Polynomial Interpolation and K-Mean Mining to Predict Crime Rates

Anish Goyal

The Gwinnett School of Mathematics, Science, and Technology

SciEng Research Project

Mr. Nguyen

April 29, 2022

Purpose

I. The purpose of this project is to test the accuracy of various supervised and unsupervised machine learning algorithms on a previous Atlanta crime data set, which is publicly available on the Atlanta Police Department website, to see the applications of machine learning methods on other cities around the country. Effective patrols in crime-heavy areas can be established when the best prediction model is used to determine the locations with the greatest crime rates. This can be applied to society because the police force can routinely deal with crimes before they ever occur on a daily basis, as their mere appearance is often sufficient to prohibit crimes from happening. Ultimately, as long as useful data is provided into the model—assuming that the best machine learning algorithm is selected, which is what I am trying to determine—the overall rate of crime will continue to fall. In this project, K-Means clustering, decision trees, random forests, Naive Bayes classifiers, and linear regression models will be conducted, and to test the accuracy of these machine learning algorithms, I will compare their projected results with the expected results in the data set of the year after. After determining which machine learning algorithm has the most accuracy, I will make a program with Python that will give forensic analysts the foresight they need to predict future crime occurrences using an Excel data set.

   A. The accuracy of a machine learning algorithm depends on factors such as the number of points present in the data set, the amount of computation, and what one wants to do with the data (Li, 2020).

B. The Principal Components Analysis (PCA) method of dimensionality reduction is useful for lessening the number of features present in a data set to find a distinct relationship between the data points present (Kaloyanova, Ganchev, & Guide, 2019).

C. Although decision trees will be used to decrease the amount of points with the threshold number of standard deviations in a reduction model, they can also be used to identify which clusters are active hotspots (Sayad, 2012).

II. Hypothesis - Out of all of the supervised and unsupervised machine learning methods I am testing, the algorithm that will provide the most accuracy and feasibility when compared to the future data set is decision trees.

A. Decision trees do not require much computational power, and it processes linear and non-linear data very quickly with accuracy (*Modern Machine Learning Algorithms: Strengths and Weaknesses*, 2017).

B. The Naive Bayes classifiers could end up being the most accurate yet prove unrealistic to implement because it assumes the mutual exclusivity of variables (Saini, 2021). Further testing will be conducted during this project in order to see whether Naive Bayes is an effective method for predicting crime rates, especially for locational clustering.

C. Null hypothesis - Random forests do not provide the most accuracy when compared to other machine learning algorithms and are difficult to process and implement.

III. Literature Check

A. An overview of machine learning algorithms and the science behind them

1. What are machine learning algorithms?
    a) Machine learning algorithms are pieces of code that assist individuals in exploring, analyzing, and deducing meaning from large amounts of data. Each algorithm is a finite set of clear, step-by-step instructions that a machine can use to accomplish a certain goal. The purpose of a machine learning model is to find or develop patterns that people can use to make predictions or categorize data (*Machine Learning Algorithms*, 2020).
2. How does machine learning work?
    a) Machine learning algorithms rely on parameters derived from training data, which is a subset of the broader set. The system produces increasingly accurate findings as the training data expands to more properly represent the environment. Data is analyzed in a variety of ways by different algorithms. They're frequently divided into three categories based on the machine learning methodologies they employ: supervised learning, unsupervised learning, and reinforcement learning. To anticipate target categories, detect unexpected data points, predict values, and discover commonalities, programs usually employ regression and classification (Saxena, 2019).

B. Specific details about each machine learning algorithm being tested

1. K-means clustering
    a) K-Means clustering groups data into clusters based on their similarities and differences with elements in other categories. The letter 'K' represents the number of clusters and the system must be informed of the number of clusters needed for the data set. The algorithm is able to compute the optimal number of clusters for the data set and place their centroids as close as possible to the data clusters using Euclidean distance (Kumar, 2021a).
2. Decision trees
    a) By learning simple decision rules derived from data attributes, decision trees generate a predictive model for the value of a target variable. Generally, decision rules are expressed as if-then-else statements. Data is represented as a "tree" of hierarchical branches that are created until they get to leaves," which are the predictions made from the data. Decision Trees easily model non-linear relationships because of their branching structure. They can be represented graphically and easily understood by nonprofessionals.(Kumar, 2021b).
3. Random forests
    a) Random forest is a machine learning technique that employs a large number of individual decision trees. The

ideal split for each node is determined using a set of randomly generated candidate variables during the tree-building process. Random Forest can be used to choose crucial variables and groups, as well as to gain a better knowledge of variable relationships, in addition to predicting the outcome of classification and regression analysis. Decision trees are easy and simple to use, yet they are inaccurate. When utilized with the training data that was used to create them, decision trees are quite effective, but they aren't flexible when it comes to categorizing the fresh sample. It means that during the validation process, the reliability is really low. It arises as a result of a process known as over-fitting, in which a model analyzes the training data to the point where it negatively affects the model's performance on fresh data (Kumar, 2021c).

4. Naive Bayes classifiers
    a) Naive Bayes is a simple machine learning technique that employs Bayes' theorem and assumes strong independence constraints among features to obtain results. The algorithm is "naive," since it presupposes that each input variable is mutually exclusive of one another and makes it unrealistic to implement in real life; however, it turns out that the algorithm is capable of solving a multitude of complex

problems and does not require a large training data set in order to function. (Kumar, 2021d).

5. Linear regression

    a) Linear regression looks for correlations between variables. It's used to figure out if and how one phenomenon affects another, or how numerous factors are linked. The coefficient of determination, abbreviated as R^2, indicates how much variance in y can be explained by the dependency on x when a specific regression model is used. A higher R^2 implies a better fit, implying that the model can explain the fluctuation of the output with diverse inputs better. Linear regression involving more than two independent variables is known as multivariable linear regression. If just two independent variables are present, the predicted regression function is $f(x_1, x_2) = b_0 + b_1x_1 + b_2x_2$. In three-dimensional space, it depicts a regression plane. The purpose of regression is to find values for the weights b0, b1, and b2 so that the generated plane is as close to the real values as possible (Stojiljković, 2019).

Form 1A Research Plan Attachment: Engineering

Team Leader Anish Goyal

Team Member(s) Anish Goyal

Title of Project Polynomial Interpolation and K-Mean Mining to Predict Crime Rates

Rationale

Atlanta has a 108% higher crime rate than the national average. It is within the top 3% most hazardous cities in the United States with approximately 30,000 crimes annually and a 61% crime rate per capita ("Atlanta crime rates," 2019). Given that Atlanta is such a high-profile city with its large number of crimes, the police department cannot respond to each case individually without wasting excess human resources. They must be led to high-crime locations ahead of time. Effective patrols in crime-heavy areas can be established when the best prediction model is used to determine the locations with the greatest crime rates. This can be applied to society because the police force can routinely deal with crimes before they ever occur on a daily basis, as their mere appearance is often sufficient to prohibit crimes from happening. This is where machine learning comes in. Using various machine learning algorithms on released data for past crime occurrences can provide valuable insight as to how crimes are distributed geographically, and which areas are the most crime heavy. Ultimately, as long as useful data is provided into the model—assuming that the best machine learning algorithm is selected, which is what I am trying to determine—the overall rate of crime will continue to fall.

Engineering Goal(s)

I plan to create a machine learning program that uses supervised and unsupervised algorithms coded in a python3 IDE called Anaconda Navigator. The purpose of the machine learning program is to predict and visualize the occurrences of different crime types for each neighborhood in Atlanta for the year 2019 after learning from crimes in the years 2009-2018 and comparing which algorithms are the most accurate with the actual crime counts from 2019 using the Atlanta PD's released COBRA datasets. The accuracy of the results will be a percentage measured by taking the average of the percent differences of each of the crime types for a neighborhood against the actual outcomes. This is needed because identifying clusters of high criminal activity permits the Atlanta PD to assign optimized police patrol routes and other crime-prevention measures like street cameras and neighborhood watches in the areas that need them the most, which allows the police department to efficiently allocate human resources while preemptively stopping large amounts of crime from occurring.

PROOF OF CONCEPT

Independent Variable

I will test each trial with a different type of machine learning algorithm that I deemed would be the most appropriate to use in this experiment given the information I retained from my literature check and prior knowledge. I will be using three supervised algorithms—K-means clustering, PCA (Principal Component Analysis) dimensionality reduction, and Naïve Bayes classifiers—and three unsupervised algorithms—decision trees, random forests, and linear regression—to predict the crime occurrence of each crime for each neighborhood and return an overall accuracy for that neighborhood using the prediction in comparison to the actual outcomes from 2019. Because my IV is categorical, it does not have units, but it is measured based on the accuracy it returns for the type of algorithm used.

Dependent Variable

- Crime occurrence (1st DV) is measured as a percentage and is defined as the predicted chance of a particular crime occurring in a particular neighborhood. It will be measured with multiple different supervised and unsupervised algorithms that will each return a different crime occurrence.
- Accuracy (2nd DV) is measured as a percentage and is defined as the total resemblance of the predicted crime occurrence (for the year 2019) with the actual crimes that occurred. This DV will be evaluated using the percent difference for the crime occurrences of each neighborhood, adding them, and taking the average. The percent differences for each individual neighborhood will likely also be measured and assessed.

If I am only allowed to report **one** dependent variable for this experiment, I would say the **accuracy** DV more or less takes precedence over the crime occurrence DV, as the crime occurrence DV is used for making internal predictions in the program and, overall, I am measuring the accuracy between the various algorithms that are applied to the data.

Controlled Variables/Constants

- Cleaned COBRA data sets
  - Time period of the csv file that will be used to train the model (2009-2018)
  - Time period of the csv file that will be used to analyze the predictions of the trained model in accordance with the type of algorithm used (2019)
- Shapefile
  - The shapefile that will be used to produce the visualizations of the crime scores or severities in each neighborhood across Atlanta will remain the same.
- Programming language used (python3)
  - Python is a very simple programming language that has many data science packages available to the user. I have extensive prior knowledge of Python in terms of data science have even used it before for robotics and other extracurriculars.
- Crime severity categorization (e.g., Categories 1-4)
  - Each crime category is associated with certain types of crimes, depending on

PROOF OF CONCEPT

Independent Variable

I will test each trial with a different type of machine learning algorithm that I deemed would be the most appropriate to use in this experiment given the information I retained from my literature check and prior knowledge. I will be using three supervised algorithms—K-means clustering, PCA (Principal Component Analysis) dimensionality reduction, and Naïve Bayes classifiers—and three unsupervised algorithms—decision trees, random forests, and linear regression—to predict the crime occurrence of each crime for each neighborhood and return an overall accuracy for that neighborhood using the prediction in comparison to the actual outcomes from 2019. Because my IV is categorical, it does not have units, but it is measured based on the accuracy it returns for the type of algorithm used.

Dependent Variable

- Crime occurrence (1st DV) is measured as a percentage and is defined as the predicted chance of a particular crime occurring in a particular neighborhood. It will be measured with multiple different supervised and unsupervised algorithms that will each return a different crime occurrence.
- Accuracy (2nd DV) is measured as a percentage and is defined as the total resemblance of the predicted crime occurrence (for the year 2019) with the actual crimes that occurred. This DV will be evaluated using the percent difference for the crime occurrences of each neighborhood, adding them, and taking the average. The percent differences for each individual neighborhood will likely also be measured and assessed.

If I am only allowed to report **one** dependent variable for this experiment, I would say the **accuracy** DV more or less takes precedence over the crime occurrence DV, as the crime occurrence DV is used for making internal predictions in the program and, overall, I am measuring the accuracy between the various algorithms that are applied to the data.

Controlled Variables/Constants

- Cleaned COBRA data sets
  - Time period of the csv file that will be used to train the model (2009-2018)
  - Time period of the csv file that will be used to analyze the predictions of the trained model in accordance with the type of algorithm used (2019)
- Shapefile
  - The shapefile that will be used to produce the visualizations of the crime scores or severities in each neighborhood across Atlanta will remain the same.
- Programming language used (python3)
  - Python is a very simple programming language that has many data science packages available to the user. I have extensive prior knowledge of Python in terms of data science have even used it before for robotics and other extracurriculars.
- Crime severity categorization (e.g., Categories 1-4)
  - Each crime category is associated with certain types of crimes, depending on

        - Year
            - The day of the year that the crime occurred in since the year 2000
            - This parameter will only be passed to the 2009-2018 COBRA data set, as all of the data from the 2019 data set is from the same year and will only be used for comparing the results of the predictions from the previous years
        - Latitude
            - The precise latitude where the crime occurred.
        - Longitude
            - The precise longitude where the crime occurred.
            - Knowing both longitudinal and latitudinal values are useful for establishing possible predictors by location or for post-processing analysis
        - Crime category
            - A number associated with the severity of the crime occurred.
            - Smaller numbers indicate a high severity, while bigger numbers indicate a minimal severity.
            - Homicide and manslaughter = 1
            - Aggravated assault, pedestrian robbery, commercial robbery, and residential robbery = 2
            - Residential burglary, nonresidential burglary, and auto theft = 3
            - Vehicular and nonvehicular larceny = 4
        - Category 1*
            - The number of crimes that occurred in category 1 on a particular day
        - Category 2*
            - The number of crimes that occurred in category 2 on a particular day
        - Category 3*
            - The number of crimes that occurred in category 3 on a particular day
        - Category 4*
            - The number of crimes that occurred in category 4 on a particular day

Inputs are for supervised algorithms only (i.e., decision trees or random forests)

- Outputs (information being fed out of the model)
    - Visualization of crime occurrences
        - A plot of the crime scores for every neighborhood in Atlanta; darker shaded neighborhoods are more crime-heavy areas with a larger crime score
        - One visualization will be made for each crime category to see whether the distribution of particular crime types remain the same across multiple neighborhoods
        - The supervised/unsupervised algorithm with the greatest accuracy for 2019 will be visualized along with the actual crime occurrences from 2019
    - Predicted crime percentage
        - The predicted crime percentage for each neighborhood for the

year 2019
- The predicted crime percentage generated by each algorithm used will be compared to the actual 2019 crime results to test for accuracy

Procedures

These procedures are for one data cycle. Each data cycle will use a different prediction algorithm:

1. Import all important python libraries (sklearn, pandas, numpy, seaborn, and matplotlib)
2. Import 2009-2018 COBRA data set (csv file)
3. Preprocess the data into relevant columns, which will be parameters that will later be fed into the algorithm.
4. Further clean up the csv file by removing all nil values and replacing them with accurate representations
5. Create a simple table of the preprocessed data for documentation
6. Run PCA dimensional reduction to reduce size while retaining features
7. Run every algorithm, which will return a table
8. Use pandas to output the % difference for each algorithm for all of the neighborhoods into an Excel spreadsheet
9. Graph results
10. Use geopandas to plot the locational data of predicted crimes (longitude and latitude) into the Atlanta Regional Commission shapefile and shade in the areas of the bitmap appropriately according to crime occurrence
11. Use geopandas to plot the locational data of actual crimes into the shapefile and shade in the areas of the bitmap appropriately according to crime occurrence
12. Compare the visualizations and clean them if needed

Risk and Safety

This project involves no physical risks because it is a computer science project with no hazardous materials.

Sample Data Table

Type of Algorithm Used vs Accuracy

	% Difference (dimensionless)	Accuracy (%)
Neighborhood 1		
Neighborhood 2		
Neighborhood 3		
Neighborhood 4		
…last Neighborhood (243)		

Atlanta has 243 neighborhoods, so the table will be very long. A tentative solution for now is to save the tables for all of the algorithms in an Excel spreadsheet, and when the time comes to import the tables, I will include only ten of the neighborhoods in the final table but in equal subintervals (I will of course attach the file as well, or it will be available on the GitHub page).

Addendums since Proof of Concept to be included in Beta II (04/27/22) (from logbook):

Type of Algorithm Used vs Accuracy

	Root Mean Squared Error Value (dimensionless)	Accuracy (%)
Neighborhood 1		
Neighborhood 2		
Neighborhood 3		
Neighborhood 4		
…last Neighborhood (243)		

I decided to use RMSE instead of % difference, as it is better for data scientists in general in the context of machine learning. I got approval to do this from you in early January, as we technically have not learned this statistical analysis method in Biology. Same number of rows—one for each neighborhood.

Graph Description

I will make two graphs. Both of them will be bar graphs with two bar graphs per category—one bar with dimensionality reduction and one bar without dimensionality reduction. There is no need for error bars because matplotlib automatically adjusts the data once the regressors are applied; in fact, the data that I will be graphing itself shows the margin of error for all of the data calculated and processed previously, and the data that was calculated and processed previously will be plotted on the visualization graphs that involve the Atlanta shapefile with all of the neighborhoods on it and colored accordingly. For my first bar graph, the x-axis will be labeled "Classification Method," and since the x-axis is categorical, it does not have units. The y-axis will be labeled "Category Precision," and will be a percentage. The first bar graph essentially tells us how accurate the data was for each classification method with and without dimensionality reduction. The second bar graph will be based on the RMSE calculations from earlier on in the experiment. The x-axis will be the type of regression used, while the y-axis will be the RMSE calculated value, and both values are dimensionless. For each algorithm used in the second bar graph, there will be two bars just like the first bar graph—one bar with RMSE alone and one bar with RMSE and dimensionality reduction.

Addendums since Proof of Concept to be included in Beta II (04/27/22) (from logbook): Employing PCA dimensionality reduction had no statistically significant impact on the data. As a result, I have decided to switch to using K-means as a classifier method for adding features to the data. This would require me to utilize the Elbow Method to choose an optimal K-value and normalize the data per the Euclidean distance between the centroids. As a result, I will be using K-means for dimensionality reduction instead of as an algorithm to predict crime metrics, and since PCA dimensionality reduction was of no use, there is no need to incorporate it in the final charts (I got an RMSE value of zero for all of the values in the PCA table). Therefore, I will be using decision trees, random forests, and linear regression as my

supervised algorithms and Naïve Bayes as an unsupervised algorithm. It may seem strange that I am using only one unsupervised algorithm compared to three supervised algorithms in this project; however, when you are forced to use one unsupervised algorithm for dimensionality reduction because the other one does nothing, you have to compromise, especially with large data sets such as these (300,000+ rows).

Analysis of Results

After the COBRA data set has been preprocessed, fed into the various machine learning algorithms individually, and outputted its results into 6 different tables (one for each algorithm), I will use the RMSE equation to determine the overall accuracy of my program. The RMSE will be calculated for each neighborhood for each algorithm and compared to see which algorithm is the most efficient. My final data tables, all of which will include the results from the RMSE tests, will also include a final "average" row, which will take the average of all of the previous neighborhood RMSE results as an aggregated RMSE value for the algorithm being tested. Keep in mind that the RMSE result in the final data table is NOT the same as the RMSE value from the data collection table—you need the RMSE value to conduct an RMSE test, which will give you the RMSE result. To make the process of applying the RMSE tests efficient for each neighborhood for each algorithm, I will use Excel instead of Python for assisting me through this part of the project (Python lambdas are very CPU heavy even for relatively small tables like these). I might even use Excel for creating charts based off of the applied data values if matplotlib or seaborn cannot do it for me.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Where y_i is the predicted value of an i'th term,
$\hat{y}_i$ is the action value of an i'th term, and
n is the number of data points

Bibliography

Elite Data Science. (2017, May 16). *Modern Machine Learning Algorithms: Strengths and Weaknesses*. Retrieved December 3, 2021, from https://elitedatascience.com/machine-learning-algorithms

Kaloyanova, E., Ganchev, M., & Guide, S. (2020, March 10). *How to Combine PCA and K-means Clustering in Python?* 365 Data Science. Retrieved December 13, 2021, from https://365datascience.com/tutorials/python-tutorials/pca-k-means/

Kumar, V. (2021, July 2). *Naïve Bayes Algorithm overview explained.* TowardsMachineLearning. Retrieved December 14, 2021, from https://towardsmachinelearning.org/naive-bayes-algorithm/

Kumar, V. (2021, July 9). *Decision Tree Algorithm*. TowardsMachineLearning. Retrieved December 11, 2021, from https://towardsmachinelearning.org/decision-tree-algorithm/

Kumar, V. (2021, July 16). *Random Forest*. TowardsMachineLearning. Retrieved December 14, 2021, from https://towardsmachinelearning.org/random-forest/

Kumar, V. (2021, July 23). *K-Means*. TowardsMachineLearning. Retrieved December 11, 2021, from https://towardsmachinelearning.org/k-means/

Li, H. (2020). *Which machine learning algorithm should I use?* Retrieved December 3, 2021, from https://blogs.sas.com/content/subconsciousmusings/2020/12/09/machine-learning-algorithm-use/

Machine Learning Algorithms. (2021, August 13). Microsoft Azure. Retrieved December 14, 2021, from https://azure.microsoft.com/en-us/overview/machine-learning-algorithms/

Saini, A. (2021, September 16). *Naive Bayes Algorithm: A Complete guide for Data Science Enthusiasts*. Analytics Vidhya. Retrieved December 14, 2021, from https://www.analyticsvidhya.com/blog/2021/09/naive-bayes-algorithm-a-complete-guide-for-data-science-enthusiasts/

Saxena, S. (2019, October 15). *Mathematics Behind Machine Learning | Data Science*. Analytics Vidhya. Retrieved December 14, 2021, from https://www.analyticsvidhya.com/blog/2019/10/mathematics-behind-machine-learning/

Sayad, S. (2012, November 8). *Decision Tree - Regression*. SaedSayad. Retrieved December 3, 2021, from http://www.saedsayad.com/decision_tree_reg.htm

Stojiljković, M. (2019, April 15). *Linear Regression in Python – Real Python*. Real Python. Retrieved December 14, 2021, from https://realpython.com/linear-regression-in-python/

Towards Data Science. (2019, March 11). *Which machine learning model to use?* Towards Data Science. Retrieved December 4, 2021, from https://towardsdatascience.com/which-machine-learning-model-to-use-db5fdf37f3dd

Atlanta crime rates and statistics—NeighborhoodScout. (2019, September 30). Retrieved March 10, 2022, from https://www.neighborhoodscout.com/ga/atlanta/crime

BETA 1*

Independent Variable

Remained the same

Dependent Variable**

- My dependent variable will be a "predicted crime score" instead of a "predicted crime percentage" (or crime occurrence)
  - It is clear that I need to be measuring the total magnitude of all of the crimes that occur in a particular neighborhood (rather than assuming the crimes are equal or trying to predict the occurrence of a specific crime) because different crimes have different severities
  - Overall, measuring the accuracy of the predicted crime score in comparison to the actual crime score is much easier than working with percentages. It also expresses the crime level for a particular neighborhood in a single number, no matter what the time period or range may be
  - For information on how the predicted crime score is measured and what units

it will be in, see the 2nd bullet in “Materials List”

(Accuracy still takes precedence over predicted crime score)

Controlled Variables/Constants**

- Crime weighting
  - Depending on the crime category, the weight of each crime category when calculating crime scores remains constant.
  - Category 1 crimes are weighted by 1000x; category 2 by 100x; category 3 by 10x; and category 4 (larceny) by 1x

Materials List**

Addendums since Proof of Concept to be included in Beta I (03/29/22):

- The shapefile being used for the visualizations of this project will no longer be from the Atlanta Regional Commission (ARC). Instead, they will be .kml files sourced directly from the Atlanta PD website. The beat/patrol zone map can be found on the website by searching up “Zone (num) Beats”
  - The reason for this change is because I had to convert the ARC shapefile (originally in XML vector format) to a bitmap layout, which took lots of computing power to render a single image. Kml files are much easier to work with when using the Seaborn library
- Aggregated “crime score” (To be added to “outputs”)
  - A sum of the crime category counts for a particular neighborhood for a particular day
  - Category 1 crimes are weighted by 1000x because they are the most severe; category 2 by 100x; category 3 by 10x; and category 4 (larceny) by 1x
  - The predicted crime score is a dimensionless scalar value, so it has no units

Procedures**

- Instead of using the Atlanta Regional Commission shapefile, I will use the beat region maps that are directly available on the Atlanta PD website.
- Instead of using crime occurrence as a metric for predicting future crimes, I will be using the predicted crime score.

* These will be developed throughout the project.

* Everything left unmentioned remained the same from the last prototype/iteration

BETA 2*

Independent Variable**

As stated in an addendum in my proof of concept, I removed PCA dimensionality reduction

for Beta II because it did nothing. Thus, I have to use K-means for dimensionality reduction instead of as a classifier, which brings the number of algorithms I am testing down to four from six. However, I will have a new independent variable, which is whether or not the data has clustered features added to it or not (basically whether K-means was applied to it or not). This independent variable will be shown in my final two graphs as second bar graph of a different color under the same algorithm category.

Dependent Variable

Remains the same

Controlled Variables/Constants

Remains the same

Materials List**

- Instead of calculating the percent difference of the expected crime score and the actual crime score, I will be using the RMSE (root mean square error) equation to determine how accurate the program was in predicting the crime results
    1. RMSE is better to use than percent difference in this scenario, as I can actually compare the predicted values to the actual results effectively
    2. It also helps me determine whether my predicted values are statistically significant in comparison to the actual values
    3. Look at analysis of results for more insight about RMSE and how I will analyze my data

Procedures

- Instead of doing percent difference for my statistical test, I will be using the RMSE equation
- There will be no principal component analysis dimensionality reduction
- Completed procedure:
    1. Import all important python libraries (sklearn, pandas, numpy, seaborn, and matplotlib)
    2. Import 2009-2018 COBRA data set (csv file)
    3. Preprocess the data into relevant columns, which will be parameters that will later be fed into the algorithm.
    4. Further clean up the csv file by removing all nil values and replacing them with accurate representations
    5. Create a simple table of the preprocessed data for documentation
    6. Run K-means clustering to reduce size while retaining features
    7. Run every algorithm, which will return a table for each algorithm, and

output them into an Excel spreadsheet using Pandas
8. Use Excel to output the RMSE values for each algorithm for all of the neighborhoods into a separate spreadsheet
9. Graph results as a bar graph
10. Use geopandas to plot the locational data of predicted crimes (longitude and latitude) into the Atlanta Regional Commission shapefile and shade in the areas of the bitmap appropriately according to crime occurrence
11. Use geopandas to plot the locational data of actual crimes into the shapefile and shade in the areas of the bitmap appropriately according to crime occurrence
12. Compare the visualizations and clean them if needed
13. Repeat steps 7-9 but without post-processing (K-means)

Abbreviated title | Polynomial Interpolation and K-Mean Mining to Predict Crime Rates

Group members | Anish Goyal

Science Fair: Data Analysis and CERA

I. Original Hypotheses/Goals

A. Null Hypothesis

Random forests do not provide the most accuracy when compared to other supervised and unsupervised machine learning algorithms and are difficult to process and implement.

B. Alternative Hypothesis

Random forests provide the most accuracy when compared to other supervised and unsupervised machine learning algorithms and are easy to process and implement.

II. Data Table(s)

Decision Trees

	Predicted Crime Score	Actual Crime Score
Downtown	1951.105	2423
Greenbriar	2015.333	2214
Wildwood (NPU-C)	2362.165	2519
…Georgia Tech (243)	1999.276	2109

Random forests

	Predicted Crime Score	Actual Crime Score
Downtown	2091.182	2423
Greenbriar	2509.444	2214
Wildwood (NPU-C)	2392.918	2519
…Georgia Tech (243)	1993.057	2109

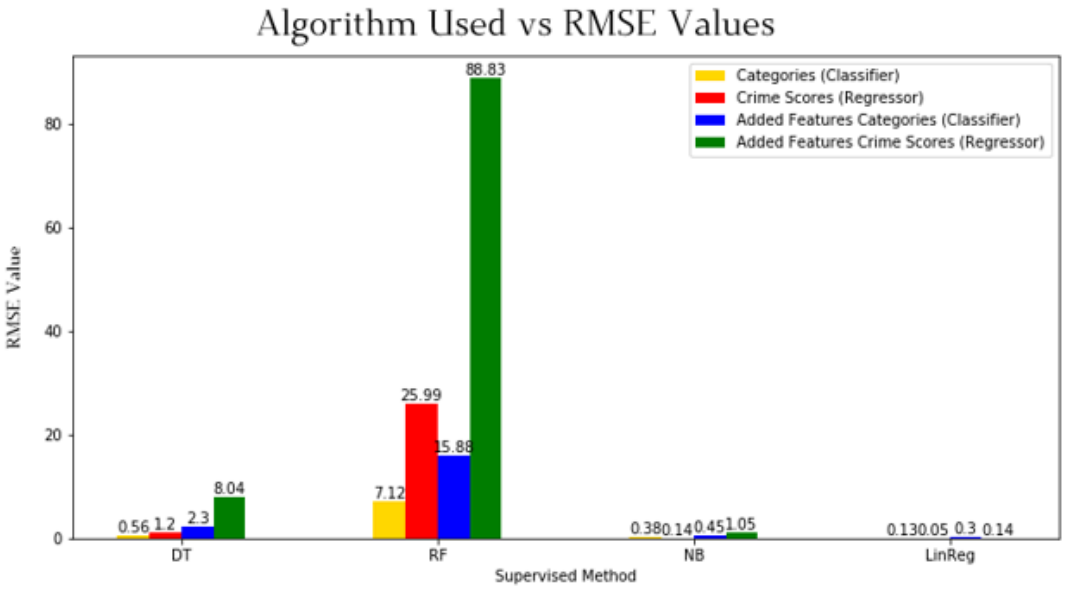
Naive-Bayes

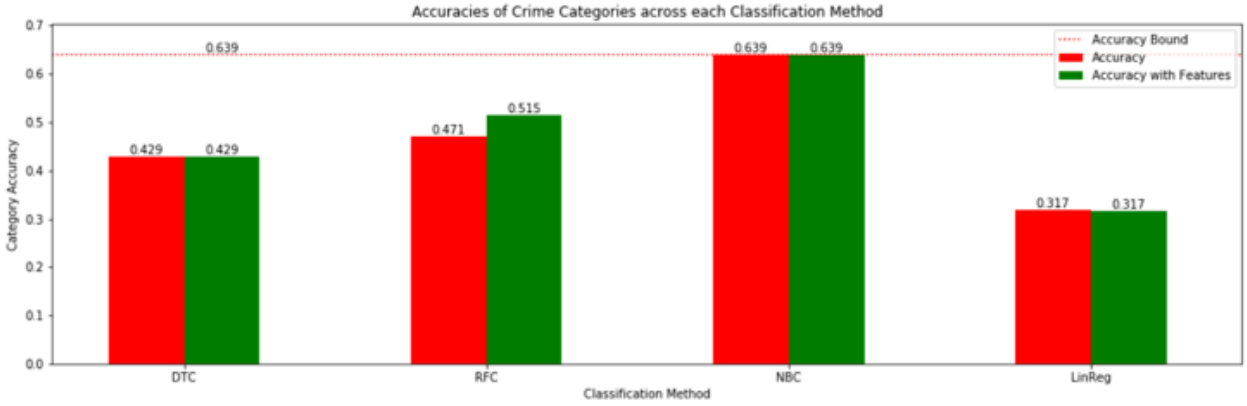
	Predicted Crime Score	Actual Crime Score
Downtown	2355.491	2423
Greenbriar	2149.219	2214
Wildwood (NPU-C)	2391.773	2519
…Georgia Tech (243)	2003.838	2109

Linear Regression

	Predicted Crime Score	Actual Crime Score
Downtown	2338.182	2423
Greenbriar	2002.818	2214
Wildwood (NPU-C)	2144.558	2519
…Georgia Tech (243)	1989.190	2109

III. Analysis





A. Graphical analysis

There is no way for sci-kit learn to put 2x SEx error bars for data that has already been preprocessed. It is not needed anyway because matplotlib automatically adjusts the data once the regressors are applied. "Added features" represents the data after dimensionality reduction was applied.

B. Statistical analysis

Decision Trees

	Root Mean Squared Error Value (dimensionless)	Accuracy (%)
Downtown	**0.56**	**41.24**
Greenbriar	**1.20**	**48.93**
Wildwood (NPU-C)	**2.30**	**54.77**
…Georgia Tech (243)	**8.04**	**65.31**
Average	**3.0123**	**42.904**

Random forests

	Root Mean Squared Error Value (dimensionless)	Accuracy (%)
Downtown	**7.12**	**46.83**
Greenbriar	**25.99**	**50.21**
Wildwood (NPU-C)	**15.88**	**56.44**
…Georgia Tech (243)	**88.83**	**49.72**
Average	**34.411**	**52.500**

Naive-Bayes

	Root Mean Squared Error Value (dimensionless)	Accuracy (%)
Downtown	**0.38**	**32.20**
Greenbriar	**0.14**	**81.17**
Wildwood (NPU-C)	**0.45**	**50.34**
…Georgia Tech (243)	**1.05**	**70.69**
Average	**0.402**	**63.903**

Linear Regression

	Root Mean Squared Error Value (dimensionless)	Accuracy (%)
Downtown	**0.13**	**34.24**
Greenbriar	**0.05**	**39.17**
Wildwood (NPU-C)	**0.3**	**28.28**
…Georgia Tech (243)	**0.14**	**26.88**
Average	**0.158**	**31.701**

Reject or fail to reject H_o?

I am failing to reject the H_o because, although decision trees were fairly accurate in its crime score predictions based on regressors such as date and location, they were beaten in accuracy by random forests and Naive-Bayesian classifiers, both which had an

average accuracy of 63% but was over 70% accurate in neighborhoods such as Greenbriar and Georgia Tech.

IV. Conclusion- CERA (Claim, Evidence, Reasoning, Application)

A. Claim

My hypothesis that decision trees would have the best overall accuracy among all of the sci-kit learn algorithms tested has been rejected because Naive-Bayes performed better than decision trees both before and after PCA dimensionality reduction was applied.

B. Evidence

An overall trend shown in the data is that the application of dimensionality reduction (or adding features) showed an increase in the accuracy of the algorithms once applied to the 2009-2018 data set. For example, adding features to the regressors and classifiers of the random forest algorithm showed a 7.22% increase from 47% accuracy to 52% accuracy as well as tripled the RMSE value of the regressors, showing an increase in accuracy. Random forests had the largest RMSE value peaking at 88.83 after applying PCA with regressors, while linear regression had the smallest RMSE value with classifiers only at 0.13.

C. Reasoning

The evidence supports my claim because decision trees had a medium RMSE value in comparison to the other sci-kit learn algorithms, which means that the distribution of the predicted values varied to a medium extent. On the other hand, Naive-Bayes had one of the lowest RMSE values peaking at 1.05 showing little variance in its predictions yet boasted the highest overall accuracy out of all of the algorithms. This shows that Naive-Bayesian classifiers predicted accurately and consistently while decision trees predicted semi-accurately and inconsistently. Linear regression had the lowest RMSE value because its predicted data was essentially fully linear with little to no jumps, which shows that linear regression had the most consistency but was imprecise. In this experiment, I did not account for overfitting and underfitting, so it is possible that, even though Naive-Bayes had the highest accuracy in terms of comparing its predicted crime scores with the 2019 crime scores, it might have fallen drastically in accuracy compared to up-to-date crime scores, while the other algorithms have not. If I had the opportunity to do this project again, I would do a fitting test to see whether my R^2 values do not overfit or underfit the data, or at least try to keep the

R^2 value between the algorithms consistent enough so that there is no bias whatsoever, and the pure accuracy of the algorithms are being tested.

D. Application

Because of this research, it can be seen that Naive-Bayesian classifiers held the greatest accuracy and consistency (low RMSE value) when compared against the other algorithms. If the model was updated continuously, it would be optimal to use Naive-Bayes to predict future crime-heavy areas and assign police patrol routes or build infrastructure such as CCTV cameras there. However, Naive-Bayes also had the greatest runtime (approximately 83 seconds) on a desktop computer with one of the best graphics cards currently available to the public, so it would actually be optimal for police servers to run random forests instead, which had a runtime of only 21 seconds and was still >50% accurate overall. Future experimentation should be conducted with linear regression so that the regression model is no longer linear—increasing the degree of the prediction regression function may yield higher accuracy results, and since regression models have the shortest runtimes out of all of the algorithms, it would be beneficial to optimize them as much as possible. Another thing I would consider for future work is implementing similar prediction models using neural networks. The accessibility of neural networks has increased significantly over time (along with their complexity), and there are many networks that could be beneficial for crime prediction analysis in the city of Atlanta or any other city on sci-kit learn and numpy. Finally, my methodology for this project could have improved without question. Given more time, I planned to merge my predictive models with other datasets regarding the socioeconomic status of each of Atlanta's neighborhoods. This would have allowed the average wage, city expenditure, and other regressors to be factored into the predictive model, which potentially would have been the most accurate regression model produced so far.

References

Elite Data Science. (2017, May 16). *Modern Machine Learning Algorithms: Strengths and Weaknesses*. Retrieved December 3, 2021, from https://elitedatascience.com/machine-learning-algorithms

Kaloyanova, E., Ganchev, M., & Guide, S. (2020, March 10). *How to Combine PCA and K-means Clustering in Python?* 365 Data Science. Retrieved December 13, 2021, from https://365datascience.com/tutorials/python-tutorials/pca-k-means/

Kumar, V. (2021, July 2). *Naïve Bayes Algorithm overview explained*. TowardsMachineLearning. Retrieved December 14, 2021, from https://towardsmachinelearning.org/naive-bayes-algorithm/

Kumar, V. (2021, July 9). *Decision Tree Algorithm*. TowardsMachineLearning. Retrieved December 11, 2021, from https://towardsmachinelearning.org/decision-tree-algorithm/

Kumar, V. (2021, July 16). *Random Forest*. TowardsMachineLearning. Retrieved December 14, 2021, from https://towardsmachinelearning.org/random-forest/

Kumar, V. (2021, July 23). *K-Means*. TowardsMachineLearning. Retrieved December 11, 2021, from https://towardsmachinelearning.org/k-means/

Li, H. (2020). *Which machine learning algorithm should I use?* Retrieved December 3, 2021, from https://blogs.sas.com/content/subconsciousmusings/2020/12/09/machine-learning-algorithm-use/

Machine Learning Algorithms. (2021, August 13). Microsoft Azure. Retrieved December 14, 2021, from https://azure.microsoft.com/en-us/overview/machine-learning-algorithms/

Saini, A. (2021, September 16). *Naive Bayes Algorithm: A Complete guide for Data Science Enthusiasts*. Analytics Vidhya. Retrieved December 14, 2021, from https://www.analyticsvidhya.com/blog/2021/09/naive-bayes-algorithm-a-complete-guide-for-data-science-enthusiasts/

Saxena, S. (2019, October 15). *Mathematics Behind Machine Learning | Data Science*. Analytics Vidhya. Retrieved December 14, 2021, from https://www.analyticsvidhya.com/blog/2019/10/mathematics-behind-machine-learning/

Sayad, S. (2012, November 8). *Decision Tree - Regression*. SaedSayad. Retrieved December 3, 2021, from http://www.saedsayad.com/decision_tree_reg.htm

Stojiljković, M. (2019, April 15). *Linear Regression in Python – Real Python*. Real Python. Retrieved December 14, 2021, from https://realpython.com/linear-regression-in-python/

Towards Data Science. (2019, March 11). *Which machine learning model to use?* Towards Data Science. Retrieved December 4, 2021, from https://towardsdatascience.com/which-machine-learning-model-to-use-db5fdf37f3dd