Ruby Cheung, Saluwa Umuhoza, Garrett Ringler
Data 5322 Final Project
June 7, 2024

Modeling Georgia Recidivism

Abstract

Utilizing recidivism data from the Georgia Department of Community Supervision [3], supervised models were created to learn what variables could lead to recidivism and test if models could predict recidivism with high accuracy. Unsupervised learning was run to determine if clusters could answer any questions toward optimal variables. While 4 key principal components and 11 K-Means clusters were identified this would only explain usefulness of continuous variables in the dataset. Following this analysis of the continuous variables, Gradient Boosting and Support Vector Machine (SVM) models were created. The binary classification model of the Gradient Boosting yielded a test accuracy rate of 73.6%, however it also yielded a false positive rate of 39.6%. A multi-class classification gradient boosting model was also run which was used to indicate if a person would reoffend and if so would they reoffend in one, two or three years. Both the binary and multi-class classification models indicated similar top variables for determining recidivism. The top two variables were percentage of days employed and jobs per year. Gang affiliation, average number of days on parole between drug tests and supervision score also ranked high in the models. Of the three SVM models run (linear, radial basis and polynomial kernels), the linear kernel SVM yielded a 72.6% test accuracy rate and a lower false positive rate of 33.4%. With the results of both the unsupervised and supervised models the team has found that a linear SVM would likely perform the best at predicting general recidivism though it should be noted that further data and testing may be needed to have higher test accuracy and lower false positives.

Introduction

The goal of this project is to develop a model that could accurately predict recidivism as well as examine some of the factors that contribute to recidivism. According to the bureau of justice statistic’s recidivism study with a 10 year follow- up period, among prisoners released in 2008 in 24 states, 82% were arrested in the 10 years following their release [4]. In the last 40 years there has been a 500% increase in incarceration. Prison as a means for community safety results in overcrowding and can be fiscally burdensome, taking away from other needed services in a state budget. It can be disruptive to family structures and communities. Predicting recidivism and understanding what contributes to recidivism rates could provide actionable insight for policymakers and change outcomes for the better. The data used is provided by the Georgia Department of Community Supervision [3]. This data includes information about persons released from prison to parole supervision from January 1, 2013 to December 31, 2015. The original dataset includes 54 fields and 25.8K rows where each row is a record for one individual. After preprocessing the dataset contained 14,170 males and 2,028 females. There are 48 variables that record demographic information such as race and education, information on arrest record, the age of release, prior convictions, and employment information after release. Outcome variables considered are Boolean variables for arrests in year 1, 2 or 3 and arrest within 3 years. For this project principal component analysis and K-Means cluster was deployed for an initial assessment of the underlying trends in the sample. For predictive models, support vector machine technique using linear, polynomial, and radial kernels were utilized. The ensemble method, gradient boosting, was also deployed as a predictive model and to produce variable importance ranks to gain insight into what variables strongly influences recidivism.

Theoretical Background

PCA:

Principal Component Analysis or PCA is a way to reduce dimensionality in a complex dataset. The reduced dimensions of variables are output as principal components that can be used to explain variance in the dataset. The more complex the dataset is, the more principal components it'll typically have, the principal component with the most variance accounted for will be the 1st principal component. Each subsequent principal component will yield lower explained variance up until 100% is explained at the maximum number of components. Figure 1 below illustrates this by showing 40% of the variance is explained by PC1 and an additional 20% by PC2 with less than 5% explained variance per component for PC6 - PC10.

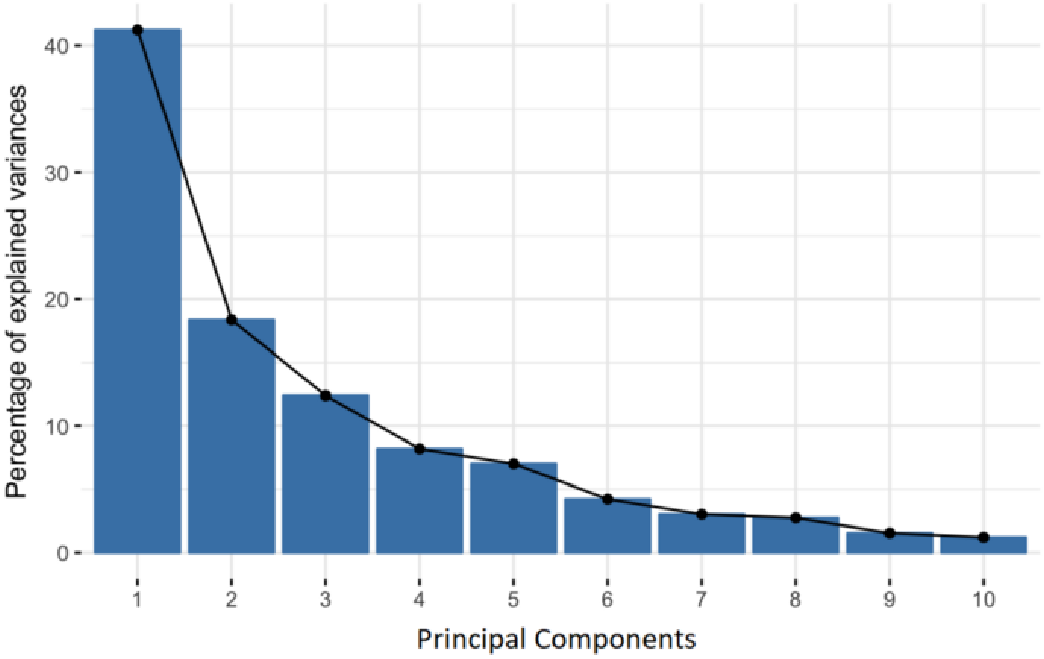

Figure 1. Principal Components [2]

Since the variables in the original dataset make up each of these components they can be graphed against the principal components to show how each variable scaled or even which variables could be stronger predictors. It is of importance to note that since PCA can only be run on continuous variables or variables that can be changed to continuous such as images. Categorical or non-numerical variables will not work with PCA and will throw an error.

K-Means Clustering:

Clustering is a set of techniques to find subgroups within a dataset. The dataset is partitioned into groups where observations within each group are similar to one another. Both clustering and PCA seeks to simplify the data. Clustering differs from PCA in that it does not represent the data in a low dimensionality rather it finds homogenous subgroups. K-means clustering is one of many clustering methods. The number of non-overlapping subgroups is pre-specified. For K-means clustering, a good cluster will have the smallest within-cluster variation possible. The most common choice for measuring within-cluster variation is squared Euclidean distance. The within-cluster variation is the sum of all the pairwise square Euclidean distances divided by the total number of observations within the cluster. The algorithm for K-means clustering is as follows. First, randomly assign each observation to one of the K numbers of clusters. This is the initial cluster assignment. For each of the K clusters the centroid is calculated. Then reassign each observation to the cluster whose centroid is the closest. This is iterated until the cluster assignment no longer changes, or a local optimum has been reached.

K-means is sensitive to initial random clusters because it finds a local optimum rather than a global optimum. To find the optimal model, randomizing the initial cluster configuration is best practice. Another consideration is whether to scale the variables. Again, best practice is to try several different choices and solutions that produce interesting, useful, and interpretable results should be considered. Aside from performing k-means with different parameters, another way to remedy the non-robustness of clustering is to randomly subset the dataset to check for the robustness of the cluster.

Gradient Boosting:

Boosting an ensemble learning method that improves predictive performance of a model by combining the predictions of multiple models. Like any ensemble method boosting models learn by creating weaker models to build a strong model. Boosting models are iterative. It creates a weak learner, typically a simple decision tree, and sequentially tries to correct the predictive error by adjusting the learning algorithm each time focusing weighs rows of data depending on whether they were predicted correctly [4]. Figure 2 shows the general algorithm of a boosting model.

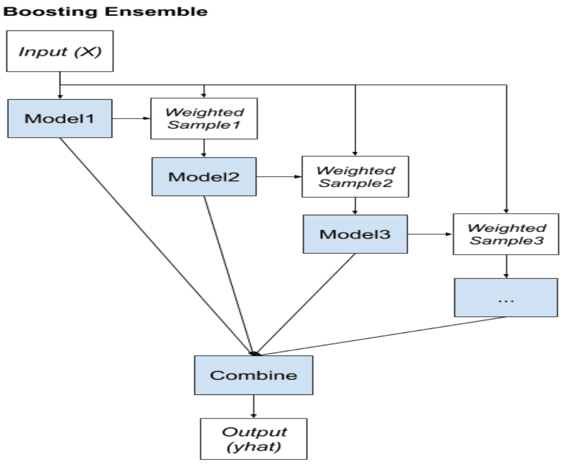

Figure 2. Boosting Algorithm [4]

Gradient boosting models have three main components: a loss function, the weak learner, and the additive model. There are a number of hyperparameters that can be tested to tune the model. The learning rate is the most important hyperparameter, it controls the contribution of the weak learner to the ensemble. Other hyperparameters utilized to tune the model are the maximum depth of each decision tree, maximum number of features to be randomly selected at each split, and the number of sequential trees to be modeled, and the minimum number of samples at each node before considering a split [1]. These hyperparameters are meant to prevent under or over fitting.

SVM:

Support Vector Machines (SVMs) are powerful supervised learning models employed for classification and regression tasks. Their functionality centers around identifying the optimal hyperplane that effectively separates distinct classes within the feature space. At the core of SVMs lies the quest for the maximal margin hyperplane, maximizing the distance between the closest data points of different classes and thus serving as the decision boundary for classifying new instances. When encountering scenarios where linear separation proves challenging, SVMs resort to the kernel trick, transforming the original feature space into higher dimensions where linear separability may be achievable.

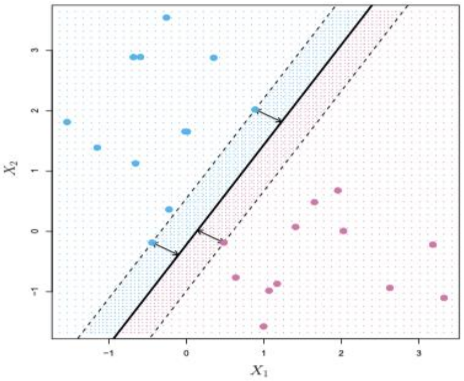

Figure 3. Maximal Margin Hyperplane [2]

Different kernel types play a crucial role in SVMs, each catering to specific data characteristics and complexities. The linear kernel, for instance, is ideal for linearly separable data, defining the decision boundary as a hyperplane governed by a linear equation. In contrast, the polynomial kernel maps data into higher-dimensional spaces using polynomial functions, effectively capturing non-linear decision boundaries. Parameters such as degree and coef0 influence the behavior of the polynomial kernel. Similarly, the radial basis function (RBF) kernel transforms data into an infinite-dimensional space using Gaussian radial basis functions, making it suitable for highly non-linear decision boundaries. The parameters C and gamma regulate the trade-off between margin and classification error, and the influence of training samples on the decision boundary, respectively.

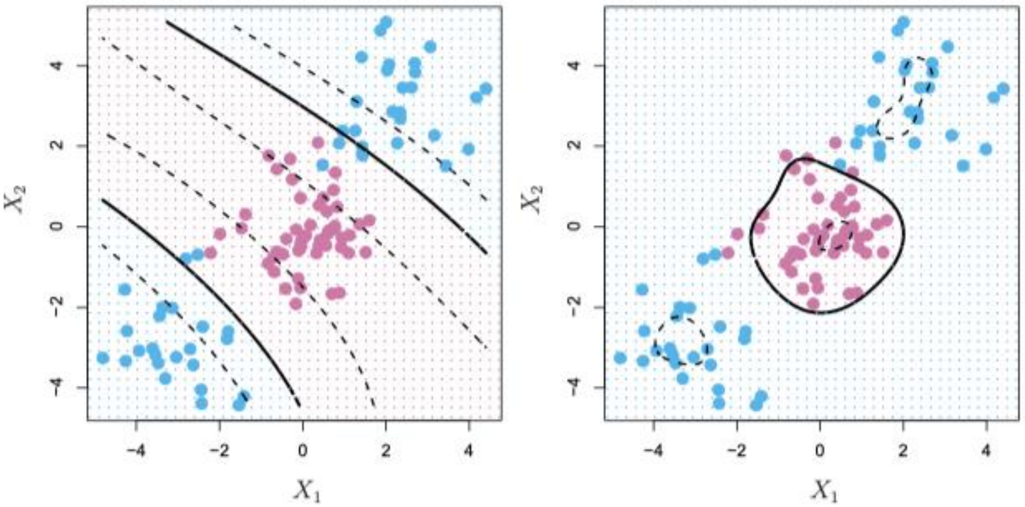

Figure 4. SVM Polynomial kernel [2]

Parameter tuning plays a pivotal role in optimizing SVM performance, with careful consideration needed to avoid overfitting and achieve optimal generalization. Techniques like grid search with cross-validation are commonly employed for this purpose, enabling practitioners to fine-tune kernel types and parameters effectively.

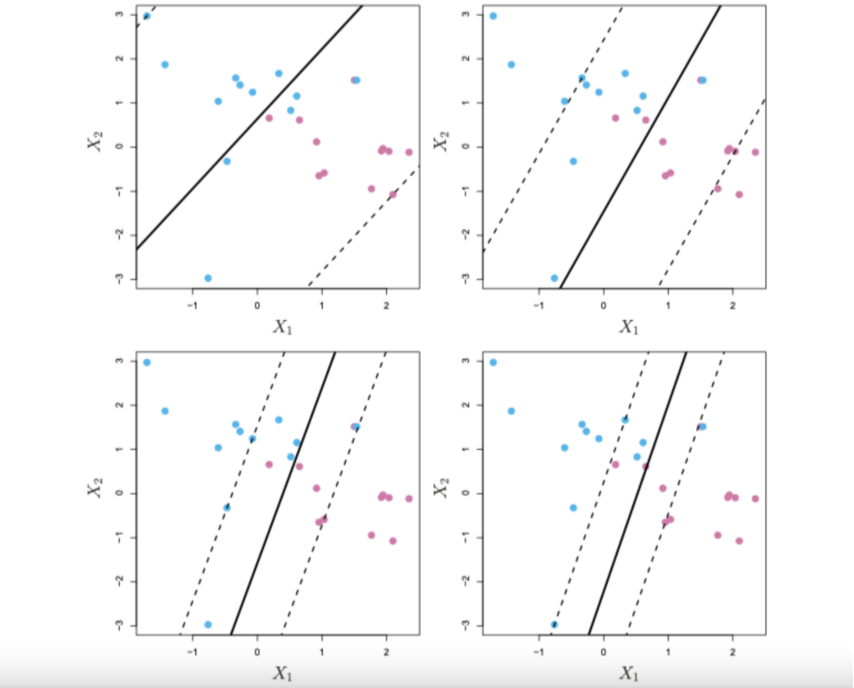

Figure 5. Support Vector Classifier using different tuning parameters [2]

SVMs offer a versatile framework for classification tasks, adept at handling both linear and non-linear decision boundaries. By comprehending the theoretical underpinnings, kernel types, and parameter tuning strategies, practitioners can harness the full potential of SVMs to construct highly accurate predictive models across various applications.

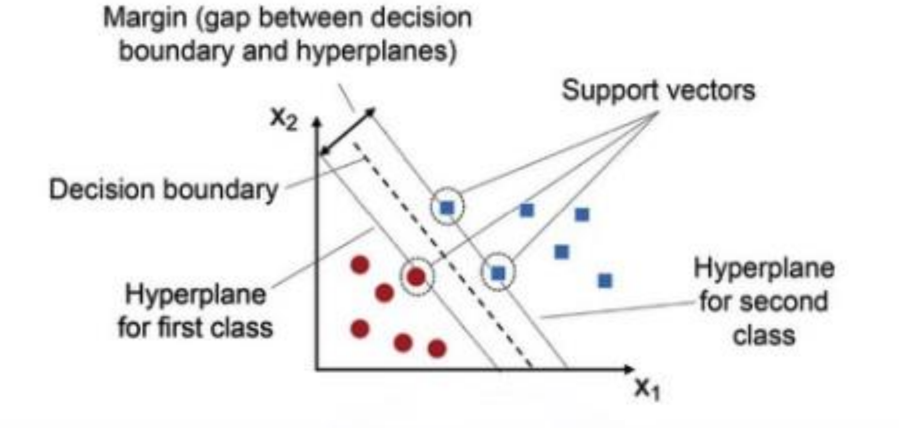
Margin (gap between decision boundary and hyperplanes)
Support vectors
x_2
Decision boundary
Hyperplane for first class
Hyperplane for second class
x_1

Figure 6. Support Vector Classifier [2]

Methodology

Data Cleaning:

The dataset utilized for this study encompassed a range of variables potentially associated with recidivism, including Gender, Race, Age_at_Release, and others such as Supervision_Risk_Score_First and Recidivism_Within_3years. To prepare the data the team first noticed that blanks in the Gang_Affiliated data column needed to be filled since the data was skewed and only males had gang affiliations tied to them. If the team removed all NAs then there would be no females remaining in the filtered dataset. After these blanks were filled with “Unknown”, the data was filtered to drop all rows with missing data. Following this all categorical columns were set to be read as categorical and commas were removed from the two numeric fields that still had them. Certain variables were updated based on predefined conditions to enhance their relevance for predictive modeling. For instance, variables such as 'Dependents', 'Prior_Arrest_Episodes_Felony', and 'Residence_Changes' were modified to reflect specific thresholds, such as '3 or more'. Any of these columns with an ‘or more’ statement were set to cap at the number so in this case the column would have the numbers 1 through 3 instead of 1 thru ‘3 or more’. This allowed the team to treat these columns as numeric as opposed to categorical thereby facilitating more meaningful analysis. For the multiclass classifier, the boolean variables for arrest in year 1,2, 3 were combined to create one variable for no arrest, arrest in year 1, 2, or 3. Finally, columns that were unnecessary for the model such as ID, Training_Sample, Recidivism_Arrest_Year1, 2, and 3 were all excluded from the data the models were trained on. After all of this was done there remained 16,198 rows of observations of the original 25,000.

Two datasets were created from the filtered data, one with dummy variables plus all numeric columns and one with scaled dummy variables and scaled numeric variables to allow for models to run regardless of needs. Each of these could be subset for certain models to allow for quicker runtimes. To evaluate the predictive performance of the models, the dataset was divided into training and testing sets. A test size of 0.2 (i.e., 20% of the data) was chosen to allocate a sufficient amount of data for evaluation while retaining the majority for training purposes. This partitioning ensured that the models were trained on a representative subset of the data and evaluated on an independent subset, thereby providing a robust assessment of their performance.

PCA/K-Means:

In order to deploy PCA and K-Means the dataset could only contain continuous variables. The dataset used included average days on parole between drug tests, percentage of positive drug tests for THC, percentage of positive drug tests for cocaine, percentage of positive drug tests for meth, percentage of positive drug tests for other drugs, percent of day employed, and number of jobs per year. These were all scaled before training the model. The maximum number of principal components was 6. The general rule of thumb is to maintain approximate 80-90% of variance. Figure 7 shows the cumulative explained variance for number of components 0 thru 6. At 4 components 85.97% of the variance is explained. This is the number of components that will be used in the K-means model.

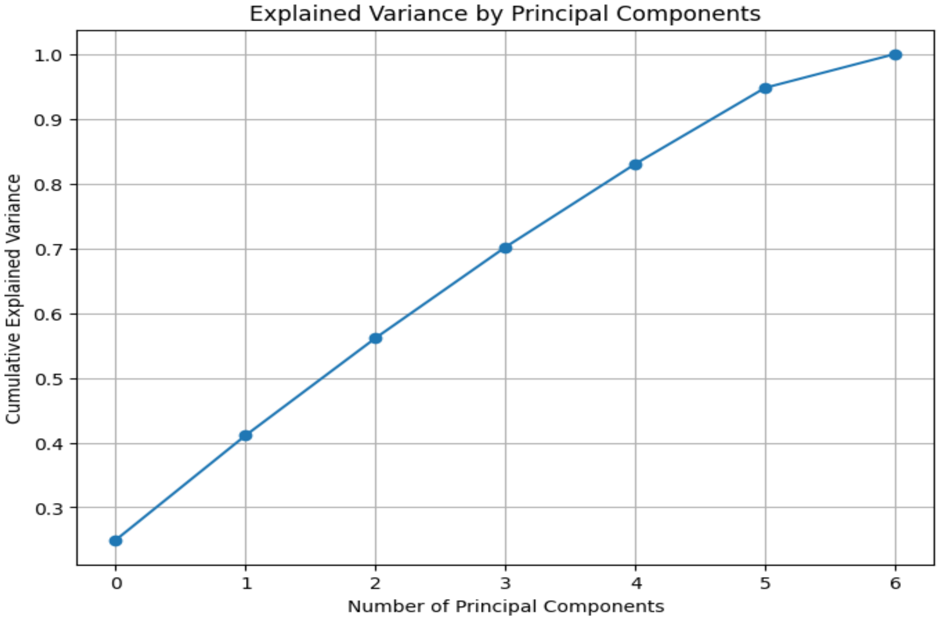

Figure 7. Variance explained by principal components

To select the optimal cluster number, a range of K- means models with 1-15 clusters were trained on the scaled data. Figure 8 shows the result of this initial test. The score represents the within cluster sum of squared distances of the observations to their cluster centroid. As the number of clusters increases the score improves and becomes smaller, the observations are closer to the cluster centroid. An increase in cluster numbers also runs the risk of overfitting. Using the ‘elbow method’, the optimal cluster number is where the marginal improvement of the score begins to diminish and level off. Using the optimal cluster number of 11 a final K-Means model is trained. The method of initialization is a parameter that determines where the initial cluster is set. The initialization was set to ‘k-means++’, which selects initial centroids based on the probability distribution of the points’ contribution to the overall inertia.

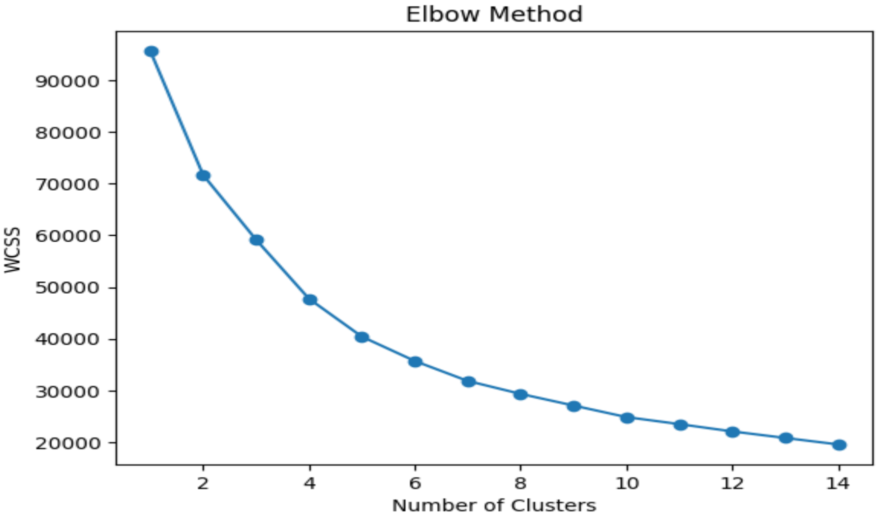

Figure 8. Clusters: Within cluster sum of squares

Gradient Boosting:

Two gradient boosting models were deployed. The first is a multiclass classifier, predicting whether an individual will reoffend in zero, one, two or three years. The second was a binary classifier, predicting whether an individual will reoffend or not. Five-fold cross validation was used to find the optimal hyperparameter values. Cross validation was used to find the optimal number of sequential trees, choosing from values 50, 100, 150 and 200, the maximum depth of the trees, choosing from values 1-20, and the minimum number of samples split which defines the minimum number of samples in a node to be considered before splitting, choosing from values 50, 100, 150, and 200. The learning rate was set 0.1, considered a slow learning rate. The maximum number of features was set at 30% of the total number of features, 32.

For the multiclass classifier the optimal values found for number of sequential trees was 100, maximum depth of the trees is 5, and minimum number of samples in a node 100. For the binary classifier the optimal values for maximum depth is 3, minimum number of samples split is 50, and the number of sequential trees is 200. For comparison, the dataset was split by gender, one for males and one for females, and a binary classifier was trained using each dataset. For the female inmate model, cross validation found the optimal values for maximum tree depth to be 3, minimum number of samples in a node to be 100, and number of sequential trees to be 100.

SVM:

Our methodology involved the meticulous tuning of parameters for each kernel using GridSearchCV, enabling an exhaustive exploration of parameter combinations to optimize model performance. Specifically, parameters such as 'C', 'degree', and 'gamma' were fine-tuned for Linear, Polynomial, and RBF kernels, respectively. This rigorous parameter tuning process aimed to identify the configurations that would best leverage the dataset's characteristics for accurate predictions. Subsequently, three Support Vector Machine (SVM) models were trained using the selected kernels: Linear, Polynomial, and RBF.

Following parameter tuning, we conducted a comprehensive analysis to compare the performance of the trained models. Key metrics such as accuracy and training time were meticulously evaluated to provide insights into each model's predictive capabilities. Leveraging cross-validation techniques, we assessed the accuracy of each model across different subsets of the data, ensuring robustness in our evaluation. Additionally, we recorded the time taken to fit each model to the training data, allowing us to gauge their computational efficiency. By systematically comparing the models, our aim was to identify the most suitable SVM kernel for predicting recidivism based on the dataset's features. This methodological approach ensured a rigorous evaluation process, enabling the selection of the most effective predictive model.

Computational Results

PCA/K-Means:

A K-Means model was deployed using the 4 principal components and 11 clusters and initial centroid selection using the ‘k-means++’ method. In the 11 cluster, the number of observations ranged from 52 to 4892. Figures 9 and 10 are plots of the 11 clusters using the features of the percentage of days employed, jobs per year, other positive drug tests, and average number days per positive drug test. The cluster plots using the features reveal the characteristics of the samples within the clusters.

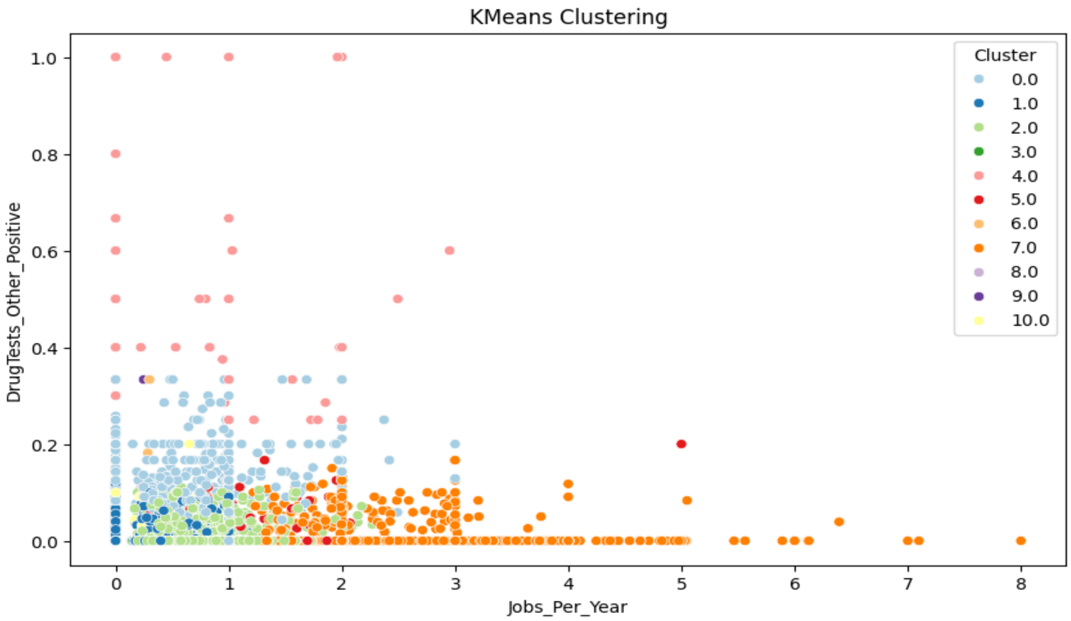

Figure 9. K-Means: Jobs per year and percentage of other positive drug tests

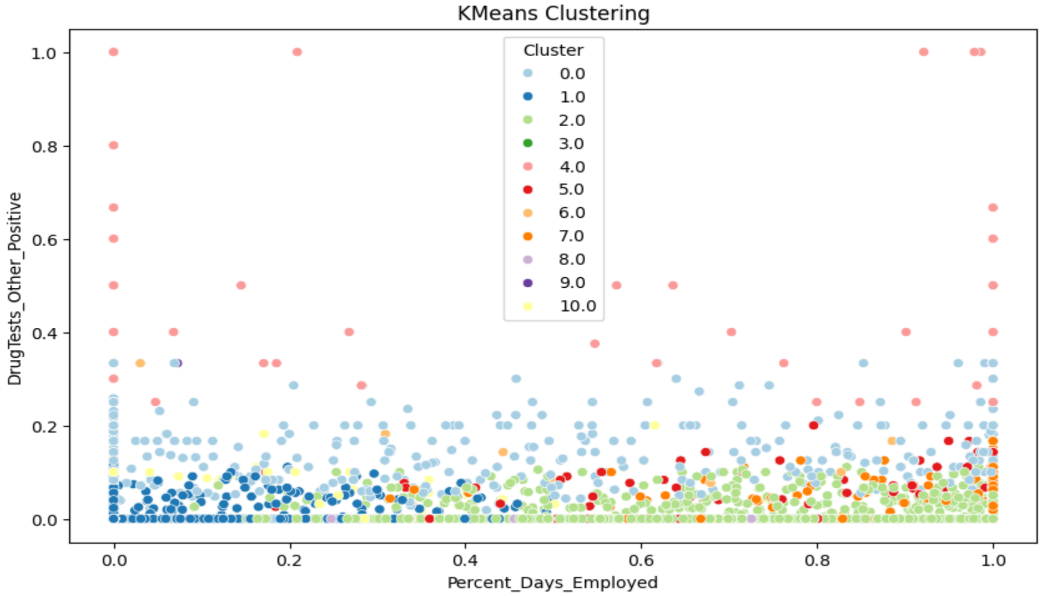

Figure 10. K-Means: Percentage of days employed and percentage of other positive drug tests

A table of summary statistics for each cluster was produced [Table 1].

Gradient Boosting:

The binary classifier trained using gradient boosting predicted whether an individual would or would not reoffend within 3 years. It achieved an overall test accuracy of 73.6%. Figure 11 is the confusion matrix, the false positive rate from this model is 39.2%. A variable importance plot, figure 12, was also created to show the top 15 features that influences whether an individual will reoffend.

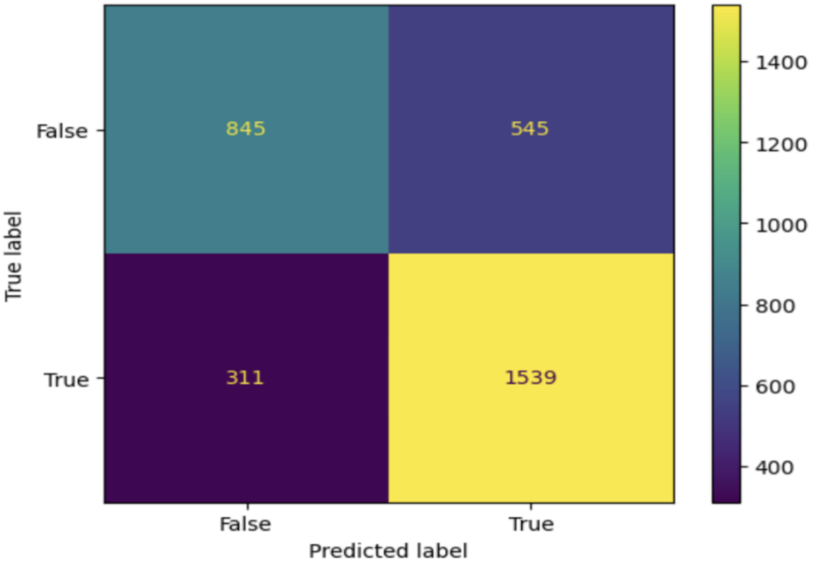

Figure 11. Binary Classification, Confusion Matrix

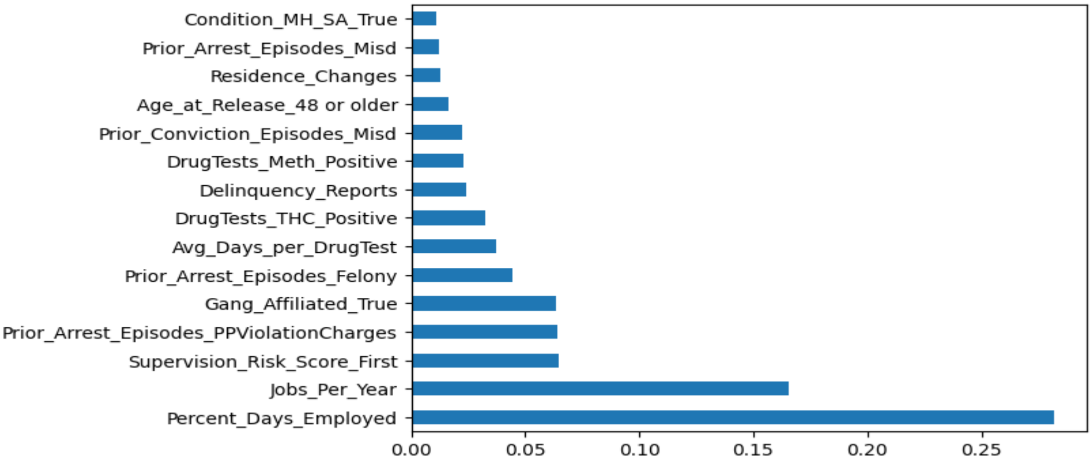

Figure 12. Binary Classification, Variable Importance

The multiclass classifier predicted whether an individual would not reoffend, reoffend in one, two, or three years. The overall accuracy rate for this model was 58.2%. Figure 13 is the confusion matrix for this model and Figure 14 is the variable importance plot for this model.

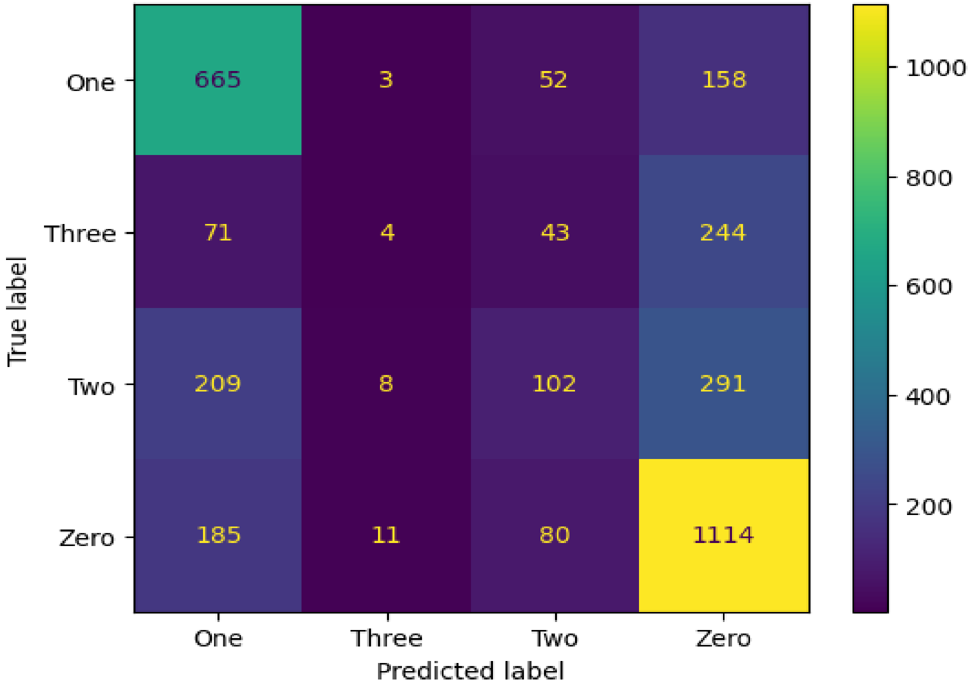

Figure 14. Multiclass Classifier, Confusion Matrix

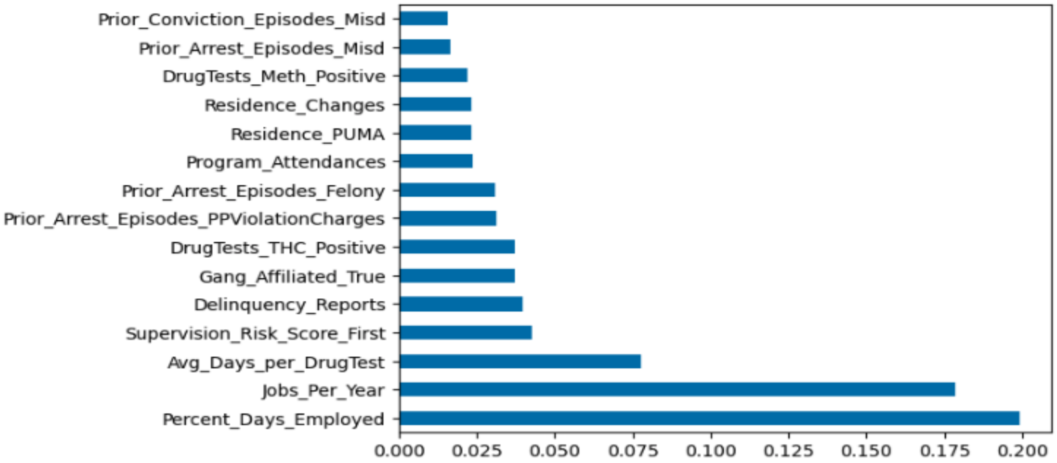

Figure 15. Multiclass Classifier, Variable importance plot

The binary classifier trained using data from only the female inmates achieved an accuracy of 70.7% Figure 16 and 17 are the confusion matrix and feature importance for the female inmate model.

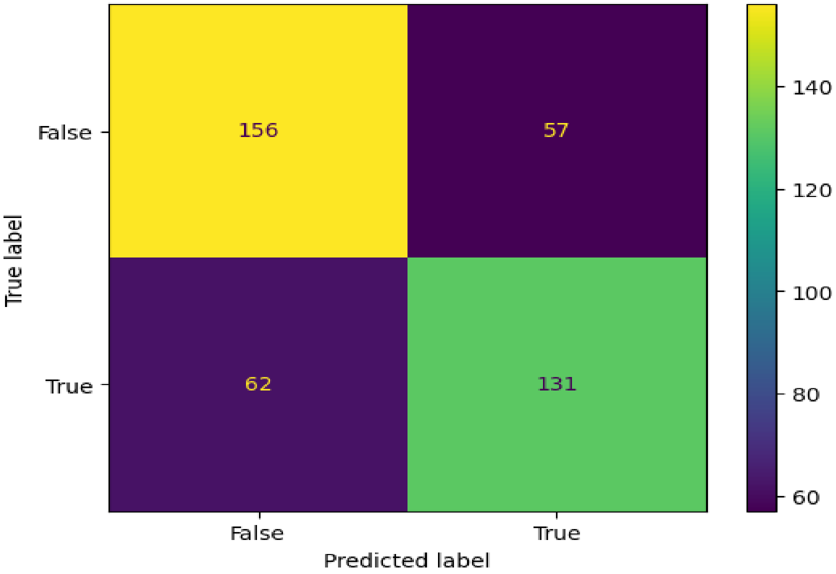

Figure 16. Binary Classifier for Females, Confusion Matrix

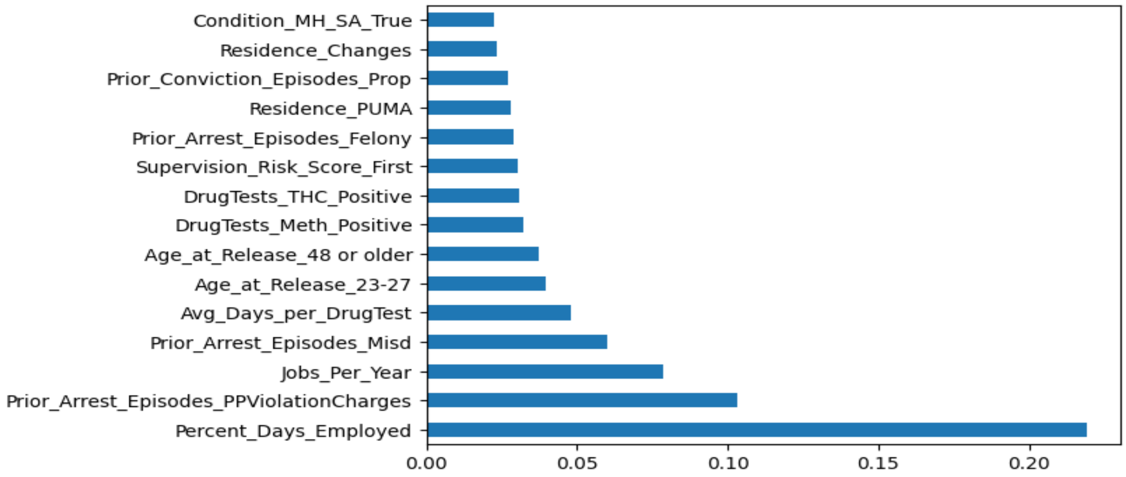

Figure 17. Binary Classifier for females, Variable Importance Plot

The binary classifier for male parolees reached an accuracy rate of 73.8%, with a false positive rate of 42.4%. Figures 18 and 19 are the confusion table and variable importance plot for the male classifier.

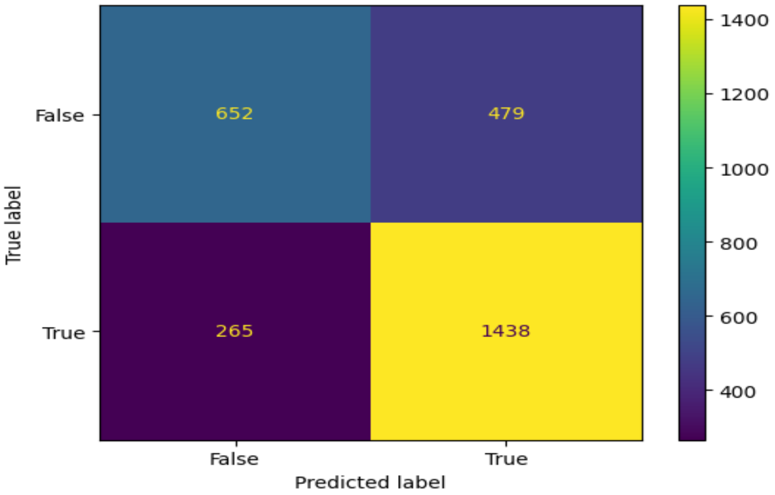

Figure 18. Binary Classifier for Males: Confusion Matrix

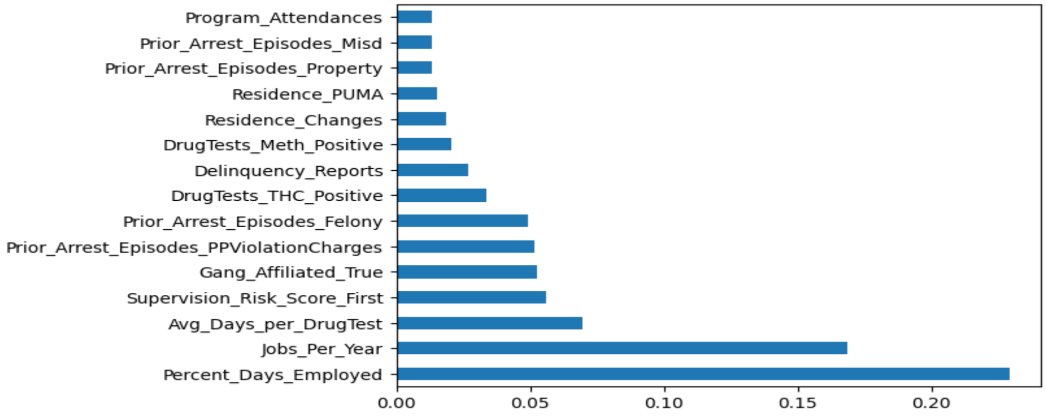

Figure 19. Binary Classification for Males: Variable importance plot

SVM:

Our analysis focused on predicting recidivism using different kernel functions in Support Vector Machine (SVM) models and visualizing their decision boundaries.

First, we identified the best parameters for each kernel: {'C': 0.1} for the Linear Kernel, {'C': 1, 'degree': 3} for the Polynomial Kernel, and {'C': 1, 'gamma': 0.01} for the RBF Kernel. Subsequently, we evaluated the models' performance in terms of accuracy and training time.

The Linear Kernel exhibited the highest accuracy score of 0.726, albeit with a longer training time of approximately 10.56 seconds. In contrast, the Polynomial Kernel achieved a lower accuracy score of 0.713 but trained faster, approximately 8.38 seconds. The RBF Kernel fell between the other two, with a moderate accuracy score of 0.716 and a faster training time of 7.8 seconds.

In the confusion matrix, the RBF kernel accurately identified 48.9% of true negatives and 71.8% of true positives, while yielding false positive and false negative rates of 38.1% and 22.0%, respectively. Comparatively, the Linear kernel demonstrated slightly better performance in true negatives (50.1%) and false positives (33.4%), with similar rates of false negatives (22.5%) and true positives (69.9%). Meanwhile, the Polynomial kernel's performance was intermediate, with 50.8% true negatives and 68.1% true positives, alongside false positive and false negative rates of 35.5% and 25.1%, respectively. These percentages offer insights into the models' abilities to accurately predict recidivism and highlight the trade-offs between true and false predictions across different kernels.

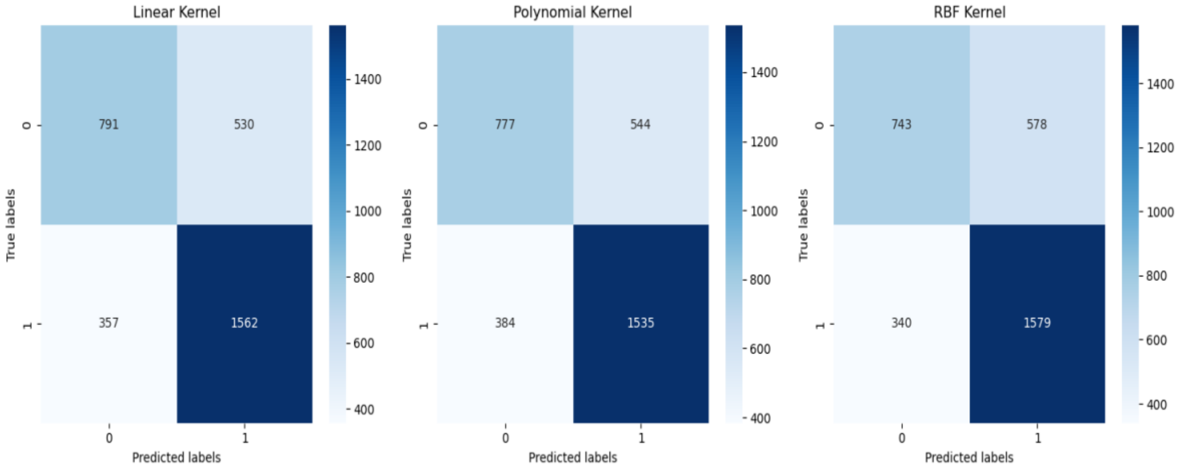

Figure 19. Confusion Matrix for Linear, Polynomial, and RBF Kernels

These results provide valuable insights into the performance of each kernel in predicting recidivism. The linear kernel demonstrates a notable balance between true positives and true negatives, suggesting its effectiveness in capturing the underlying relationships in the data.

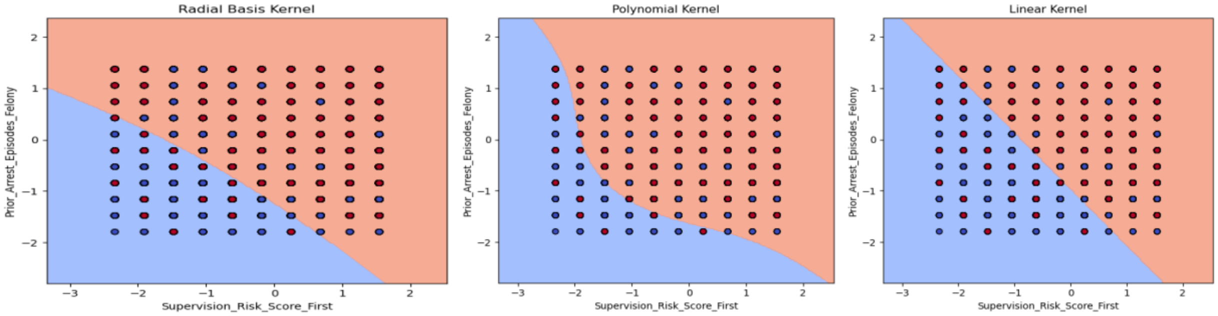

Figure 20. Plots for supervision risk score and prior arrest episodes for felony charges for each kernel

Visualizing decision boundaries using the features Supervision_Risk_Score_First and Prior_Arrest_Episodes_Felony provided insights into the models' predictive capabilities. Despite its simplicity, the linear kernel model demonstrated distinct decision boundaries, effectively separating recidivism classes and suggesting a strong ability to capture underlying relationships in the data. This finding positions the linear kernel as a promising choice for predicting recidivism based on these features.

Discussion

PCA/K-Means:

The objective of the K-means model is to partition the dataset into groups of similar data points. The table revealed some characteristics of the clusters that are not apparent looking at the plots[Table 1]. Cluster 7, for example, had the lowest average days on parole between drug tests, 55 (STD: 50.853651), and also had the highest average percentage of days employed, 95% (STD:0.108950). Cluster 1 and 2 had similar average number of days on parole between drug test, cluster 1: 68.77 days (STD: 54.52) and cluster 2: 68.01 days (STD:53.15), and overall the lowest averages percentages across all types of drug tests, what differentiates cluster 1 is a low average percentage of employed days, 5% (STD: 1%) and cluster 2 had a high percentage of employed days, 77% (STD: 24%). Grouping as such could be useful in developing targeted strategies to help prevent recidivism.

Gradient Boosting:

While the binary classifier performed much better than the multiclass classifier, it is notable that the two models had the same variables in their top 15 variables of importance. Percentage of days employed and jobs per year were the two top variables of importance. Gang affiliations, average number of days on parole between drug tests and supervision risk score also ranked high in these models. The difference in accuracy scores between the binary class and multiclass model suggest that it is much more challenging to determine when an individual is likely to reoffend. It is possible that a time-series analysis would achieve higher predictive performance and offer more insight into what influences when an individual is likely to reoffend.

The binary classifiers for males performed slightly better than the model for females. The datasets for each model varied in number of observations and gang affiliation was unrecorded among females, these differences may contribute to the discrepancy. Percentage of days employed was the top variable of importance in both models. Similar to the general binary classifier, the number of jobs per year ranked number two in the male model. For the females, the second most important variable was the number of prior arrests with probation or parole violation charges, which ranked number 6 for males. Gang affiliation remained of high rank for males as well as supervision risk score, number 4 in rank for males. For females, supervision risk score ranked number 10, age of release variables also ranked higher for females and did not appear in the top 15 for men. The predictive performance of the models did not differ significantly enough to justify separate predictive models for females and males. If the objective were to develop an interpretable model to understand what influences recidivism, it would be suggested to train models separately to gather more meaningful insight.

SVM:

The analysis of various kernel functions within Support Vector Machine (SVM) models offers valuable insights into their effectiveness in predicting recidivism. Our findings highlight the critical role of parameter selection in optimizing model performance, showcasing distinct trade-offs between accuracy, training time, and predictive capabilities.

Through meticulous parameter tuning, we identified optimal configurations for each kernel, underscoring the importance of fine-tuning model parameters. Subsequent evaluations revealed intriguing performance patterns: the Linear Kernel emerged with the highest accuracy, albeit requiring longer training times. In contrast, the Polynomial Kernel exhibited faster training times despite slightly lower accuracy. Meanwhile, the RBF Kernel struck a balance between the two, demonstrating moderate accuracy with quicker training times.

However, it's imperative to consider the implications of false positives, false negatives, true positives, and true negatives on recidivism prediction. False positives may lead to unwarranted interventions or harsher penalties for individuals unlikely to reoffend, exacerbating issues like over-incarceration. Conversely, false negatives may result in missed opportunities for intervention, allowing high-risk individuals to remain undetected within the system.

In contrast, true positives and true negatives signify accurate predictions of recidivism outcomes, facilitating appropriate interventions or reducing unnecessary measures. The confusion matrices provide further insight: the Linear kernel showcases a notable balance between true positives and true negatives, indicating its effectiveness in capturing underlying data relationships. This underscores the importance of evaluating predictive outcomes and highlights the potential of the linear kernel for recidivism prediction

Conclusion:

Considering the results obtained from the Linear kernel in the Support Vector Machine (SVM) model and the Gradient Boosting models, the Linear kernel SVM emerges as the preferred approach for predicting recidivism. The Linear kernel SVM model demonstrated a balanced performance with true negatives accounting for 50.1% of the total predictions and false positives making up 33.4%. This indicates its effectiveness in correctly identifying individuals who did not recidivate while also highlighting areas for improvement in classification accuracy. Additionally, false negatives represented 22.5% of the total predictions, suggesting room for enhancement in capturing all instances of recidivism. On the other hand, the Gradient Boosting models, particularly the binary classifier, showcased promising accuracy rates but also presented significant false positive rates, such as 39.2% in the binary classifier trained using gradient boosting. Additionally, while the binary classifiers for males performed slightly better than those for females, the differences in predictive performance did not justify separate models for each gender. The K-means model provided insights into clustering patterns within the dataset, revealing characteristics of different clusters that could inform targeted strategies to prevent recidivism. However, the clustering analysis did not directly contribute to predictive modeling for recidivism. Overall, the Linear kernel SVM model stands out for its balanced performance, computational efficiency, and interpretability. Its ability to effectively capture underlying data relationships and maintain a balance between true positives and true negatives makes it a valuable tool for recidivism prediction. Therefore, based on the comprehensive evaluation of model performance and implications for practical applications, the Linear kernel SVM emerges as the preferred approach for predicting recidivism.

References

1. Jain, Aarshay. *Complete Machine Learning Guide to Parameter Tuning in Gradient Boosting (GBM) in Python.* June 15, 2022. https://www.analyticsvidhya.com/blog/2016/02/complete-guide-parameter-tuning-gradient-boosting-gbm-python/
2. James, Gareth. Witten, Daniela. Hastie, Trevor. Tibshirani, Robert. *An Introduction to Statistical Learning with application in R*. New York, Springer Science+Business Media, 2013.
3. Georgia Department of Community Supervision. *National Institute of Justice's Recidivism Challenge Full Dataset*. https://data.ojp.usdoj.gov/Courts/NIJ-s-Recidivism- Challenge-Full-Dataset/ynf5-u8nk/data_preview
4. Brownlee, Jason. *A Gentle introduction to Ensemble Learning Algorithms.* April 27, 2021. https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/

Table 1.

Cluster	Count	Average days per drug test (Mean)	Average days per drug test (STD)	Percentage THC positive drug test (Mean)	Percentage THC positive drug test (STD)	Percentage cocaine positive drug test (Mean)	Percentage cocaine positive drug test (STD)	Percentage meth positive drug test (Mean)
0	871	68.04	63.38	0.09	0.12	0.01	0.03	0.15
1	4892	68.77	54.52	0.03	0.06	0.00	0.01	0.00
2	7061	68.01	53.15	0.02	0.05	0.00	0.01	0.00
3	350	535.00	293.30	0.51	0.42	0.01	0.04	0.00
4	67	150.98	176.61	0.18	0.17	0.10	0.17	0.09
5	867	73.05	53.57	0.25	0.17	0.07	0.08	0.00
6	209	150.40	144.13	0.27	0.23	0.46	0.17	0.00
7	2906	55.36	50.85	0.03	0.06	0.00	0.01	0.01
8	1256	337.31	125.79	0.05	0.11	0.00	0.00	0.00
9	52	301.60	248.14	0.18	0.24	0.00	0.00	0.78
10	947	74.04	59.22	0.26	0.18	0.08	0.09	0.00

Cluster	Percentage meth positive drug test (STD)	Percentage other positive drug test (Mean)	Percentage other positive drug test (STD)	Percentage of days employed (Mean)	Percentage of days employed (STD)	Number of jobs per year (Mean)	Number of jobs per year (STD)
0	0.11	0.08	0.09	0.41	0.40	0.65	0.67
1	0.02	0.00	0.01	0.05	0.10	0.15	0.27
2	0.02	0.00	0.01	0.77	0.24	0.85	0.37
3	0.03	0.00	0.00	0.30	0.38	0.38	0.51
4	0.13	0.49	0.24	0.38	0.44	0.71	0.84
5	0.02	0.01	0.02	0.80	0.23	1.25	0.62

6	0.00	0.01	0.05	0.27	0.38	0.51	0.70
7	0.02	0.00	0.02	0.96	0.11	2.25	0.74
8	0.02	0.00	0.00	0.40	0.36	0.42	0.35
9	0.25	0.01	0.05	0.35	0.42	0.59	0.74
10	0.02	0.01	0.03	0.06	0.13	0.16	0.31