Chicago Crime Pattern and Hotspot Detection

Final Report

Check out our Proposal
Check out our Midterm Report

Introduction & Background

Chicago faces significant crime challenges, including high rates of homicides, armed robberies, and gang violence. Although smaller than cities like New York and Los Angeles, Chicago's crime rate remains comparatively higher [1]. Traditional law enforcement has shown limitations in their traditional approaches, leading researchers and policymakers to explore data-driven methods, such as machine learning, to predict and mitigate crime.

Machine learning enables the analysis of historical crime data to uncover patterns and potential hotspots. Studies like the University of Chicago's "Algorithm Predicts Crime and Police Bias" have shown that predictive models can reveal crime patterns while taking socio-economic factors into account, potentially reducing enforcement biases [2]. Additionally, research by Chattopadhyay (2022) highlighted enforcement biases across socio-economic areas, emphasizing the need for equitable prediction tools [2]. This is applicable due to the perception of the socially-acknowledged South Side of Chicago which makes every other surrounding area appear low-crime or the 'South Side' seem too fearful to even protect. LevelUp's project further identified links between crime rates and local infrastructure, supporting the integration of contextual data in predictive models [3]. These studies underscore machine learning's potential to improve public safety outcomes with factual data in cities like Chicago.

Dataset Description

The Chicago Crime Dataset, sourced from the City of Chicago's public repository, includes 7 million entries and 22 features, each representing a crime instance with details such as type, location, and arrest status. This dataset updates daily with new crime incidents.

Problem Definiton

Chicago's persistent crime challenges demand innovative approaches to enhance public safety. Despite the availability of a large, detailed crime dataset, real-time insights for effective crime prevention and resource allocation remain difficult to derive. Traditional approaches often focus on socially-defined high-crime areas like the South Side of Chicago, perpetuating enforcement biases and neglecting nuanced patterns in low-crime districts.

This project aims to develop regression-based machine learning models capable of predicting the frequency of specific types of crimes across Chicago's neighborhoods. By leveraging geospatial and temporal features—such as longitude, latitude, time of day, and seasonal variations—our goal is to identify emerging crime and forecast crime trends. Additionally, the project seeks to address data imbalances and incorporate feature manipulation techniques to ensure accurate predictions across high- and low-crime areas, ultimately promoting equitable and data-driven crime prevention strategies.

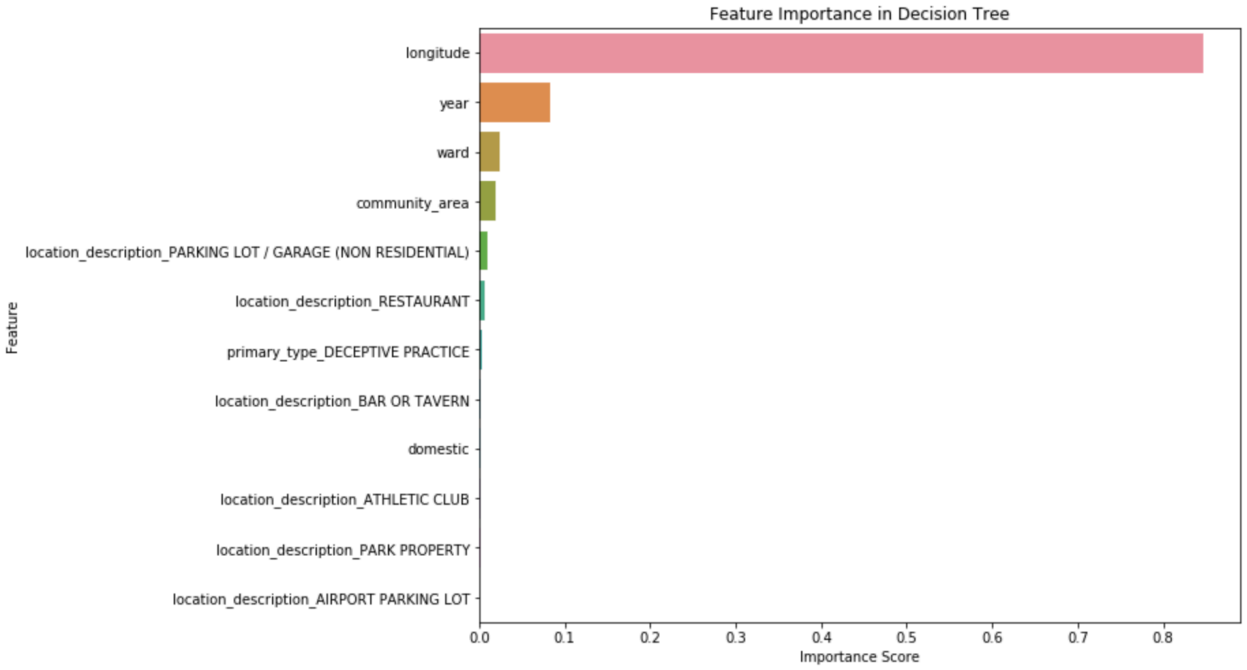
Methods

Pre-Processing Methods

Given the large dataset on Chicago crime, we performed the following pre-processing methods:

1. Data Cleaning
    1. This dataset updates daily, continually adding new records. To focus on recent data and make our analysis more manageable, we constrained the dataset to records from January 1st, 2023 to October 1st, 2024. We dropped records with missing values in essential features to ensure data quality and consistency.
2. Feature Selection and Engineering
    1. We selected features we believe may impact crime frequency the most out of the 22 features by dropping some features. Temporal features, such as month, day, and hour, were extracted from the data field, as each of these components could independently influence crime trends. This granularity allows us to capture temporal patterns in crime occurrence.
3. Encoding Categorical Values
    1. Since regression models work with continuous variables, we encode categorical variables (e.g. crime type, locations, location descriptions, primary type, and whether it's domestic) using One-Hot Encoding to convert them into binary columns.
4. Data Aggregation and Feature Scaling
    1. We aggregated crime counts with temporal, spatial, and encoded features. To standardize these values, we used StandardScaler from sklearn.preprocessing to prevent features with large scales from skewing the model a lot.

5. Recursive Feature Elimination (RFE)
    1. We applied recursive feature elimination (RFE) from sklearn to select the most predictive features for crime frequency on a Linear Regression model. RFE works with any model, so we chose linear regression since we are conducting pre-processing for a linear regression model. RFE removes the less important features and identifies the k strongest predictors based on importance to the model. In our case, we made k = 10 because the more features allowed, the more context our model has, but too many features could make the model overfit. This method was chosen for its ability to rank features in terms of predictive strength, allowing us to reduce the feature set to those most relevant.
    2. For example, here is a decision tree plot to show the importance of the 10 selected features into predicting crime counts using our Linear Regression Model:



    As you can see, longitude is the largest contributor in predicting crime count. What is surprising to see is that latitude was not selected as an important feature in prediction despite the social disparity between North Chicago and South Chicago.

6. Train-Test Split
    1. Using sklearn.model_selection's train_test_split, we split our processed data where 70% is training and 30% is testing. This ratio is quite standard in the industry. Going forward, the model will be trained on the training data. During the model evaluation phase, we will test the model on the testing data and compare.

7. Log Data
    1. Taking the log of the training and test data will reduce the complexity and imbalance by significantly reducing the outlier values to relative change and minimally reducing the non-outlier values.
        1. Chicago crime data is quite imbalanced due to the concentration points in the city, unpredictable (non-organized) crime, and imbalance of socioeconomic areas in crime frequency and type of crime.

8. SMOTE
    1. Can oversample the minority class to reduce imbalance data since the model can be heavily skewed towards locations that are acknowledged to have high crime counts.
        1. Can improve predictions for crime in low-crime districts
    2. Tools:
        1. Using imblearn.over_sampling's SMOTE, we were able to balance the dataset by generating synthetic samples for our minority classes. This is an important step in order to account for the observation that there is significant imbalance in the target variable, which in this case is the crime_count.

Other ways to preprocess the data include dimensionality reduction. Two ways we can do this is with Lasso Regression and PCA. Lasso Regression (scikit-learn.linear_model's Lasso) conducts feature elimination based on the strength of the features by minimizing the weights of high variance features. We can also use PCA to linearly reduce high-dimensional socio-economic data, capturing key components while omitting noise. PCA groups together highly correlated features as key components, and we can use scikit-learn to efficiently use the preprocessing method.

Machine Learning Algorithms

1. Linear Regression
    1. Acts as a baseline model to predict the frequency of crime occurrences (sets the stage for assessing the effectiveness of future complex models).
    2. Analyzes the quantitative relationships between crime frequency and features such as socio-economic and temporal types.
    3. Efficient and interpretable when trying to evaluate the overall impact that factors such as month, day, and neighborhood characteristics have.
    4. Tools: LinearRegression from scikit-learn.
        1. The model offers a reliable evaluation of crime frequency based on the prediction features previously selected.
2. Random Forest Classifier (Regression)
    1. Predicts crime frequency based on socio-economic and temporal features.
    2. Is ideal for handling large datasets and complex relationships well among variables. Given that Chicago's crime data is extensive and multidimensional, a random forest model can handle these challenges by combining insights from multiple decision trees.
    3. Effective at managing the noisy crime data while avoiding overfitting, which is significantly important considering the unpredictability of Chicago's crime data. Key advantages include robustness to noise, feature importance analysis, and scalability.
    4. Tools: RandomForestClassifier from scikit-learn.
        1. Aggregates crime frequency predictions from multiple decision trees based on the determined features.

2. Evaluated using metrics such as MAE, MSE, and R^2 to assess the accuracy and reliability of the model.
3. Hyperparameters:
    1. Number of Estimators: 100
    2. Random_State: 42
    3. Max Depth of Decision Trees: 10

3. XGBoost
    1. XGBoost is a type of ensemble learning decision tree algorithm reputed for efficient training for Chicago's multidimensional data, specifically boosting.
        1. Boosting is a type of ensemble learning that builds a stronger model by refining preceding weaker models.
    2. XGBoost also takes care of some of the pre-processing, specifically handling missing values.
    3. XGBoost is well-suited for noisy datasets like Chicago's crime data due to its regularization using L1 and L2 norm, preventing overfitting.
    4. Tools: XGBClassifier from xgboost library.
        1. The model uses an ensemble of decision trees to capture nonlinear relationships among socio-economic and temporal features, refining predictions at each step.
        2. I experimented around to see if the accuracy can increase by playing with the parameters. After doing so, we initialized the following hyperparameters:
            1. Number of estimators: 200
            2. Learning Rate: 0.5
            3. Max Depth of Decision Trees: 8

Results and Discussion

Linear Regression

Analyzing Quantitative Metrics

Let's first dive deeper into the quantitative metrics from the model: Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-Squared. These metrics provide insights into the predictive power and accuracy of the model.

The MAE is 0.0180, which is relatively low based on the scale of the database. The MAE represents the average magnitude of error in predictions. Therefore, this low MAE indicates closer predictions to the actual values.

The MSE is similar to the MAE but penalizes larger errors more due to the squaring, which helps to highlight significant deviations in the predictions. In our case, the MSE is also low at a value of 0.0125.

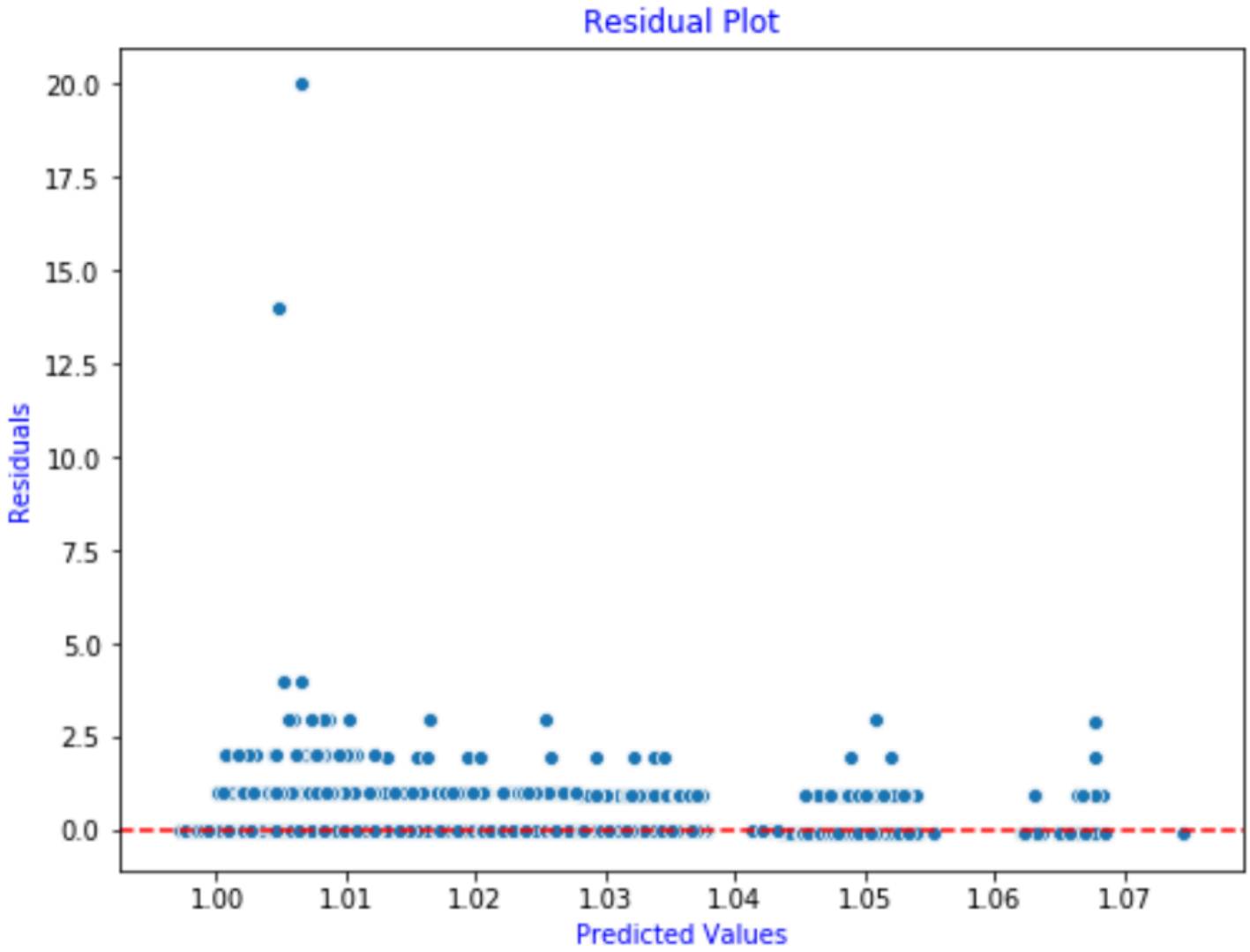
The R-Squared value indicates the proportion of variance in the target variable which is explained by the model. Generally, values closer to 1.0 suggest a better fitting model; however, in our case, the value of R-Squared is 0.0039 which is very low. Therefore, this suggests that the model does not explain much variance in crime counts. Since the value is close to 0, this shows that the model is not accurately able to capture and identify meaningful patterns in the data to help explain the crime counts. This might be due to potential missing predictive features, insufficient complexity in the model, or it could be possible that the crime data is inherently difficult to predict with simple features which might be the most probable case given the complexity of the data set.

Visualizations

The following visualizations were created to further understand the performance of the model and how the predictions of the model align with the actual dataset.

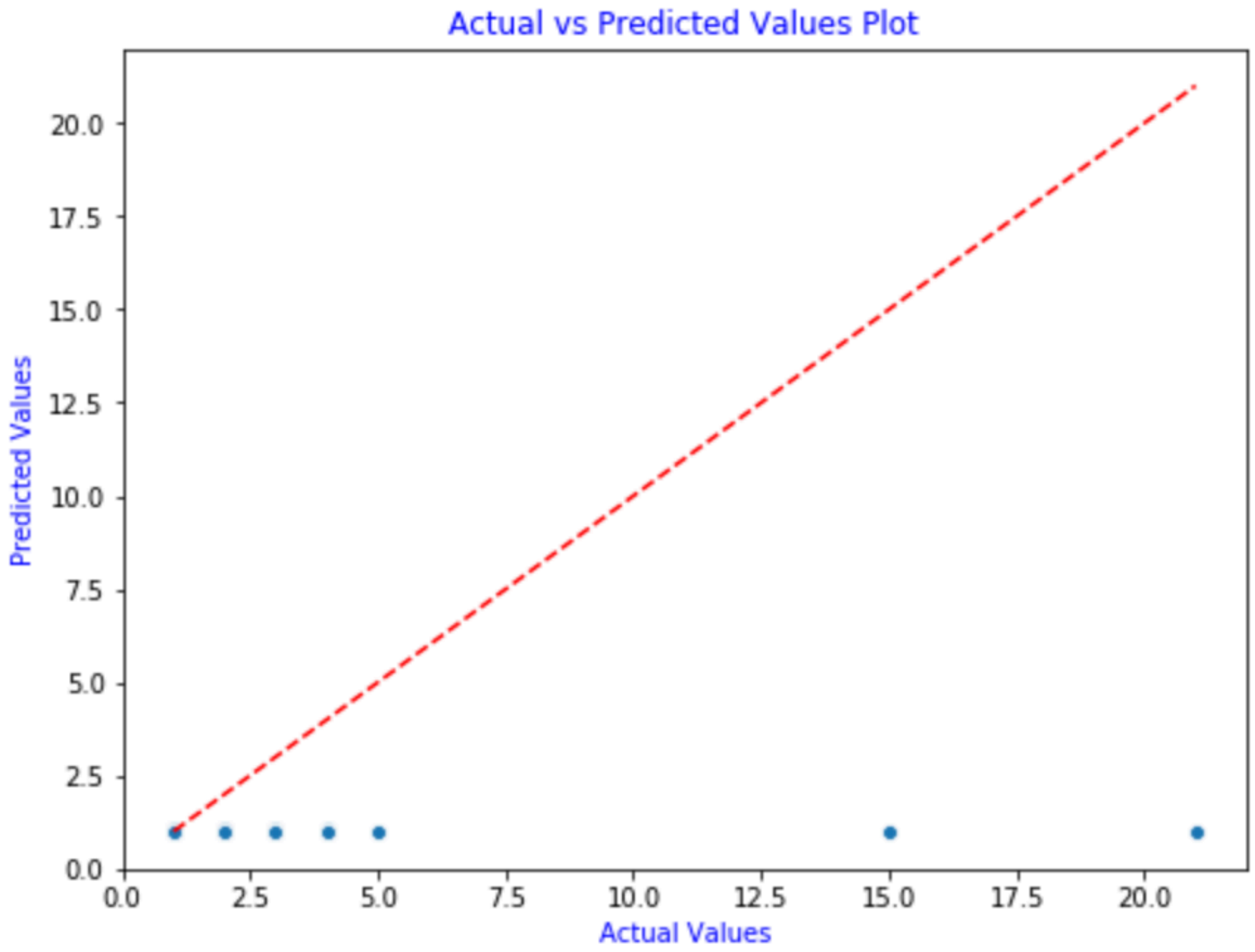
Residual Plot

This plot shows the differences between the actual and predicted values across the different predicted values. If the model is high-performing, the residuals should be randomly scattered around zero. Furthermore, if there is a visible pattern in residuals, this indicates that the model is missing underlying patterns or non-linear relationships in the data.



Actual vs Predicted Scatter Plot

For this plot, we plotted the actual crime_count values on the y-axis and the predicted values on the x-axis to help visualize the relationship and trend between the actual and predicted values. The data points should align along a y = 0.5x line of best fit, where predictions match the actual values relatively closely. Furthermore, if there is significant scatter away from this line of best fit, it would suggest that the model struggles to accurately make predictions.



Analyzing Model Performance

Based on the visualizations and the quantitative metrics, a variety of key insights can be drawn.

Firstly, the very low R-Squared value, the high residual values, and the disparity between the actual and predicted values suggests that the linear regression model that we created is not fully capturing the complexity of the crime count patterns. This could be possible due to a variety of factors such as missing relevant features from the dataset(socioeconomic data, environmental data, etc.) or the model may simply be too simple to holistically capture the complexity of the relationships present in the dataset.

Additionally, it is also important to consider that some of the chosen features in the model (time of day, location, domestic status) may not be relevant features that sufficiently predict crime counts independently and the interactions between the various features need to be explored further to establish valid relationships.

Finally, it is important to note that crime data is inherently influenced by various unpredictable, non-linear patterns that are influenced by socioeconomic and temporal factors. Therefore, a linear regression model may not be the best choice to accurately capture the relationships and interactions between these factors effectively in order to predict the frequency of crime. Furthermore, features such as geographic coordinates may require a more thorough and sophisticated geospatial analysis compared to simple regression.

Next Steps

Based on the analysis of the results of the methods used to derive our model, there are several next steps that should be considered moving forward.

1. Try using Non-Linear models to better capture the complexities of this dataset. Random Forest or Gradient Boosting can help capture the complex interactions and non-linear relationships that may exist in the dataset.
2. Further explore the feature engineering and enrichment pre-processing methods. For example, adding socioeconomic, weather, or event-based data could help provide contextual features that may explain how the crime counts are influenced. Additionally, it is important to explore interaction terms that could potentially capture existing interactions between features, such as time of day and location type.

By exploring the integration of non-linear models and including additional contextual data, we should be able to address the challenges and limitations that were identified during the analysis of the current linear regression model.

Tradeoffs & Strengths & Limitations

Tradeoffs

Linear regression is straightforward to implement and computationally efficient, making it an ideal baseline model to compare against other complex models. This allows easy interpretability which is important in understanding the model's behavior and trends. However, linear regression only works well with linearly related features, which does not hold true for crime data.

Strengths

This model is simple to implement and serves as a baseline model. The foundational equation (linear combination of features) allows us to quantify the impact of each feature on the target variable, offering insights on feature importance.

Limitations

Linear Regression predicts poorly for non-linear relationships and complex interactions. Crime data is inherently complex with many variables to consider with non-linear trends. Additionally, crime data is imbalanced, meaning that model is victim to outlier sensitivity, skewing predictions. Although the evaluation metrics were low, it seems to be overfitting because by examining the visualizations, the model does not capture the data well enough to be generalizable for new testing data. Hence, linear regression is not the best model to predict crime frequency.

Random Forest Classifier

Analyzing Quantitative Metrics

The three quantitative metrics to assess a regression model are Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-Squared. These metrics measure the accuracy of the predictive model of crime frequency.

The mean absolute error is 0.02372, which is quite low in the average residual. In other words, the low MAE indicates closer predictions to actual values.

The mean squared error is also 0.01675 which means despite augmenting the large errors, the mean squared error remains low, indicating that the errors are not that large.

Together, these metrics indicate that the model has relatively small prediction errors overall.

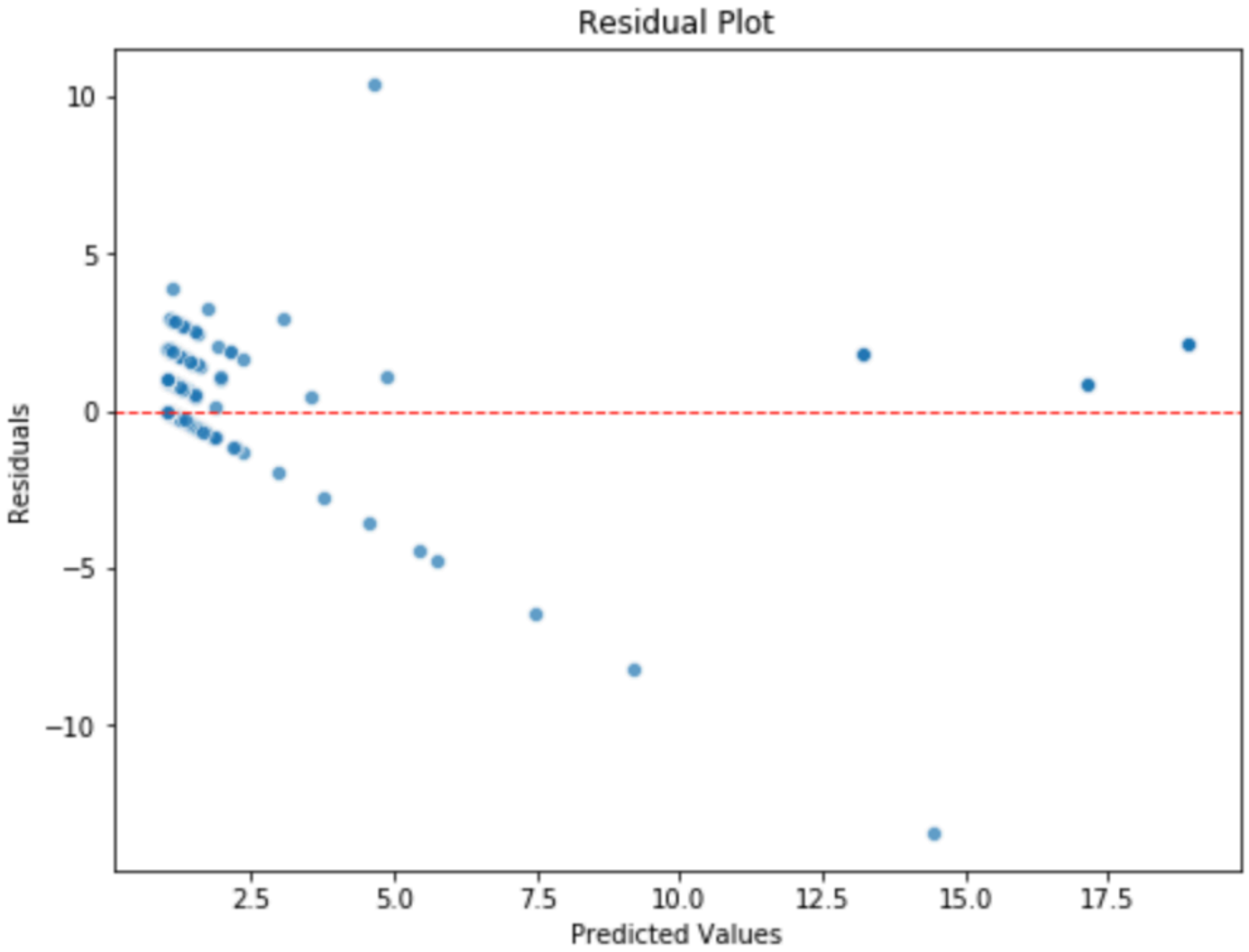
The R-squared value is 0.603. The R-squared value indicates proportion of variance in the predicted values from the Random Forest classifier. If the value is closer to 1.0, it suggests a better fitting model. However, in our case it is 0.60. This suggests that 60% of the variability in the crime data can be explained by the Random Forest model. There is still 40% variability which could be due to unpredictable or unaccounted factors.

The accuracy is 98.51% with a threshold percentage of 0.1 or 10%. This means if the percentage error is less than the threshold, then it is considered accurate. This is pretty high, which can mean the model is performing well in predicting crime counts within a 10% margin of actual values. A high accuracy percentage suggests that the model is effectively capturing the underlying patterns for the majority of predictions. This can also mean that if the dataset is heavily skewed lower or higher crime counts, the model might perform well for dominant cases. Another way of interpreting this metric is that the model can be overfitting to the training data, which might achieve high accuracy on test data but fail to generate new data effectively.

Visualizations

Residual Plot

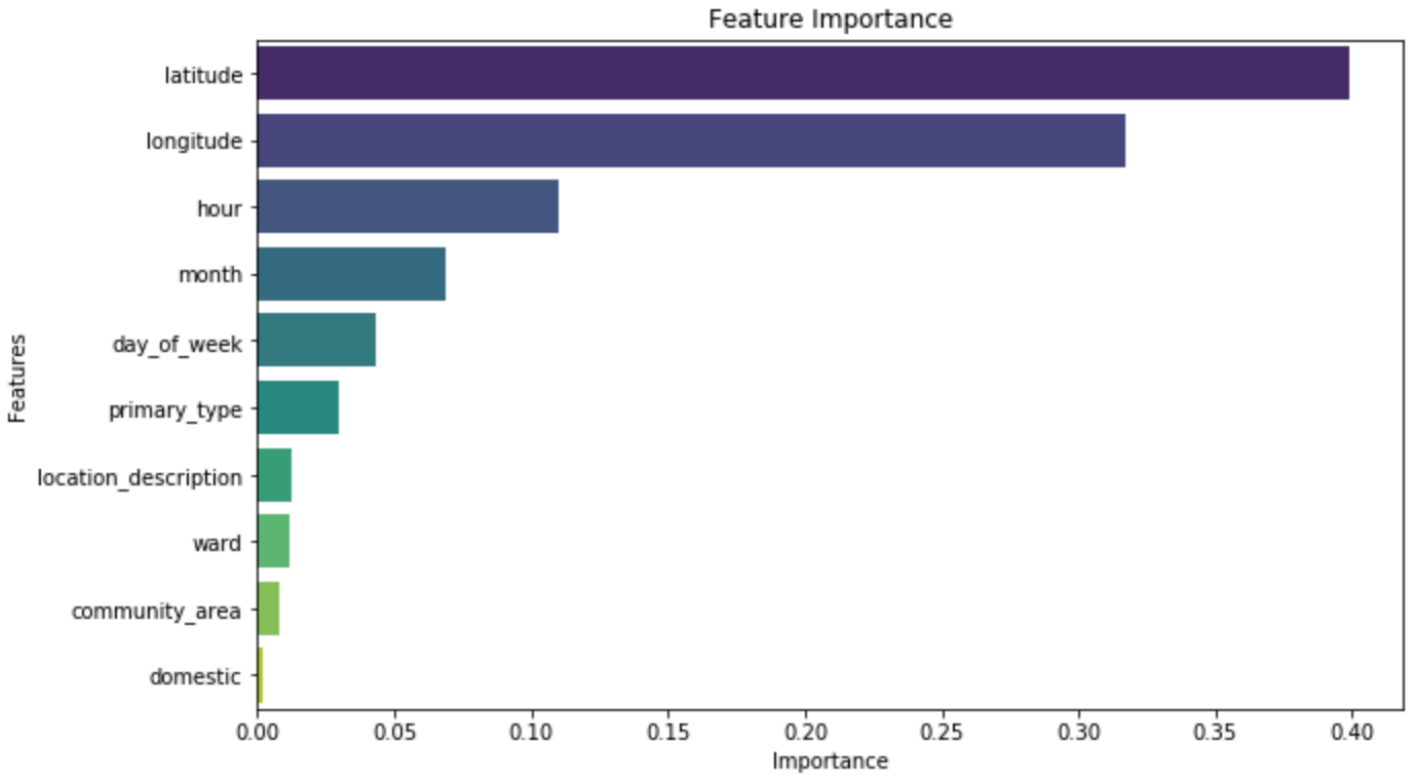
The residual plot shows the differences between the actual and predicted crime count. If the model is high-performing, the residuals should be randomly scattered around a residual of 0. If there is a visible pattern, this indicates the model is not capturing a pattern or there is more complexity in the data. It seems below that there is a visible pattern or cluster of points for low predicted crime count. Hence, this means that the model struggles to accurately predict crime counts in cases the predicted values are low. This clustering could indicate that it is either oversimplifying or failing to capture important relationships in the data. Or, there could be bias in the model. Overall, this means that the model does not fully capture the model, and there are a lot of unknowns to explain this gap.



Feature Importance Plot

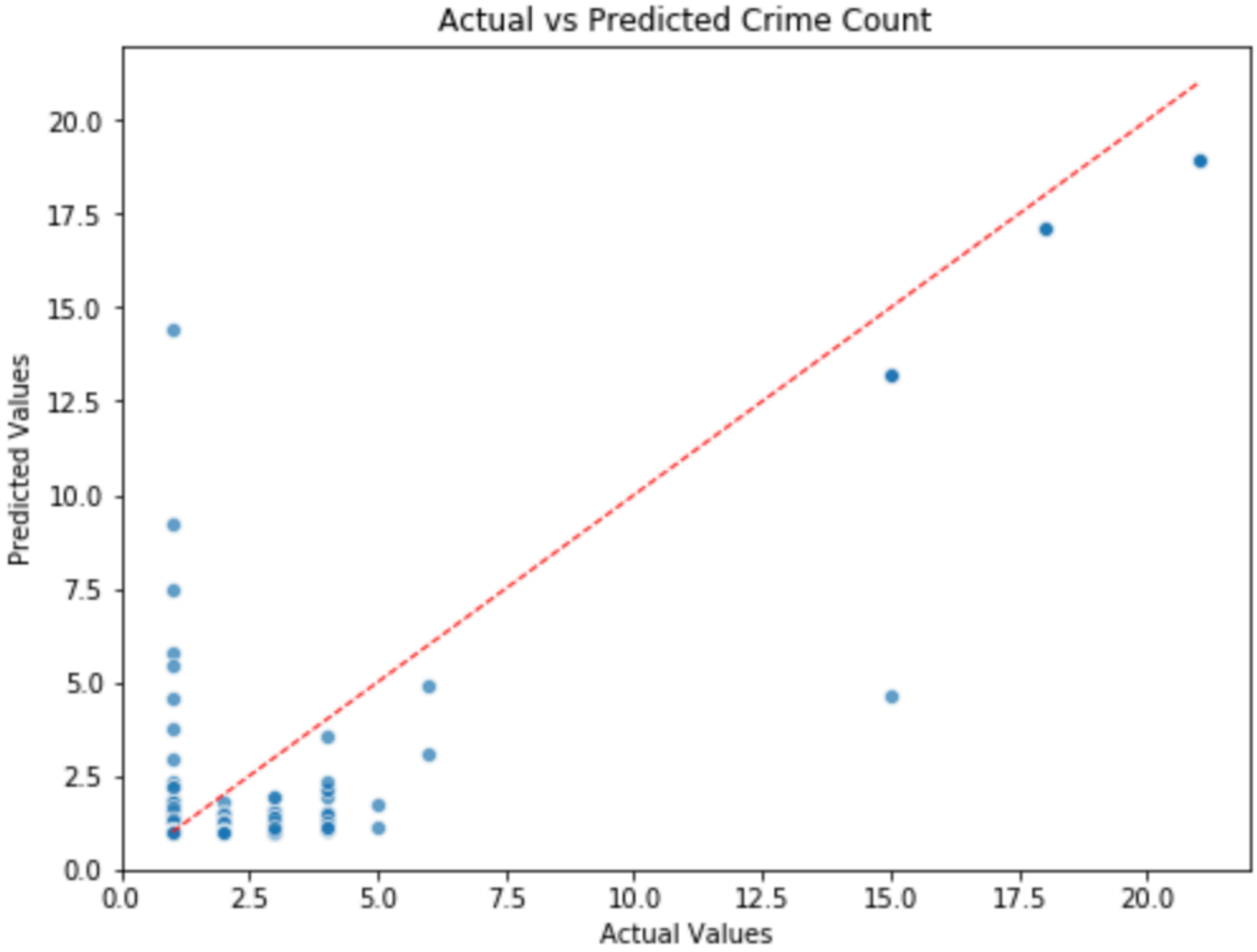
Unlike linear regression, the most important feature is latitude, and then longitude. This makes sense due to the socioeconomic areas and crime frequency of Chicago, which is due to social perception. Then, the third most important feature is the hour. According to the Office of Justice

Programs, the peak is at 9 pm for 18 year olds and older [4]. Now, it is socially known that there is more opportunity for crime at night than day, so that can factor into the prediction. Another mention is month. Chicago captures all seasons, so seasons can also influence crime frequency. There is less opportunity for movement in colder weather than warmer weather [5].



Actual vs Predicted Crime Count Plot

For this plot, we plotted the actual crime count on the y-axis and the predicted crime count on the x-axis to visualize the relationship and trend between the values. One key observation is a visual clustering for lower crime counts, which we saw in the residual plot. There is inherent randomness and non-linear relationships in crime data that adds to the complexity. For example, certain crimes may occur sporadically, defying predictable patterns, or specific locations have unique contextual influences that the model cannot generalize. Perhaps, for lower crime count, it is more unpredictable of what could occur, or it is likely to have more variance.



Next Steps

While the model demonstrates promising results, further improvements can be achieved by fine-tuning the hyperparameters, incorporating additional features, or experimenting with other non-linear capturing models.

Tradeoffs & Strengths & Limitations

Tradeoffs

Random Forest is a flexible model that excels in capturing non-linear patterns between features. However, the computational complexity with the multitude of decision trees comes as a consequence. Additionally, hyperparameters can lead to overfitting, especially the depth of the trees and the number of estimators. Random Forest can be misinterpreted with the top features being the most important when rather they simply have a low entropy or high information gain.

Strengths

This model handles non-linear relationships pretty well since it makes decision trees on the available data rather than confining to a line. It is also robust to noise since it takes consideration of purity score or low entropy, looking at the majority of the data. It also provides feature importance which is a key insight to observe the model's behavior and learn more about the influence of variables.

Limitations

Random forest model is computationally intensive because it creates decision trees on data samples on randomly selected features. The more depth on the trees, the longer it takes to create the decision trees. It also runs parallel processing on all the trees and takes the mean value to create the prediction. However, the model may overfit depending on the hyperparameters, specifically max depth of the decision trees. Fine-tuning the hyperparameters is crucial in achieving optimal performance.

XGBoost

Analyzing Quantitative Metrics (1.1)

As mentioned above the three quantitative metrics primarily used to assess a regression model are Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-Squared. These metrics measure the accuracy of the predictive model of crime frequency.

The mean absolute error is 0.0768, which is quite low in the average residual. In other words, the low MAE indicates closer predictions to actual values.

The mean squared error is also 0.0561 which means despite augmenting the large errors, the mean squared error remains low, indicating that the errors are not that large. Together, these metrics indicate that the model has relatively small prediction errors overall.

The R-squared value is 0.4541. The R-squared value indicates proportion of variance in the predicted values from the XGBoost Model. If the value is closer to 1.0, it suggests a better fitting model. However, in our case it is 0.4541. This suggests that 45.41% of the variability in the crime data can be explained by the XGBoost model. There is still 54.59% variability which could be due to unpredictable or unaccounted factors. This is relatively high compared to the other models.

The accuracy is 89.69% with a threshold percentage of 0.1 or 10%. This means if the percentage error is less than the threshold, then it is considered accurate. This is pretty high, which can mean the model is performing well in predicting crime counts within a 10% margin of actual values. A high

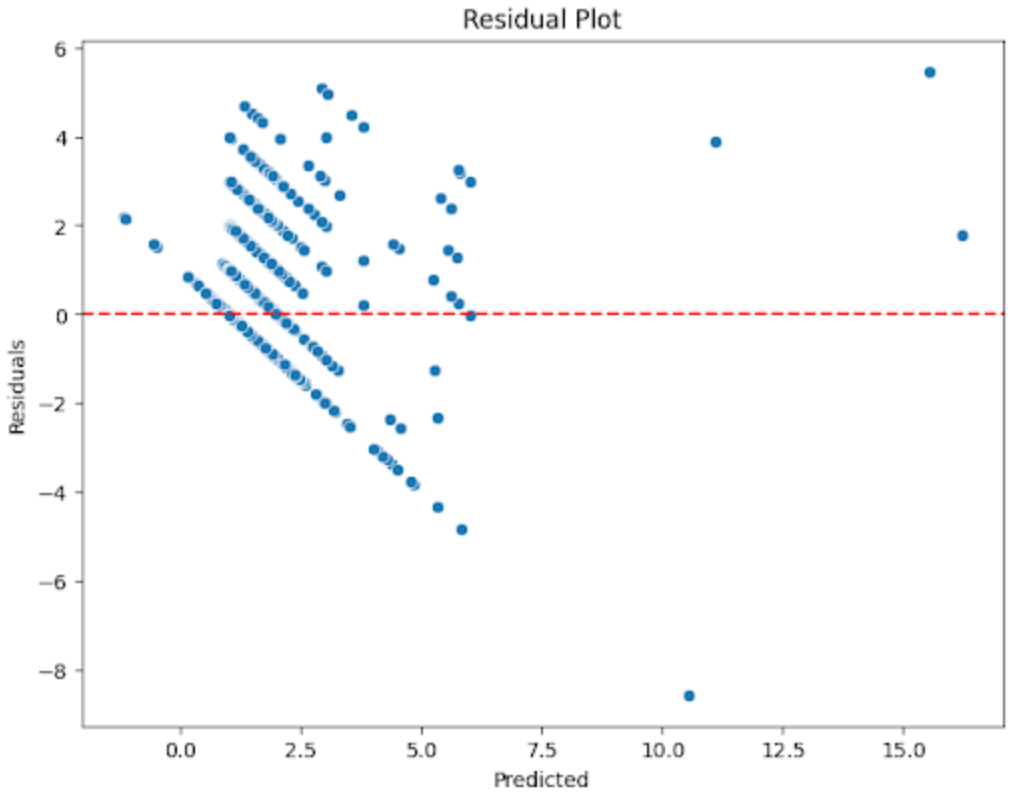
accuracy percentage suggests that the model is effectively capturing the underlying patterns for the majority of predictions. This can also mean that if the dataset is heavily skewed lower or higher crime counts, the model might perform well for dominant cases. Another way of interpreting this metric is that the model can be overfitting to the training data, which might achieve high accuracy on test data but fail to generate new data effectively.

Visualizations (1.2)

Residual Plot

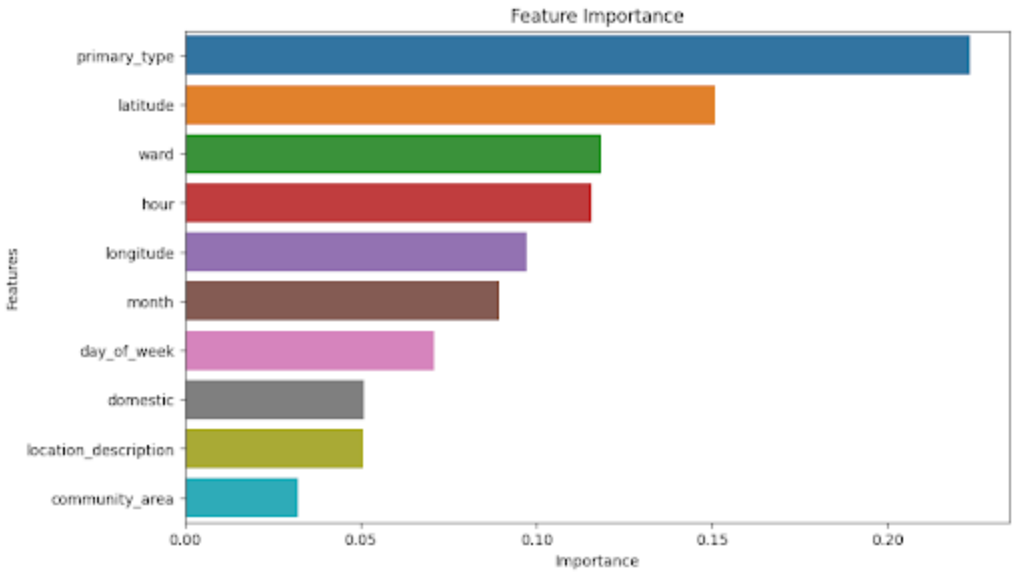
The residual plot shows the differences between the actual and predicted crime count. If the model is high-performing, the residuals should be randomly scattered around a residual of 0. If there is a visible pattern, this indicates the model is not capturing a pattern or there is more complexity in the data.

According to the plot based on the model, it is evident below that there is a visible pattern or cluster of points for low predicted crime count. Therefore, the model struggles to accurately predict crime counts in cases the predicted values are low. This clustering could indicate that it is either oversimplifying or failing to capture important relationships in the data. It could also indicate the presence of potential bias in the model. Overall, this means that the model does not fully capture the model, and there are a lot of unknowns to explain this gap.



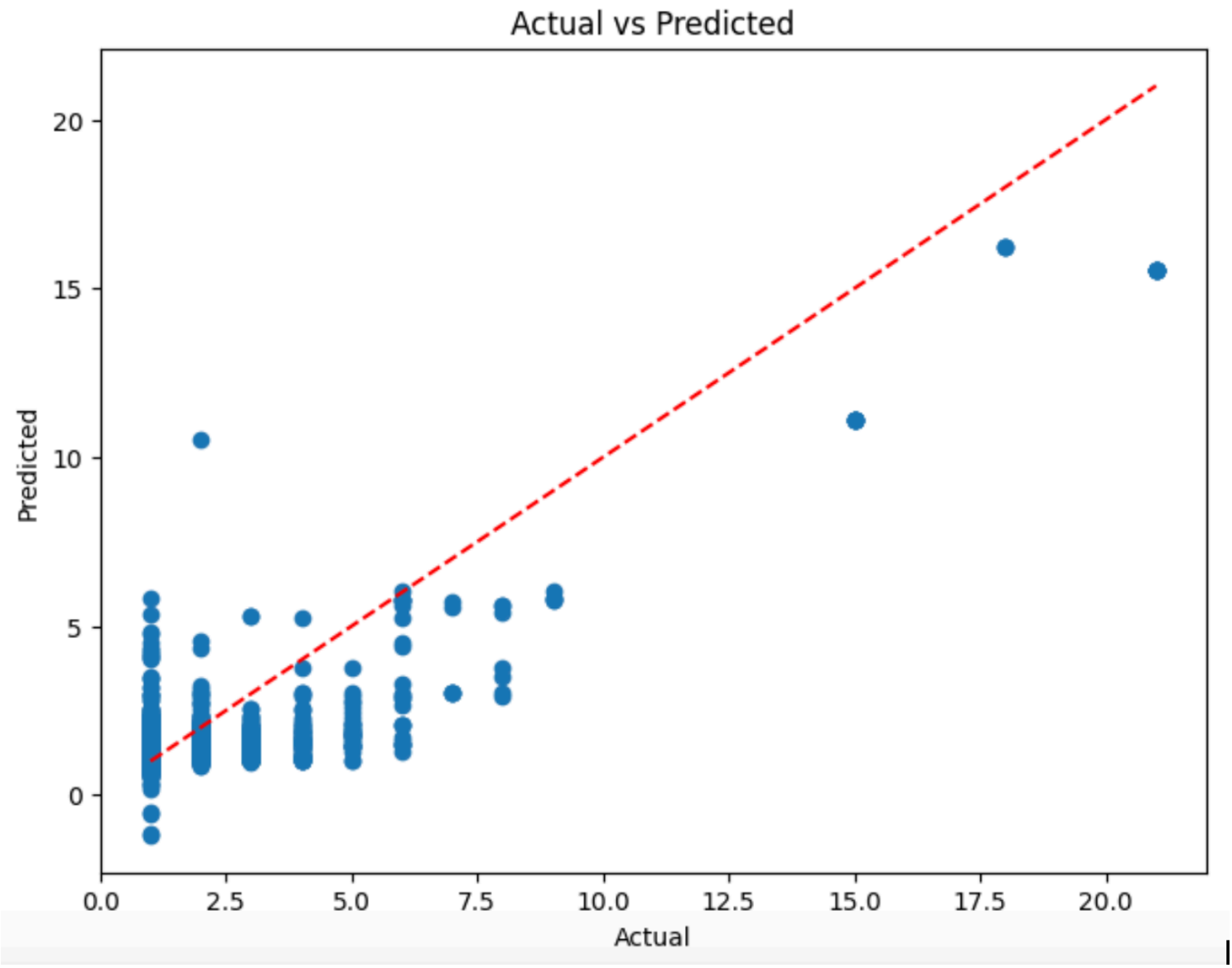
Feature Importance Plot

Unlike Random Forest, the most important feature is primary_type. This makes sense logically because when there is a place with such high density of crime counts, it is important to distinguish and assess the specific type of crime that is occuring. Then, the second and third most important features are the latitude and the ward, respectively. This makes sense to follow each other as the actual coordinates are important, but normally when reporting a crime, once the exact coordinates are established, it is also important to establish the surrounding areas to pinpoint the exact location and context of the occurrence of the crime.



Actual vs Predicted Crime Count Plot

For this plot, we plotted the actual crime count on the y-axis and the predicted crime count on the x-axis to visualize the relationship and trend between the values. One key observation is a visual clustering for lower crime counts, which we saw in the residual plot and also in the residual and actual versus predicted crime count plots from the Random Forest model. There is inherent randomness and non-linear relationships in crime data that adds to the complexity, which means that these results were generally expected. Crimes are naturally known to occur at random even despite several predicting factors simply because crimes are dependent on the psychological state of the individuals who are committing the crime, which varies from person-to-person and cannot be generalized to a predictive model. Perhaps, for lower crime count, it is more unpredictable of what could occur, or it is likely to have more variance.



The residual plot does not seem to be scattered, but it looks like it seems gathered into a cluster, which shows the complexity of the data. So, I decided to log the training and testing data to reduce the skewness of the data. This significantly reduced the evaluation metrics and helped scatter the data more in the residual plot, therefore increasing the accuracy.

As a result, below is the updated analysis of the quantitative metrics and visualizations based on the log transform that was performed.

Analyzing Quantitative Metrics (2.1)

The mean absolute error is now 0.0302, which is quite low in the average residual and lower than the previous MAE. In other words, the low MAE indicates even closer predictions to actual values.

The mean squared error is also now 0.0076 which is significantly lower than the previous MSE calculated before the log transform was performed. This means despite augmenting the large

errors, the mean squared error continues to remain low, indicating that the errors are not large at all.

Together, these metrics indicate the improvement in the model that has evidently relatively small prediction errors overall.

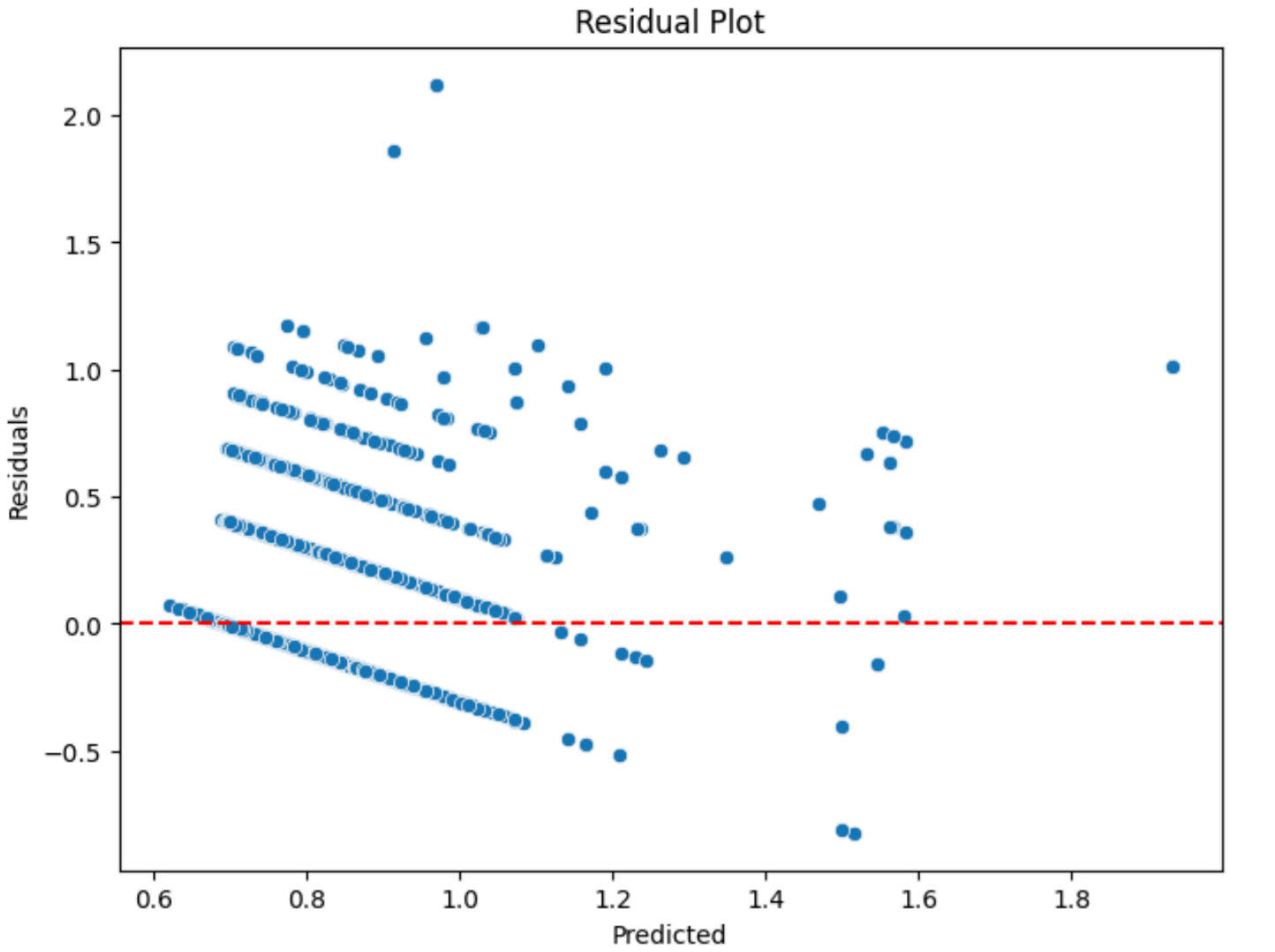
The R-squared value is 0.1386, which is lower than the previous R-squared value. Therefore, this shows that the log transform change did not positively impact the amount of variability the model is able to explain. This suggests that 13.86% of the variability in the crime data can be explained by the XGBoost model. There is still 86.14% variability which could be due to unpredictable or unaccounted factors. This is relatively high compared to the other models and also higher than before the log transform.

However, the accuracy is much higher in this case after performing the log transform with a 94.13% with a threshold percentage of 0.1 or 10%. This means if the percentage error is less than the threshold, then it is considered accurate. This is still pretty high, which can mean the model is performing well in predicting crime counts within a 10% margin of actual values. A high accuracy percentage suggests that the model is effectively capturing the underlying patterns for the majority of predictions. The fact that the accuracy increased by approximately 5% after the log transform indicates the positive change it had on the model.

Visulizations (2.2)

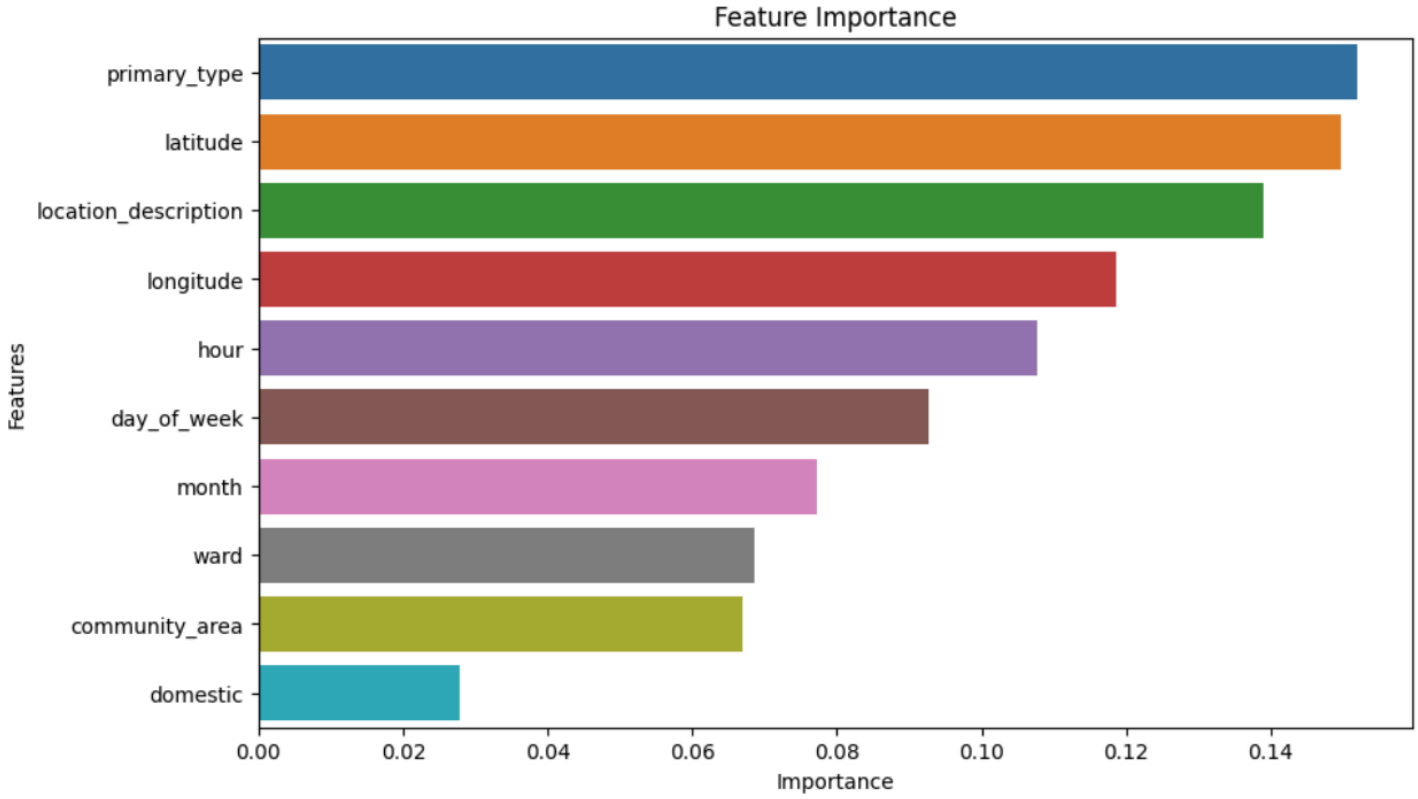
Residual Plot

Here it is evident that the residuals are now more tightly clustered around the zero line, which shows a significant improvement in the model's ability to make accurate predictions across the range of crime counts. Additionally, the scala of the residuals is also reduced, which reflects a more stable and consistent prediction model. Furthermore, the funnel-shaped pattern is less apparent, which demonstrates that the log transform helped to address the fact that the variance of errors in the model was not consistent across all observations. The randomness also suggests that the model better captures the underlying relationships in the data. Also, the extreme residuals are minimized, which shows that the log transform reduces the influence of outliers by compressing their magnitude.



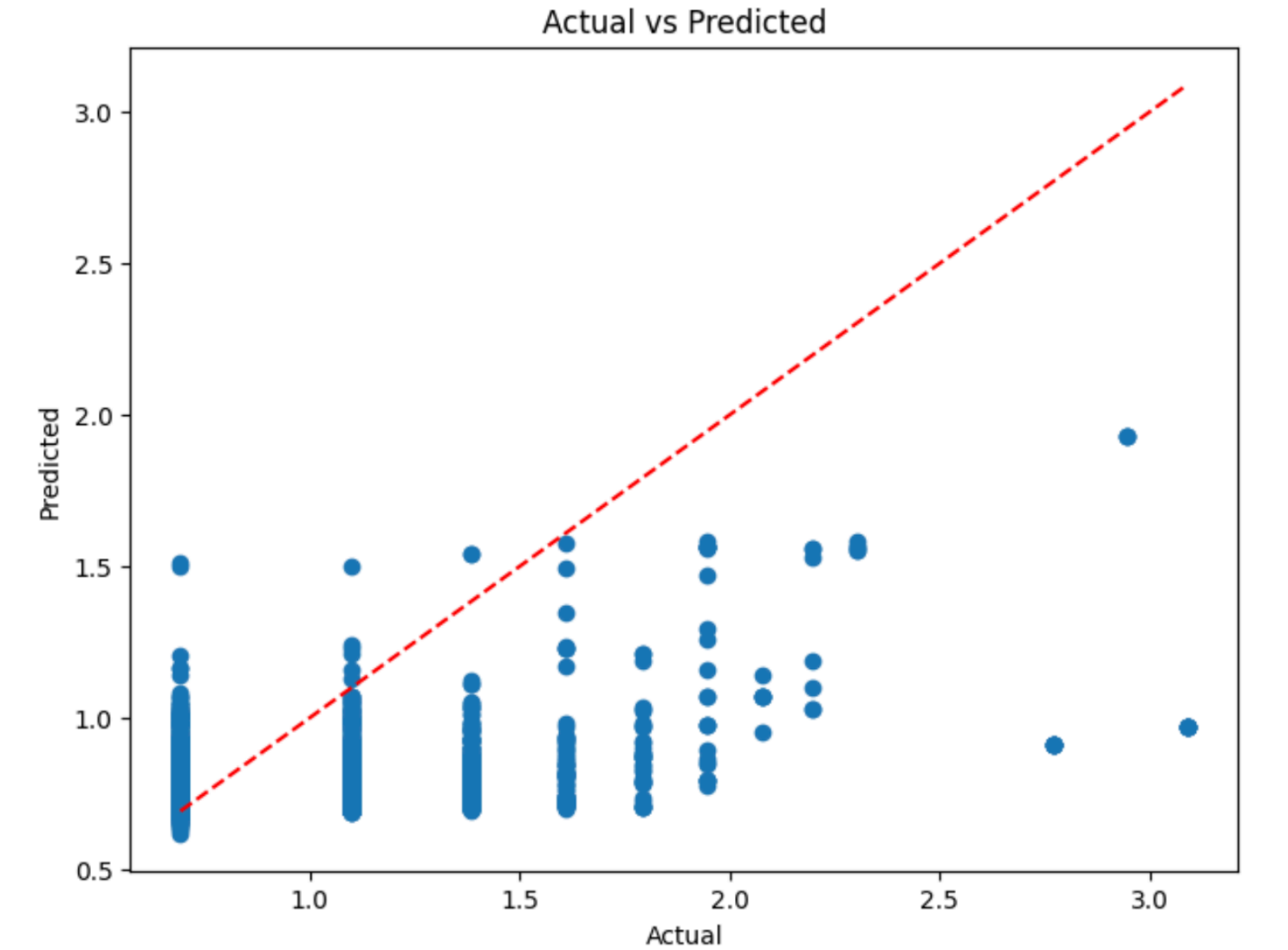
Feature Importance Plot

Here, the primary_type remains the most critical feature but has a reduced relative importance compared to the other features. This indicates that the log transformation reduced the dominance of the categorical variable and distributed the importance more evenly across features. Additionally the latitude feature became significantly more important and became closer in ranking to the primary_type feature. This demonstrates that geospatial aspects are better captured after the log transformation. The location_description also moved up in importance, which suggests that the model is now better able to extract insights from location descriptions after it was able to reduce the skewness of the data. Furthermore, the temporal features such as hour and day_of_week have increased in importance, which is likely due to the fact that the log transformation allowed the model to better capture patterns in the dataset that are associated with time.



Actual vs Predicted Crime Count Plot

Here, it is evident that after the log transformation, the predictions are much closer to the diagonal, which represents the equality between actual and predicted values. This shows that there is an improved prediction accuracy especially for lower crime counts, which were previously scattered before the log transformation was performed. The transformation compressed the scale of the higher crime counts, which helped the model to better generalize the overall range of the data and as a result, outliers have less influence on the overall predictions. Furthermore, the points also align more consistently with the line of equality, which suggests that the model is capturing a more linear relationship after the log transformation was performed. Also, the log transform helps to address the inherent skewness and randomness in the data, which leads to predictions that are more proportional to the actual values.



Next Steps

Similar to the Random Forest model, XGBoost is more sensitive to hyperparameters that greatly influences the strength of the model. Thus, experimenting with hyperparameters to increase R^2 value can be the next step since XGBoost is pretty powerful already. Additionally, we can run diagnostics on the data to understand the behavior of the data on a lower level, which can offer insights in improving our preprocessing methods.

Tradeoffs & Strengths & Limitations

Tradeoffs

Despite the relatively high accuracy and positive quantitative metrics, there are quite a few tradeoffs to be considered when evaluating the XGBoost model.

For example, XGBoost is computationally expensive due to its approach in iterative boosting where the weak learners are sequentially refined. Especially in the context of this large crime dataset, training the model will take longer and require more computational space compared to the more simpler models such as Linear Regression or Random Forest. Therefore, while the model does provide high performance, the increase in computation requirements may limit the scalability of the model for larger datasets such as this one.

Furthermore, XGBoost is very sensitive to hyperparameters because there are so many hyperparameters such as learning rate, max depth, etc. that require extensive tuning. If the hyperparameters are not optimized, the model may underperform or run the risk of overfitting. In this case especially, XGBoost has a high risk of overfitting since the dataset has noise and potentially irrelevant features. Crime data is inherently noisy and contains a variety of unpredictable factors that might cause the model to fit too closely to the training data and perform low on unseen data.

Strengths

XGBoost Model performs well on high-dimensional data due to continual refinement on previous weak learners. It also incorporates regularization and handles some of the preprocessing within the model.

Limitations

There are a lot of hyperparameters to tune, which means the accuracy is dependent on the hyperparameters more than the model. The hyperparameters also influence the computation intensity. Hence, there needs to be experimentation with the hyperparameters to retrieve optimal performance.

Key Insights

Model Performance

1. Random Forest achieved the highest R^2 and accuracy, showcasing its strength in capturing this particular nonlinear complex dataset.
    1. Possibly due to the hyperparameter tuning
2. XGBoost performed comparably with a slightly lower R^2 and accuracy, which is again heavily dependent on the hyperparameter tuning or could be due to the method of ensemble learning
3. Linear Regression greatly struggled to explain the variance and capture the complex patterns, highlighting the limitations of the model with complex or realistic datasets like crime data in

this stochastic world.

4. XGBoost and Random Forest are both decision tree algorithms, and XGBoost is powerful in many ways. However, Random Forest performed the best out of the three models in both accuracy and capturing the complex patterns of the model. Random Forest uses Bagging as its ensemble learning. XGBoost uses Gradient Boosting for its ensemble learning method.

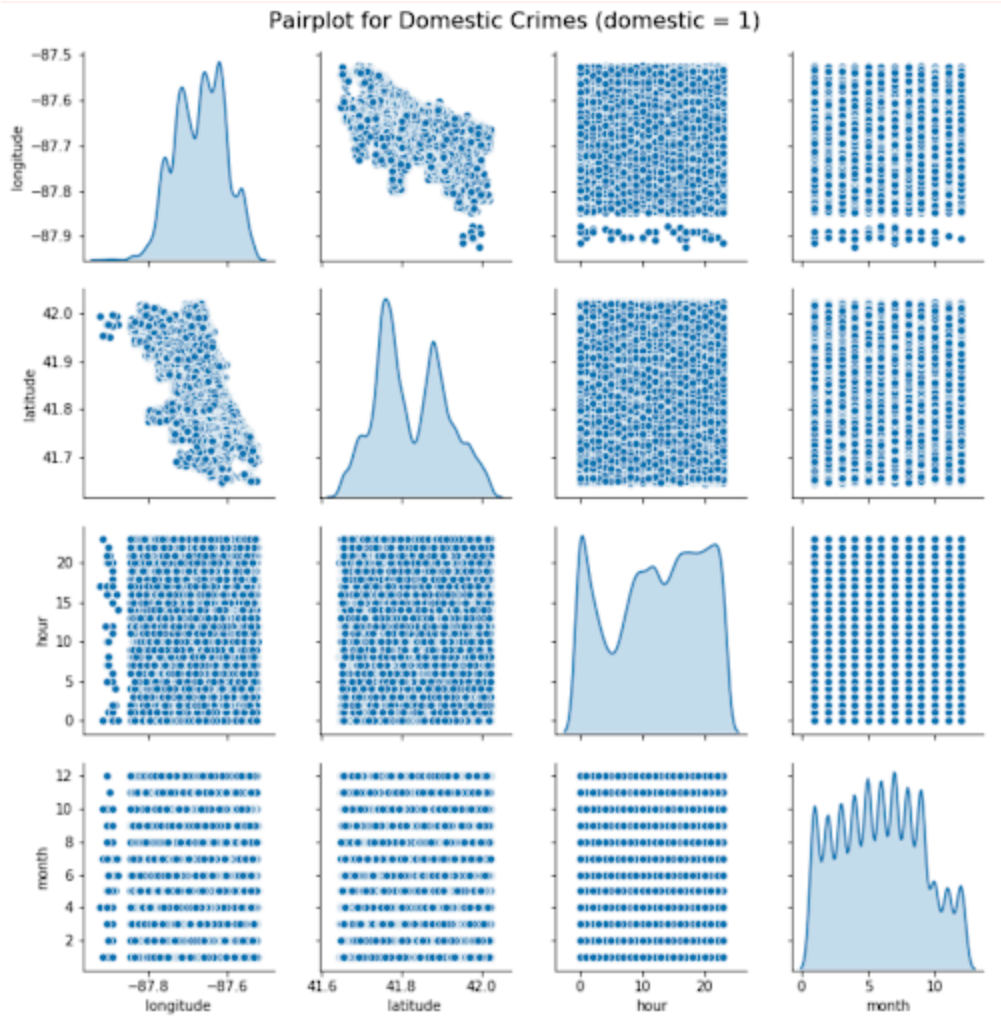
Feature Importance

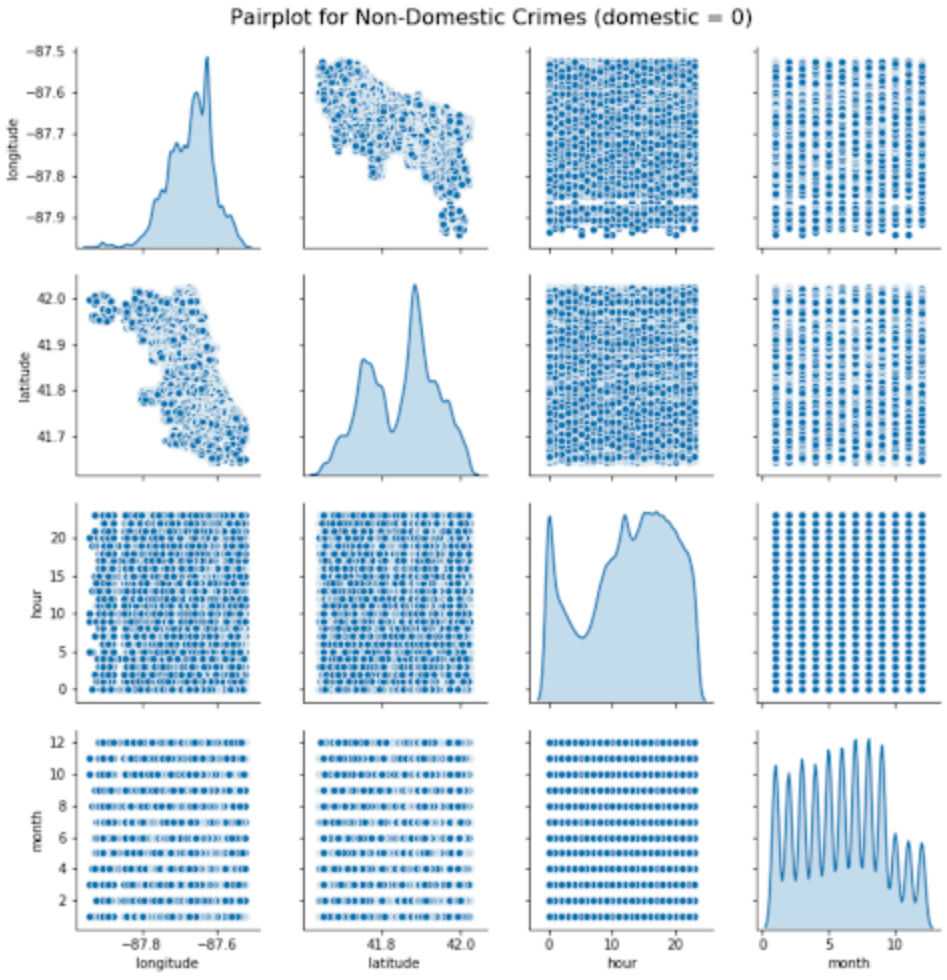
1. Spatial features (latitude, longitude) dominated across models, reflecting the geospatial nature of crime.
2. Temporal features (hour, month) highlighted patterns in crime occurrence, such as nighttime crimes or seasonal variations, affecting the crime frequency/count.
3. Interesting Takeaway: Each model had different order of feature importances, reflecting the behavior of the model.

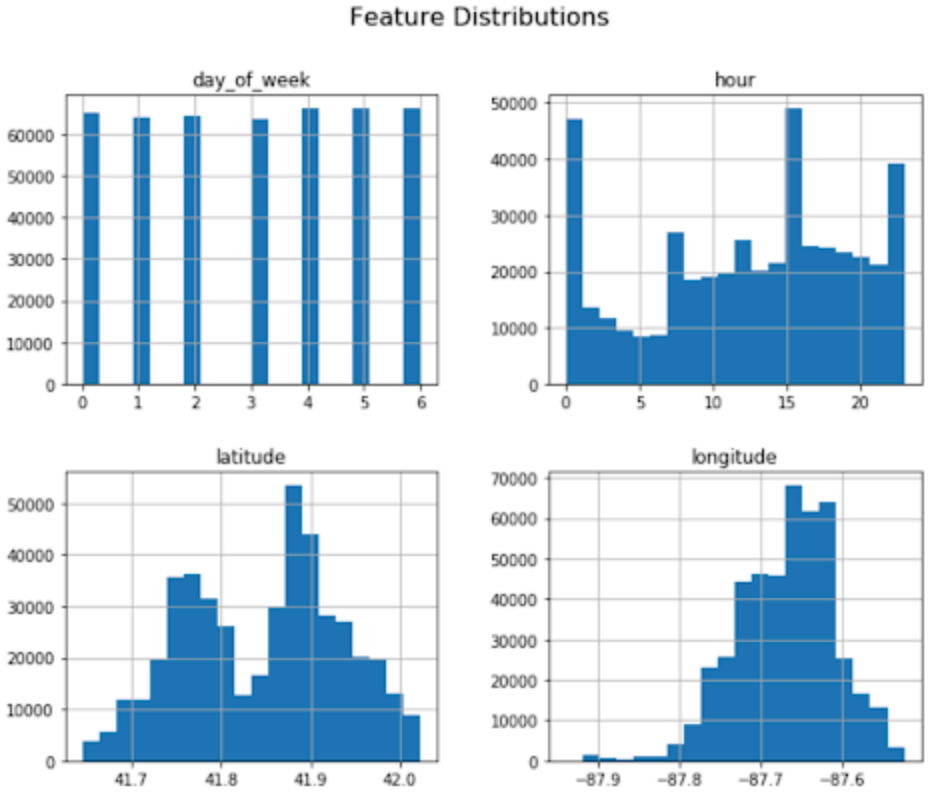
Data Complexity

The models revealed the inherent randomness and non-linear relationships in crime data and within each feature (as seen in the Pair Plots). Low crime counts, in particular, posed challenges in predicting accurately due to the unpredictability with visible clustering in residual plots. There is only so much we can do to balance the imbalance and unpredictable crime data, which is reality in itself.

Pairplots For Each Feature


Pairplot for Domestic Crimes (domestic = 1)
longitude
latitude
hour
month
−87.5
−87.6
−87.7
−87.8
−87.9
42.0
41.9
41.8
41.7
20
15
10
5
0
12
10
8
6
4
2
−87.8
−87.6
41.6
41.8
42.0
0
10
20
0
5
10


Pairplot for Non-Domestic Crimes (domestic = 0)
longitude
latitude
hour
month


Feature Distributions
day_of_week
hour
latitude
longitude

Next Steps

1. Improve Feature Engineering
    1. Incorporating socio-economic data, weather patterns, and event-based features either by joining other corresponding datasets with this crime one can enhance the predictability or cause more overfitting
2. Fine-tune hyperparameters with the non-linear models (Random Forest Model & XGBoost)
    1. Implement k-fold cross-validation to optimize hyperparameters instead of guessing and checking values and assuming relationships between hyperparameters
3. Experiment with Alternative Non-Linear Models
    1. Neural Networks can capture deep, non-linear patterns with hidden layers and a selection of different activation functions
    2. Ensemble Random Forest and XGBoost together despite the different types of ensemble learning used with Decision Trees
4. Try another problem statement with a focus on geospatial data (i.e. classification problem)
    1. Geospatial features were dominant in feature importance plots, so we could focus on finding hotspots of crime given features or if a crime will occur or not.
    2. Evaluate using classification metrics like F1 score (precision & recall) and confusion matrix

References

[1]	A. Data, “Predicting crime with AI: Navigating ethical issues,” Ayadata, https://www.ayadata.ai/predicting-crime-with-ai-navigating-ethical-issues/#:~:text=In%20June%202022%2C%20a%20cross,of%20around%201000ft%20(300m) (accessed Oct. 4, 2024).
[2]	M. Wood, “Algorithm predicts crime a week in advance, but reveals bias in police response, Biological Sciences Division | The University of Chicago, https://biologicalsciences.uchicago.edu/news/algorithm-predicts-crime-police-bias (accessed Oct. 4, 2024).
[3]	E. Meyer, “Uncovering new factors in Chicago crime through Regression models-part Two,” Medium, https://levelup.gitconnected.com/uncovering-new-factors-in-chicago-crime-through-regression-models-part-two-d76860664e2e (accessed Oct. 4, 2024).
[4]	Office of Juvenile Justice and Delinquency Prevention, “Comparing Offending by Adults & Youth,” Violent crime time of day (per 1,000 in age group), https://www.ojjdp.gov/ojstatbb//offenders/qa03401.asp (accessed Nov. 30, 2024).

[5]	C. R. Block, “Is Crime Seasonal?” Bureau of Justice Statistics, Chicago, 1984 (accessed Nov. 30, 2024).

Gantt Chart (Updated Phase 3)

Here is the link to access: Gantt Chart Fall 24.

Contribution Table

Name	Contributions
Agnes Chacko	Model #2 Preprocessing; Presentation; Recording
Harish Viswanathan	Model #3 Preprocessing; Model #3 Model Selection; Model #3 Results & Discussion
Nikhil Surapaneni	Model #3 Preprocessing; Presentation; Recording
Rhea Garg	Model #2 Preprocessing; Model #2 Model Selection; Model #2 Algorithm Coding; Presentation
Ria Patel	Model #2 Results & Discussion; Model #3 Model Selection; Model #3 Algorithm Coding; Model Comparison; Recording; GitHub Report Configuration