiro – Wilk test on all columns in the dataset. The pandas syntax was used to run the Shapiro – Wilk test, *scipy.stats.shapiro(x)* (SciPy community, 2023, NumFOCUS, Inc, 2023):

*Import pandas as pd  
From scipy import stats  
Import os  
crime = pd.read\_csv('cleanedcommunitiescrime.csv', sep= ',')  
crime2 = crime.apply(stats.shapiro)  
crime2.to\_csv('normalitytests.csv')*

### Changing Target Variable to Categorical Variable

In order to use classification models the target variable had to be changed into an ordinal categorical variable. Using the pandas qcut() function to sort the continuous values into quantile bins which include categories: “Very Low Crime”, “Low Crime", "Medium Crime", "High Crime", “Very High Crime". The following syntax was used (NumFOCUS, Inc, 2023):

*Import pandas as pd*

*x = crime.drop(crime.columns[[0, 95]], axis = 1)  
y = crime['ViolentCrimesPerPop']  
y = pd.qcut(y, q=5, labels=['Very Low Crime', 'Low Crime','Medium Crime', 'High Crime', 'Very High Crime'])*

### Splitting Data

The data was split to stratified training and testing data 70:30 ratio. The stratification was included to ensure that the proportional categories of the target variable is included in both training and testing datasets. This is a common split used to train and test models. The training was 0.7 and the test was 0.3. The following syntax was used (Pedregosa et al., 2011):

*from sklearn.model\_selection import train\_test\_split*

*x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, stratify = y, test\_size= 0.3, train\_size= 0.7 random\_state=345 shuffle=True)*

### Missing/Null Values

For features that contained some missing/null values, they were replaced by column median scores. The following syntax was used (NumFOCUS, Inc, 2023):

*Import pandas as pd*

*x\_train = x\_train.fillna(x\_train.median(), method=None, axis=None, inplace=False, limit=None, downcast=\_NoDefault.no\_default)*

## Feature Selection Techniques

Three feature selection techniques will be applied to the dataset to select the top 10 features. The feature number limit was selected based on a previous study, Yerpude & Gudur (2017). The three feature selection techniques will be the filter method Kendall Tau correlation coefficient, wrapper method forward selection, and the embedded method Gradient Boosting. Each of these feature selection techniques will be evaluated on the accuracy, precision, recall, Matthew’s correlation coefficient and computational complexity which includes duration for analysis and the memory capacity used.

1) Filter Method Kendall Tau Correlation Coefficient: The filter method used is the Kendall Tau correlation coefficient. Due to the non-normal distribution of the data and the numeric continuous values the Kendall Tau correlation coefficient was chosen to select the 10 best variables. The syntax used (SciPy community, 2023):

*from scipy.stats import kendalltau  
import operator  
kendall\_list = []  
for col in x\_train.columns:  
 tau, p = kendalltau(x\_train[col], y\_train, initial\_lexsort=None, nan\_policy='propagate', method='auto', variant='b', alternative='two-sided')  
 kendall\_list.append((col, tau))  
selection = sorted(kendall\_list, key=lambda x\_train: abs(x\_train[1]), reverse=True)  
kendall\_list = selection[:10]  
selected\_features = list(map(operator.itemgetter(0), kendall\_list))  
newx\_train = x\_train[selected\_features]  
print("Top 10 Selected Features:", selected\_features)*

Top Ten Features: 'PctPersDenseHous', 'pctWInvInc', 'FemalePctDiv', 'TotalPctDiv', 'MalePctDivorce', 'PctPersOwnOccup', 'PctKids2Par', 'racePctWhite', 'PctHousNoPhone', 'PctIlleg'

Link: <https://github.com/mbitel/CIND820_CommunitiesCrimeProject/blob/093e9acbf7f28e1ead564585e3fbbc3e170918e6/Filter_FS_Classifier.ipynb>

2) Wrapper Method Forward Selection: The wrapper method used was forward selection. This technique was used to select the 10 best features. The syntax used (Pedregosa et al., 2011):

*from sklearn.feature\_selection import SequentialFeatureSelector  
from sklearn.svm import SVC  
selection = SequentialFeatureSelector(SVC(\*, C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache\_size=200, class\_weight=None, verbose=False, max\_iter=1, decision\_function\_shape='ovr', break\_ties=False, random\_state=None), direction='forward', n\_features\_to\_select=10)  
selection.fit(x\_train, y\_train)  
newx\_train = selection.transform(x\_train)  
selected\_features = x\_train.columns[selection.get\_support()].tolist()  
print("Top 10 Selected Features:", selected\_features)*

Top 10 Features: 'population', 'racePctWhite', 'medIncome', 'perCapInc', 'whitePerCap', 'TotalPctDiv', 'PctKids2Par', 'NumImmig', 'HousVacant', 'MedRentPctHousInc'

Link: <https://github.com/mbitel/CIND820_CommunitiesCrimeProject/blob/093e9acbf7f28e1ead564585e3fbbc3e170918e6/Wrapper_FS_Classifier.ipynb>

3) Embedded Method Gradient Boosting: The embedded method used was Gradient Boosting. This technique was used to select the 10 best features. The syntax used to select the top 10 features (Pedregosa et al., 2011):

*from sklearn.feature\_selection import SelectFromModel  
from sklearn.ensemble import GradientBoostingClassifier  
gb = GradientBoostingClassifier(\*, loss='log\_loss', learning\_rate=0.1, n\_estimators=100, subsample=1.0, criterion='friedman\_mse', min\_samples\_split=2, min\_samples\_leaf=1, min\_weight\_fraction\_leaf=0.0, max\_depth=3, min\_impurity\_decrease=0.0, init=None, random\_state=None, max\_features=None, verbose=0, max\_leaf\_nodes=None, warm\_start=False, validation\_fraction=0.1, n\_iter\_no\_change=None, tol=0.0001, ccp\_alpha=0.0)  
gb.fit(x\_train, y\_train)  
selection = SelectFromModel(gb, max\_features=10, threshold=None, prefit=False, norm\_order=1, importance\_getter='auto' )  
selected\_features = x\_train.columns[selection.get\_support()].tolist()  
newx\_train = x\_train[selected\_features]  
print("Top 10 Selected Features:", selected\_features)*

Top 10 Features: 'racepctblack', 'racePctWhite', 'pctWInvInc', 'FemalePctDiv', 'PctFam2Par', 'PctKids2Par', 'PctWorkMomYoungKids', 'PctIlleg', 'PctPersDenseHous', 'HousVacant'

Link: <https://github.com/mbitel/CIND820_CommunitiesCrimeProject/blob/093e9acbf7f28e1ead564585e3fbbc3e170918e6/Embedded_FS_ClassifierV2.ipynb>

## Performance Evaluation

### Accuracy Score of Zero-Folds

A measure of zero-folds for accuracy was conducted to in order to use the accuracy score to determine the variance to compare with the variance obtained from the 10-Fold Cross Validation method. The following syntax was used.

*from sklearn.svm import SVC  
from sklearn.metrics import classification\_report  
model = SVC(\*, C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache\_size=200, class\_weight=None, verbose=False, max\_iter=1, decision\_function\_shape='ovr', break\_ties=False, random\_state=None)  
model.fit(newx\_train, y\_train)  
print("zero-fold scores:", classification\_report(y\_train, model.predict(newx\_train)))*

The accuracy score was then taken and used in the following to determine the variance: variance = accuracy \* (1 – accuracy).

### Repeated K-Fold Cross Validation Technique

A repeated 10-fold cross validation technique was used to evaluate each feature selection model. The technique trained the model against 9 random subsets of the dataset and tested the model against the 10 subsets. In addition, this was repeated a total of 3 times. This repeat allows to improve the estimates of the mean model performance at the cost of fitting and evaluating many more models. The average accuracy and variance score was determined. The model used is the Support Vector Classifier (SVC) because the target variable is a categorical ordinal variable (ViolentCrimesPerPop: total number of violent crimes per 100K population). Furthermore, the SVC is effective with high dimensional data and quite versatile (Pedregosa et al., 2011).

*from sklearn.model\_selection import cross\_val\_score  
from sklearn.svm import SVC  
cv = RepeatedKFold(n\_splits=10, n\_repeats=3, random\_state=1)  
model = SVC(\*, C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache\_size=200, class\_weight=None, verbose=False, max\_iter=1, decision\_function\_shape='ovr', break\_ties=False, random\_state=None)  
score = cross\_val\_score(model, newx\_train, y\_train, scoring= 'accuracy', cv=cv, n\_jobs=-1, groups=None, verbose=0, fit\_params=None, pre\_dispatch='2\*n\_jobs', error\_score=nan)  
mean\_score = sum(score)/30  
print("R10-fold mean accuracy score:", mean\_score)  
print("R10-Fold accuracy variance:", mean\_score\*(1-mean\_score))*

### Accuracy, Precision, and Recall of Test Data

Accuracy is measured by considering the total number of correct predictions over the total number of predictions made (Juba & Le, 2019; Saunders & Freitas, 2022; Halibas et al., 2018). Precision is measured by considering the number of correct positive predictions made over the total number positive predictions (both correct and incorrect) (Juba & Le, 2019; Saunders & Freitas, 2022; Halibas et al., 2018). Lastly, recall is measured by considering the number of correct positive predictions made over the total number of correct predictions (Juba & Le, 2019; Saunders & Freitas, 2022; Halibas et al., 2018). These three measures are commonly used to assess the performance of the modeling applied to the dataset (Yerpude & Gudur, 2017). These measures have been previously used to assess the performance of classifiers for the same UC Irine Machine Learning dataset under consideration for this study (Yerpude & Gudur, 2017). To evaluate the performance of the model, accuracy, precision and recall were assessed. The syntax used was the sklearn.metrics.classification\_report (Pedregosa et al., 2011):

*from sklearn.metrics import classification\_report*

*newx\_test = selection.transform(x\_test)  
print("Test data performance report:", classification\_report(y\_test, model.predict(newx\_test), labels=None, target\_names=None, sample\_weight=None, digits=2, output\_dict=False, zero\_division='warn'))*

### Matthew’s Correlation Coefficient

Matthew’s correlation coefficient is a statistical tool used to evaluate the performance of a model (Chicco & Jurman, 2020, Yao and Sheppard, 2020). It considers the differences between the predicted values and the actual values (Chicco & Jurman, 2020, Yao and Sheppard, 2020). Due to the limitations of accuracy, precision and recall measures which only consider the positive values in analysis, the MCC considers all 4 entities (i.e., True positives, True negatives, False positives, False negatives) (Chicco & Jurman, 2020, Yao and Sheppard, 2020).The formula for MCC is**:** (True Positives\*True Negatives – False Positives\*False Negatives) / √(True Positives+False Positives)(True Positives+False Negatives)(True Negatives+False Positives)(True Negatives+False Negatives) (Chicco & Jurman, 2020, Yao and Sheppard, 2020). The ranges of MCC scores are -1 to +1 (Chicco & Jurman, 2020, Yao and Sheppard, 2020). A -1 score indicates that there is total disagreement between the predicted values and the actual values (Chicco & Jurman, 2020, Yao and Sheppard, 2020). A score of zero indicates that there is no difference between the predicted values and the actual values is no better than random (Chicco & Jurman, 2020, Yao and Sheppard, 2020). A +1 score indicates that there is total agreement between the predicted values and the actual values (Chicco & Jurman, 2020, Yao and Sheppard, 2020). The syntax used was sklearn.metrics.matthew\_corrcoef (Pedregosa et al., 2011):

*from sklearn.metrics import matthews\_corrcoef*

*newx\_test = selection.transform(x\_test)  
y\_pred = model.predict(newx\_test)  
mcc = matthews\_corrcoef(y\_test, y\_pred, sample\_weight=None)  
print("MCC score:", mcc)*

### Speed, Memory, and Repeat Execution Time Measures

To further evaluate the performance of the three different models, speed (or duration), memory and the stability of the models were assessed. Computational time can be measured to determine the duration required to run the method for analysis (Zebra et al., 2020; Verma et al., 2018). This will determine how efficient and feasible is the application of the selected method on the dataset (Zebra et al., 2020; Verma et al., 2018). By running feature selection techniques to reduce the dimensionality of the dataset, this could help improve the accuracy and efficiency of the data mining computation (Zebra et al., 2020; Verma et al., 2018). This result is reflected by reduced time and amount of memory required for the computation of the dataset (Zebra et al., 2020; Verma et al., 2018).

Speed was measured in seconds. The purpose of this measure the duration or how long it took to execute the code. This was the syntax used:

*Import time*

*start\_time = time.time()*

\*\*\*Beginning of code to the end of the model fit code (see algorithm for more details)\*\*\*

*end\_time = time.time()*

*print("Duration of execution (sec):", end\_time - start\_time)*

Memory was measured in kibibytes. The purpose of this measure was to assess the amount of memory used to execute the code. The top ten lines of code that used the most memory was selected and evaluated for performance. This was the syntax used:

*import tracemalloc*

*tracemalloc.start()*

\*\*\*Beginning of code to the end of the model fit code (see algorithm for more details)\*\*\*

*tracemalloc.stop()  
memory = snapshot.statistics('lineno')  
for stat in memory[:10]:  
 print("Memory Used:", stat)*

# Results

## Zero-Fold Scores

In this table, the accuracy scores are presented from the zero-fold classification report, together with the calculated variance score for each feature selection technique used for the SVC model.

|  |  |  |
| --- | --- | --- |
| Feature Selection Technique | Accuracy Score | Variance Score (accuracy \*(1 – accuracy) |
| Kendall Tau Correlation Coefficient | 0.55 | 0.25 |
| Forward Selection | 0.56 | 0.25 |
| Gradient Boosting | 0.54 | 0.25 |

## 10-Fold Cross Validation Scores

In this table, the average accuracy scores are presented from the 10-fold cross validation evaluation, together with the calculated average variance score for each feature selection technique used from the SVC model.

|  |  |  |
| --- | --- | --- |
| Feature Selection Technique | Accuracy Score | Variance Score (accuracy \*(1 – accuracy) |
| Kendall Tau Correlation Coefficient | 0.52 | 0.25 |
| Forward Selection | 0.53 | 0.25 |
| Gradient Boosting | 0.51 | 0.25 |

Based on training data accuracy scores completed with zero-folds and 10-folds (with 3 repeats), the accuracy scores were the highest for Forward Selection (0.56 and 0.53, respectively) and lowest for Gradient Boosting (0.54 and 0.51, respectively). Kendall Tau had the second best accuracy score of the training data (0.55 and 0.52, respectively). The variance was not different for any technique when comparing zero-fold to 10-folds (variance = 0.25). This shows that all three models remained stable. However, the accuracy score of all three models were below 60% indicating that they had poor accuracy in correctly classifying data instances over the total number of data observations.

## Accuracy, Precision, and Recall of Test Data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kendall Tau Correlation Coefficient | Precision | Recall | Support | Accuracy |
| Very High Crime | 0.69 | 0.63 | 116 | 0.49 |
| High Crime | 0.40 | 0.39 | 116 |  |
| Medium Crime | 0.36 | 0.38 | 127 |  |
| Low Crime | 0.31 | 0.11 | 95 |  |
| Very Low Crime | 0.55 | 0.81 | 145 |  |

Using the Kendall Tau correlation coefficient feature selection model “very high crime” and “very low crime” categories have shown to have the highest precision and recall scores compared to the other 3 categories. On the other hand, “low crime” had the lowest precision and recall scores, specifically the recall was very low (0.11). This could mean that there were a lot of false negatives detected for this category which would suggest that the model didn’t predict the low crime as accurately compared to other categories. Furthermore, the accuracy score (0.49) was lower than the accuracy score from the training data (0.55). This suggests that there was some overfitting because the model was well trained towards the data in the training dataset.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Forward Selection | Precision | Recall | Support | Accuracy |
| Very High Crime | 0.66 | 0.59 | 116 | 0.48 |
| High Crime | 0.39 | 0.34 | 116 |  |
| Medium Crime | 0.35 | 0.43 | 127 |  |
| Low Crime | 0.35 | 0.07 | 95 |  |
| Very Low Crime | 0.55 | 0.82 | 145 |  |

Using the forward selection feature selection model “very high crime” and “very low crime” categories have shown to have the highest precision and recall scores compared to the other 3 categories. On the other hand, “low crime” had the lowest precision and recall scores, specifically the recall was very low (0.07). This could mean that there were a lot of false negatives detected for this category which would suggest that the model didn’t predict the low crime as accurately compared to other categories. Furthermore, the accuracy score (0.48) was lower than the accuracy score from the training data (0.56). This suggests that there was some overfitting because the model was well trained towards the data in the training dataset.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Gradient Boosting | Precision | Recall | Support | Accuracy |
| Very High Crime | 0.72 | 0.66 | 116 | 0.53 |
| High Crime | 0.45 | 0.47 | 116 |  |
| Medium Crime | 0.38 | 0.42 | 127 |  |
| Low Crime | 0.44 | 0.08 | 95 |  |
| Very Low Crime | 0.59 | 0.86 | 145 |  |

Using the gradient boosting feature selection model “very high crime” and “very low crime” categories have shown to have the highest precision and recall scores compared to the other 3 categories. On the other hand, “low crime” had the lowest precision and recall scores, specifically the recall was very low (0.08). This could mean that there were a lot of false negatives detected for this category which would suggest that the model didn’t predict the low crime as accurately compared to other categories. Furthermore, the accuracy score (0.53) was similar to the accuracy score from the training data (0.54). This suggests that there was no overfitting.

Results here show that the Kendall Tau correlation coefficient and forward selection precision and recall scores are for the most part lower compared to gradient boosting. Forward selection accuracy score (0.48) is the lowest of the three methods. On the other hand, the accuracy score of gradient boosting (0.53) is higher compared to Kendall Tau correlation coefficient and forward selection methods. However, just like the accuracy scores from the training dataset, all three models had an accuracy score below 60% in the testing dataset, indicating that they had poor accuracy in correctly classifying violent crime over the total number of data observations.

## Matthew’s Correlation Coefficient

|  |  |
| --- | --- |
| Feature Selection Technique | Matthew’s Correlation Coefficient |
| Kendall Tau Correlation Coefficient | 0.36 |
| Forward Selection | 0.35 |
| Gradient Boosting | 0.41 |

Based on the Matthew’s correlation coefficient scores, the gradient boosting model had the best score (0.41). This means that the gradient boosting feature selection model is the best model out of the three that considers the classification of all elements, including negative elements such as true negatives and false negatives. However, the actual scores are below 60% which would indicate that the models are not very accurate at predicting the violent crime.

## Execution Time and Memory Use

|  |  |  |
| --- | --- | --- |
| Feature Selection Technique | Execution Time (seconds) | Memory Use (KiB) |
| Kendall Tau Correlation Coefficient | 12.89 | 49,526 |
| Forward Selection | 266.78 | 49,481 |
| Gradient Boosting | 28.18 | 50,899 |

Based on these performance measures, Kendall Tau correlation coefficient was the fastest code (12.89 seconds) in a single run and used less memory (49,526 KiB) compared to the gradient boosting method (28.18 seconds and 50, 899 KiB). The Forward selection model was the slowest out of the 3 models (266.78 seconds). Gradient boosting was the second fastest model but used the most amount of memory to complete the execution. Considering the accuracy, mcc and speed scores, the Kendall Tau correlation coefficient model is fast but won’t produce as accurate results at the gradient boosting model. The gradient boosting model, even though a bit slower, does provide more accurate results. Therefore, the gradient boosting model might be the preferred model since it provides a better balance in performance compared to the filter and wrapper methods.

# Discussion

The use of the 3 different feature selection techniques been used to select the 10 best features which were used in the Support Vector Classifier to predict violent crime rates against five categories, “Very High Crime”, “High Crime”, “Medium Crime”, “Low Crime”, and “Very Low Crime”. The 3 feature selection techniques include a filter method Kendall Tau correlation coefficient, wrapper method forward selection, and an embedded method gradient boosting. Based on the literature it would have been assumed that the most accurate of the 3 methods would have been the wrapper method forward selection, followed by the embedded method gradient boosting and filter method Kendall Tau (Elssied et al., 2014; Jovic et al., 2015). Furthermore, the fastest of these 3 methods would have been filter method, followed by embedded and then wrapper (Elssied et al., 2014; Jovic et al., 2015). Based on previous studies (Elssied et al., 2014; Jovic et al., 2015), the computational performance observed in the current study agrees with past literature as the computational performance of each method was seen in this order: 1) Kendall Tau (12.89 sec), gradient boosting (28.18 sec), and forward selection (266.78 sec). However, when looking at the accuracy scores, the observations in the current study did not agree with previous literature (Elssied et al., 2014; Jovic et al., 2015). The accuracy scores were ordered from best to worst: 1) gradient boosting, 2) Kendall Tau, and 3) forward selection.

Some of the reasons for this discrepancy are that the forward selection technique did overfit with the training data as it had the highest accuracy scores compared to the other two feature selection techniques (Loughrey & Cunningham, 2005). This overfitting could have resulted in the reduction in effectively predicting violent crime categories based on the features it has selected (Loughrey & Cunningham, 2005).

Moreover, the variance scores, although were low (0.25) across all feature selection techniques, the variance remained stable across all models after completing a 10-fold cross validation assessment. This was a method used to check for the stability of the models. If the models were consistent in their predictions of the target variable the variance should have been the same when comparing no folds to the 10-fold performance evaluation. Otherwise, if the variance was different from no folds to 10-fold performance, then there might be issues with the stability of the model, especially if it requires to do numerous repeats of raw data. Such lack of stability could result in poor accuracy of real-world data (not training data) (Khaire & Dhanalakshmi, 2022).

Considering the features that have been selected by the 3 models, the embedded method, which was the most accurate model compared to the other two models, showed that 'racepctblack', 'racePctWhite', 'pctWInvInc', 'FemalePctDiv', 'PctFam2Par', 'PctKids2Par', 'PctWorkMomYoungKids', 'PctIlleg', 'PctPersDenseHous', 'HousVacant' were the top 10 variables that would predict the different categories of violent crime. Furthermore, there was some overlap between the 3 models and their selection of features which included ‘PctKids2Par’, ‘racePctWhite’, ‘PctPersDenseHous’, ‘PctIlleg’, ‘FemalePctDiv’ and ‘pctWInvInc’, This might indicate that these features might be important for consideration and warrant further study to assess what implications they might have in their relationship to violent crime.

This was similar to Yerpude (2020) who used the same UC communities and crime data to find top 10 features using classification and regression models. Common features included: ‘PctKids2Par’, ‘racePctWhite’, ‘PctIlleg’, ‘PctFam2Par’, ‘racePctHisp’, ‘FemalePctDiv’, and ‘PctNotSpeakEnglWell’. Some of these features do overlap with what was observed from this current study’s feature selection techniques. However, the models used in Yerpude (2020) were shown to be more accurate in their scores. One reason could that the target variable violent crime was divided into binomial categories “high crime” and “not high crime”. This could have made the classification easier to predict and score highly on the accuracy, precision, recall and F1 scores.

## Limitations of Feature Selection Techniques

Limitations to this study include no removal of highly correlated data with other features. The highly correlated features could reduce information gain if selected by the feature selection technique, making the selected features redundant and the overall model less accurate (Reunanen, 2003). This would inevitably mean that performance metrics such as accuracy, Matthew’s correlation coefficient, precision, and recall would score low.

Furthermore, limiting to only 10 features out of 94 features might not produce the most effective model. Meaning more features might be required to adequately predict violent crime, especially if there is an interaction between different features.

In addition, there could have been multiple features that were irrelevant in the dataset to predict crime. Such understanding of variables might require expertise to review and remove irrelevant features. During the development of this model, there was no expert in the field of crime and communities to determine what features were irrelevant. The only feature removal process that took place was the removal of features that were not predictors as labelled by the owners of the dataset, and removal of features that were duplications and had numerous missing/null values. The use of irrelevant features could have also led to low performance measures as seen in this study.

Some limitations with the filter feature selection techniques include low precisions and prone to selecting inadequate features. This means that filter methods like the Kendall Tau consider each feature individually without taking any interactions between features into account. This could result in selecting redundant or correlated features that don’t add information to the model. This could also result in missing important interactions between features that could affect the performance of the model (Biwas et al., 2016; Pudjihartono et al., 2022).

Furthermore, the sequential forward selection wrapper methods on the other hand have a nesting drawback. This means as forward selection adds the best features to its list of features, during this process it can not replace or exclude the feature later, even if it might be possible to increase the evaluation score (Reunanen, 2003). The could result in a lower performance score.

## Ethical Considerations

An ethical consideration would be to be careful not to create stereotypes from the predictive variables provided by the feature selection techniques. For example, if you are from a racial group that predicts high crime, making societal assumptions based on race could have considerable moral implications such as racism. Therefore, such data must be interpreted with care and avoid sharing the wrong message. Furthermore, the number of factors included in this dataset might not be exhaustive, therefore, assuming these factors predict crime could wrongfully marginalize groups of people based on just 10 factors. Moreover, due to the use of only 10 factors to predict crime, this might not actually provide adequate information to make informed decisions. This could then open doors to more blind spots in how this information is applied in the real world. Furthermore, using such information to take proactive approaches might result in assigning a pre-destined outcome to someone who contains these features despite not committing a crime. Without considering the moral implications of such data and analysis, actions taken based on these models might be unethical.

# Conclusion

## Future Considerations

Future tests of these algorithms on the communities and crime dataset provide by the UC Irvine Repository should consider reviewing the variables for their interactions to understand the combination of variables that might actually provide more accurate results than only single features alone (Tang et al., 2019). Furthermore, during the data preprocessing phase, one should consider removing the highly correlated features to reduce redundancy of the data. Future steps should look at selecting more features than 10 and compare if the accuracy of the models improves or diminishes. This might show how complex violent crime predict is and how multiple factors can contribute to the outcome. In addition, it might also be beneficial to use more recent data and determine if the models used to predict data from 1995 crime data is relevant to predicting violent crime in more recent datasets. This might determine if the factors selected from the older dataset are relevant to today’s context.

## Conclusion

The use of three feature selection techniques provided insights into selecting features from the UC Irvine Repository communities and crime dataset. The features selected did find overlap among the three models and with previous literature (Yerpude, 2020). Furthermore, the most effective model was the model that used the gradient boosting feature selection technique to select the best 10 features to predict the different categories of violent crime rates. Although, the performance measures of the models were poor, the insights gained from the study showed the comparisons between the three different models and how they performed.

# References

Biswas, S., Bordoloi, M., & Purkayastha, B. (2016). Review on Feature Selection and Classification using Neuro-Fuzzy Approaches. International Journal of Applied Evolutionary Computation (IJAEC), 7(4), 28-44. <http://doi.org/10.4018/IJAEC.2016100102>

Bhowmik, P., Sohrawordi, M., Ali, U.A.M.E., Bhowmik, P.C. (2022). An Empirical Feature Selection Approach for Phishing Websites Prediction with Machine Learning. In: Islam, A.K.M.M., Uddin, J., Mansoor, N., Rahman, S., Al Masud, S.M.R. (eds) Bangabandhu and Digital Bangladesh. ICBBDB 2021. Communications in Computer and Information Science, 1550, 173-188. <https://doi.org/10.1007/978-3-031-17181-9_14>

Chicco, D., Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 6. <https://doi.org/10.1186/s12864-019-6413-7>

DeepaLakshmi, S., & Velmurugan, T. (2016). Empirical Study of Feature Selection Methods for High Dimensional Data. Indian Journal of Science and Technology, 9(39), 1-6. <https://dx.doi.org/10.17485/ijst/2016/v9i39/90599>

Dhanya, R., Paul, I.R., Akula, S.S., Sivakumar, M., & Nair, J.J. (2020). F-test feature selection in Stacking ensemble model for breast cancer prediction. Procedia Computer Science, 171, 1561-1570.

Elssied, N.O., Ibrahim, O., & Osman, A.H. (2014). A Novel Feature Selection Based on One-Way ANOVA F-Test for E-Mail Spam Classification. Research Journal of Applied Sciences, Engineering and Technology, 7, 625-638. <https://doi.org/10.19026/RJASET.7.299>

Elizabeth González-Estrada, E., & Cosmes, W. (2019). Shapiro–Wilk test for skew normal distributions based on data transformations. Journal of Statistical Computation and Simulation, 89(17): 3258-3272. <https://doi.org/10.1080/00949655.2019.1658763>

Filus, K., Boryszko, P., Domańska, J., Siavvas, M., Gelenbe, E. (2021). Efficient Feature Selection for Static Analysis Vulnerability Prediction. Sensors, 21, 1133. https://doi.org/10.3390/s21041133

Halibas, A. S., Reazol, L. B., Delvo, E. G. T., & Tibudan, J. C. (2018). Performance Analysis of Machine Learning Classifiers for ASD Screening. 2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), 1-6. <http://dx.doi.org/10.1109/3ICT.2018.8855759>

Saunders, J. D., & Freitas, A. A. (2022). Evaluating the Predictive Performance of Positive- Unlabelled Classifiers: a brief critical review and practical recommendations for improvement. SIGKDD Explor Newsl, 24(2): 5–11. <https://doi.org/10.1145/3575637.3575642>

Jović, A., Brkić, K., & Bogunovic, N. (2015). A review of feature selection methods with applications. 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 1200-1205. <https://doi.org/10.1109/MIPRO.2015.7160458>

Juba, B., & Le, H. S. (2019). Precision-Recall versus Accuracy and the Role of Large Data Sets. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(01), 4039-4048. <https://doi.org/10.1609/aaai.v33i01.33014039>

Jung, S. G., Jung, G., & Cole, J. M. (2023). Gradient boosted and statistical feature selection workflow for materials property predictions. J. Chem. Phys, 159 (19). <https://doi.org/10.1063/5.0171540>

Khan, M., Ali, A., & Alharbi, Y. (2022), "Predicting and Preventing Crime: A Crime Prediction Model Using San Francisco Crime Data by Classification Techniques", *Complexity*, 2022. <https://doi.org/10.1155/2022/4830411>

Kolomoytseva, Angelina. (2021). Statistical Analysis of the Communities and Crime Data Set.

Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., & Fotiadis, D. I. (2014). Machine learning applications in cancer prognosis and prediction. Computational and structural biotechnology journal, 13, 8–17. <https://doi.org/10.1016/j.csbj.2014.11.005>

Lamari, Y., Freskura, B., Abdessamad, A., Eichberg, S., de Bonviller, S. (2020. Predicting Spatial Crime Occurrences through an Efficient Ensemble-Learning Model. *ISPRS International Journal of Geo-Information*. 9(11). <https://doi.org/10.3390/ijgi9110645>

Loughrey, J., Cunningham, P. (2005). Overfitting in Wrapper-Based Feature Subset Selection: The Harder You Try the Worse it Gets. In: Bramer, M., Coenen, F., Allen, T. (eds) Research and Development in Intelligent Systems XXI. <https://doi.org/10.1007/1-84628-102-4_3>

Mani, K., & Kalpana, P. (2016). An Efficient Feature Selection based on Bayes Theorem, Self Information and Sequential Forward Selection. International Journal of Information Engineering and Electronic Business, 8, 46-54. <https://doi.org/10.5815/IJIEEB.2016.06.06>

NumFOCUS, Inc. (2023). Panda (Version 2.1.1). Sphinx. <https://pandas.pydata.org/docs/>

NumPy Developers (2022). NumPy (Version 1.26). Sphinx. <https://numpy.org/doc/stable/>

Parimala, R., & Nallaswamy, R. (2011). A Study of Spam E-mail classification using Feature Selection package. Global journal of computer science and technology, 11.

Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python (Version 1.3.1). Journal of Machine Learning Research, 12: 2825-2830. <https://scikit-learn.org/stable/user_guide.html>

Perangin-Angin, D., & Bachtiar, F. (2021). Classification of Stress in Office Work Activities Using Extreme Learning Machine Algorithm and One-way ANOVA F-Test Feature Selection. 4th International Seminar on Research of Information Technology and Intelligent Systems, 503-508. <http://dx.doi.org/10.1109/ISRITI54043.2021.9702802>

Pudjihartono, N., Fadason, T., Kempa-Liehr, A. W., O’Sullivan, J. M. (2022). A Review of Feature Selection Methods for Machine Learning-Based Disease Risk Prediction. Frontiers in Bioinformatics, 2. <https://doi.org/10.3389/fbinf.2022.927312>

Redmond, M. (2009). Communities and Crime. UCI Machine Learning Repository. <https://doi.org/10.24432/C53W3X>.

Reif, M., & Shafait, F. (2014). Efficient Feature Size Reduction via Predictive Forward Selection. Pattern Recognition, 47(4): 1664–1673. <https://doi.org/10.1016/j.patcog.2013.10.009>

Reunanen, J. (2003). Overfitting in Making Comparisons Between Variable Selection Methods. Journal of Machine Learning Research, 3,1371-1382. [reunanen03a.dvi (jmlr.org)](https://www.jmlr.org/papers/volume3/reunanen03a/reunanen03a.pdf)

SciPy community (2023). SciPy (Version 1.11.3). Sphinx. <https://docs.scipy.org/doc/>

Tang, X., Dai, Y., & Xiang, Y. (2019). Feature selection based on feature interactions with application to text categorization. Expert Systems with Applications, 120, 207-216. <https://doi.org/10.1016/j.eswa.2018.11.018>

USAFacts. (2021, July 8). Homicides Increased by 25% but Overall Crime Rate Fell in 2020. <https://usafacts.org/articles/homicides-increased-by-25-but-overall-crime-rate-fell-in-2020/>

Van Hulse, J., Khoshgoftaar, T.M., Napolitano, A. & Wald, R. (2012). Threshold-based feature selection techniques for high-dimensional bioinformatics data. Network Modeling Analysis in Health Informatics and Bioinformatics, 1, 47–61. <https://doi.org/10.1007/s13721-012-0006-6>

Verma, S.S., Lucas, A., Zhang, X., Veturi, Y., Dudek, S., Li, B., Li, R., Urbanowicz, R., Moore, J., Kim, D., & Ritchie, M. (2018). Collective feature selection to identify crucial epistatic variants. BioData Mining **11(**5). <https://doi.org/10.1186/s13040-018-0168-6>

Wang, H., Yan, J., & Yan, X. (2023). Spearman Rank Correlation Screening for Ultrahigh-Dimensional Censored Data. *Proceedings of the AAAI Conference on Artificial Intelligence*, *37*(8), 10104-10112. <https://doi.org/10.1609/aaai.v37i8.26204>

Xu, J., Mu, H., Wang, Y., & Huang, F. (2018). Feature Genes Selection Using Supervised Locally Linear Embedding and Correlation Coefficient for Microarray Classification. *Computational and Mathematical Methods in Medicine,* 2018. <https://doi.org/10.1155/2018/5490513>

Xu. Z., Huang, G., Weinberger, K. Q., & Zheng, A. X. (2014). Gradient boosted feature selection. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '14), 522–531. <https://doi.org/10.1145/2623330.2623635>

YData Lab Inc. (2023). Ydata-profiling (Version 4.6.0). PyPI. <https://pypi.org/project/ydata-profiling/>

Yao, J. & Shepperd, M. 2020. Assessing software defection prediction performance: why using the Matthews correlation coefficient matters. In Proceedings of the 24th International Conference on Evaluation and Assessment in Software Engineering (EASE '20), 120–129. <https://doi.org/10.1145/3383219.3383232>

Yerpude, P. (2020). Predictive Modelling of Crime Data Set Using Data Mining. International Journal of Data Mining & Knowledge Management Process (IJDKP), 7(4). <https://ssrn.com/abstract=3656953>

Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., & Saeed, J. (2020). A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction. Journal of Applied Science and Technology Trends, 1(2), 56 - 70. <https://doi.org/10.38094/jastt1224>

Zhuang, H., Liu, X., Wang, H., Qin, C., Li, Y., Li, W., & Shi, Y. (2021). Diagnosis of Early Stage Parkinson's Disease on Quantitative Susceptibility Mapping Using Complex Network with One-Way ANOVA F-Test Feature Selection. Journal of Mechanics in Medicine and Biology. 21(5). <https://doi.org/10.1142/S0219519421400261>