Crime Detection Using K-Nearest Neighbours Algorithm

Padmini Chandrashekar[1], Vishnu kk[2], Sneha Singh[3], Nazir Basha[4]
[1]Assitant Professor, Dept of of Computer Science and Engineering
[2, 3, 4] Dept of of Computer Science and Engineering
[1, 2, 3, 4] Bangalore, Karnataka, India

Abstract- *Crime is one the of the most alarming events that takes place in our daily life, which can be traumatizing. Every day, crimes occur in masses, and different levels of magnitude. The decision taken for this crime can vary on various factors like the type of crime that was committed. Petty thefts are usually pardoned, with either community service, a penalty, or jail time ranging from few days to a week, whereas crimes involving homicide and sexual assault usually taken to court, and the perpetrator is given a life sentence. We tried using machine learning algorithms on the crime dataset. In our project, we have used San Francisco Crime Dataset, which holds all the crime incidents that took place over a period of 3 years. We will be using K-Nearest Neighbours to determine the decision that was taken based on the details of the crime, like co-ordinates, timestamp, and category of crime. We have achieved a detection model with around 83% of accuracy.*

Keywords- Crime Detection, Machine Learning, Classification, k-Nearest Neighbours.

I. INTRODUCTION

Crimes have always been a major problem ever since the existence of mankind. To resolve this problem, a judiciary system was practiced, so that the higher authorities, usually governing bodies can make valid decisions to give justice to the victim, and the perpetrator is reprimanded and rehabilitated. We have made use of the crime records that was documented by the Police Department of San Francisco, over a period of 3 to 5 years. San Francisco, being the home of Hollywood, surrounded by rich aristocrats, celebrities and tech giants, burglary is the most occurred crime, in the city. Studies also shows that crimes tend to occur the most in broad daylight, and the Southern regions of San Francisco are more prone to occurrence of crime events.

The ever-increasing computing power has facilitated determining answers for those problems, that were once considered unanswerable. It has also made way to artificial intelligence, deep learning, and machine learning, which involved complex mathematical computations. Geoffrey Hinton, regarded as the godfather of deep learning, proposed several algorithms, which were not implemented until the dawn of the 21st century. So, in the early 2010s,machine learning rose to prominence and ushered new horizons for analysis, prediction in various domains.

Machine Learning is the process of training a machine, based on the occurrences that happened in the past. The machine learns like how a human learns how to distinguish between a dog and a cat. There are two categories - unsupervised and supervised machine learning algorithms. Supervised learning has data to learn from, which involves regression and classification methods. Unsupervised learning involves learning from the pre-existing features.

1.1 GOALS &OBJECTIVE

Our main objective is to build a classifier model, an algorithm usually used to determine categorical data. We have used k-Nearest Neighbours algorithm, where we provide details regarding the incidents of the crime. Our next objective would be to implement the classifier model into a web application, so that the users can input data, and obtain results.

1.2 SCOPE

Our classifier model is built on the San Francisco Crime Records, so out model will not be able to detect resolutions taken for crime activities that have taken place in other geographical regions. Our web application is built using the Python library, Flask, so the web application can be accessed only through web browsers.

1.3 METHODOLOGY

We have implemented the k-Nearest Neighbours algorithm, supervised machine learning classifier algorithm, commonly used to determine categorical variables.

1.3.1.K-NEAREST NEIGHBOURS

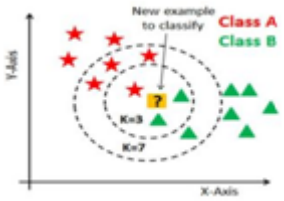


Figure. k-Nearest Neighbours, source

k-Nearest Neighbours algorithm is used to classify data based on categorical variables. We first plot the points on the cartesian plane, where each axis represents the independent variable. We then group the points on the plane using similarity measures. We usually use the shortest distance as our similarity metric.

k-Nearest Neighbours involves the following steps –

1. First, we initialise k number of points randomly on the plane We call them as centroids
2. We take the distance measure, usually the Minkowski distance measure, for all the points on the plane, from these randomly generated centroids.
3. Depending on the distance values, we assign the points to the centroid to which it has the shortest distance.
4. We obtain new clusters, but since we randomly chose the centroids, we cannot assume these centroids as the accurate values. We need to determine the actual centroids. So, we take distance measure for every other point, and generate a distance matrix. Based on the values of distances, we take the arithmetic mean of the coordinates to generate a new centroid.
5. With the new centroids, we calculate the distances from these new points from the rest of the points, and new clusters are assigned. As we can see, it is an iterative procedure.
6. We repeat step 4 until our new centroid will be the exact same value as that of the previous iteration. This means that the centroid is fixed, and the distance measures used to classify data will be more accurate.
7. Initially, we chose the value of k randomly, but this might not give accurate results. To obtain accurate results, we apply the algorithm for various values. The k value to which the accuracy level is highest is the value we choose.

II. LITERATURE SURVEY

Kim, et. al [2] presented a crime prediction model, which uses k-Nearest Neighbours and Boosted Decision Trees The model was trained on Vancouver Crime data for the past 15 years. They obtained 39% accuracy using k-Nearest Neighbours & 44% accuracy using Boosted Decision Trees.

Shermila, et. al. [3] used multilinear regression, k-nearest neighbours, and concurrent neural networks to identify the details of the perpetrator involved in the crime. The model was able to produce an accuracy of 60% to predict the perpetrator age, 96% for sex and 97% for relationship status.

Pratibha, et. al [4] used k-Nearest Neighbours, Artificial Neural Networks, Extra Trees, Support Vector Machines and Decision Trees to highlight the violent crimes that occur in various geographical locations. They did not use Support Vector Machines as it can be time consuming. They obtained 88% accuracy with Decision Trees, 87% with k-Nearest Neighbours, 88% with Extra Trees, 16% with Artificial Neural Networks and 66% with Support Vector Machines.

2.1 EXISTING &PROPSEDSYSTEM

As we can wee that the accuracy levels were not satisfactory in the papers, we tried to implement a classifier model, which determines the verdict that was taken after the incidents of the crime. The papers already provide a solution to determine the details of the perpetrator, and the occurrence of crime in a geographical location. The main purpose of our research paper is to highlight the power of machine learning algorithms in the field of judiciary system.

2.2 FEASIBILITYSTUDY

Our application predicts the verdict of a crime event before the First Information Report is filed after the occurrence of the crime, and then the verdict is given. Our classifier model can be used to identify an estimate about the decision that will be made, with the help of statistical methods. Our application requires only a browser, so it is platform independent, and is feasible across any device that supports internet. The user does not require any prior technical skills, as they just must provide inputs, and obtain the results. Our application is free to use, so it is economically feasible aswell.

2.3 HARDWARE & SOFTWARE REQUIREMENTS USED

All the machine learning models were trained on Google Colab notebook, an online code editor, exclusively for Python, which facilitates quick prototyping. Google Colab uses Dual Core Intel Xeon processors, of clock speed 2.3GHz, with 12GB of RAM and 25 GB of internal storage, on processor mode. Since machine learning algorithms involve complex matrix multiplications, it is advised to use Graphical Processing Unit (GPU) or Tensor Processing Unit (TPU) a

special chipset developed by Google for data science and analytics, as GPUs are built to render high resolution images, which are technically multi-dimensional arrays of pixels, and TPUs are specifically created for complex matrix operations. As this is a cloud environment, the developer would not require anything more than what a basic browser would demand, which is usually not more than 4GB of RAM.

The web application was built using Visual Studio Code, a well optimised text editor for web and app development. VS Code requires about 4-8 GB or RAM, 4 cores of central processing units, and little to no computing power. The web application was built using Flask, a lightweight Python library used for static webpages, and backend REST API servers. The web application is deployed on Heroku Cloud Service, where repositories can be deployed without any hassle.

III. SOFTWARE REQUIREMENTSSPECIFICATION

3.1 FUNCTIONALREQUIREMENTS

Our application takes the inputs from the users, which involves data regarding the events of the crime. Our classifier model is trained to detect the verdict taken for the crime events, depending on the inputs. The inputs are coordinates of the crime incident, the timestamp of the crime incident, and the type of the crime that took place. We present a classifier model that detects the values with 83%accuracy.

We also present a web application, through which the inputs can be entered, and the categorial value can be determined. This web application works on any given browser We have deployed the web application on a Heroku cloud server, so the latency of the web application might be higher than usual.

3.2 NON FUNCTIONALREQUIREMENTS

Our application can be accessed through any device that supports web browsers. Our application is 83% reliable, but as the classifier model was built based on the values of San Francisco crime incidents, we can use this only on the crime events that occur in San Francisco alone. Our web application is secure to use, as our application does not involve any client-side JavaScript, so using our application with JavaScript disabled on the browser would still work perfectly. Our application might not be secure to use in certain instances, as we have not implemented the HTTPSprotocol.

IV. DETAILEDDESIGN

4.1 USE CASEDIAGRAM

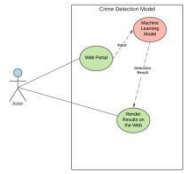


Figure. Use Case Diagram

Use case diagrams are the graphical representations of how different components of the application are organised, and how they communicate. As observed in fig, above, in our web application, the REST server and the classifier model runs on the same backend. The user data is collected through a form. Depending on these inputs, the classifier model outputs a result.

4.2 SEQUENCEDIAGRAM

Sequence diagrams show us how various components of the application interact over a period. The green vertical bars signify the duration over which the component remains active. As you can see in fig below, all the components are very much active during the whole procedure, as all the components of the application reside on a single server. The user needs to enter the crime details. This data is later sent to classifier model, through the backend. The classifier model then outputs an integer value. The backend server then decodes the categorical variable, which is the detected result.

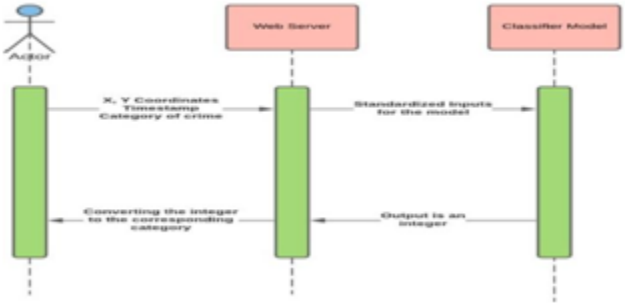


Figure. Sequence Diagram

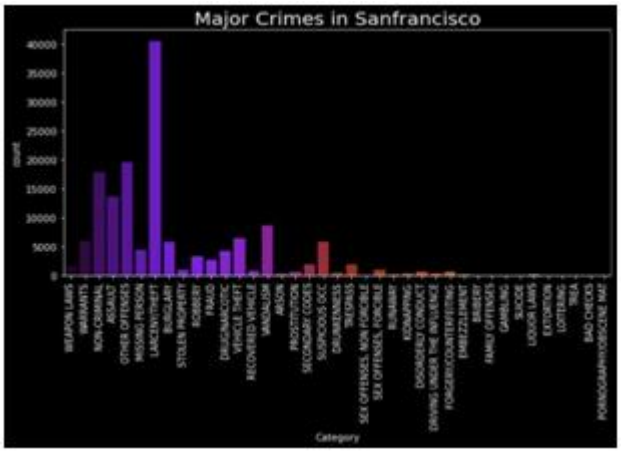
4.2 ACTIVTY DIAGRAM

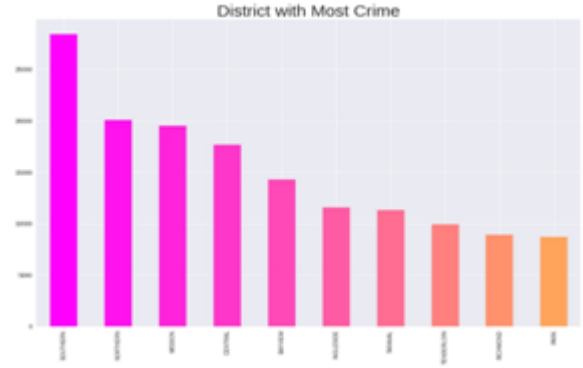
Figure. Activity Diagram

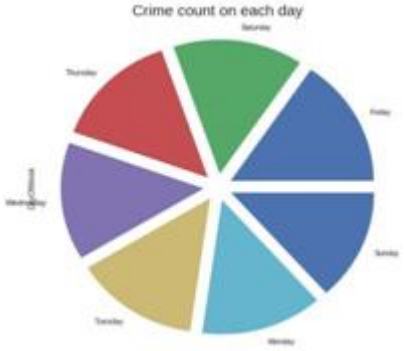
Activity diagrams visualise the roadmap that the user must follow, to execute the application and obtain the output. The black circle at the far left indicates the start of the application, whereas the circle with black outline in the far

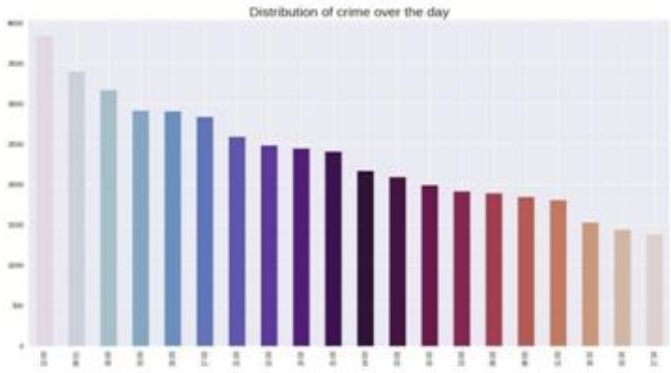
right determines the endpoint of the application. The green blocks are the steps that are executed in a sequential order.

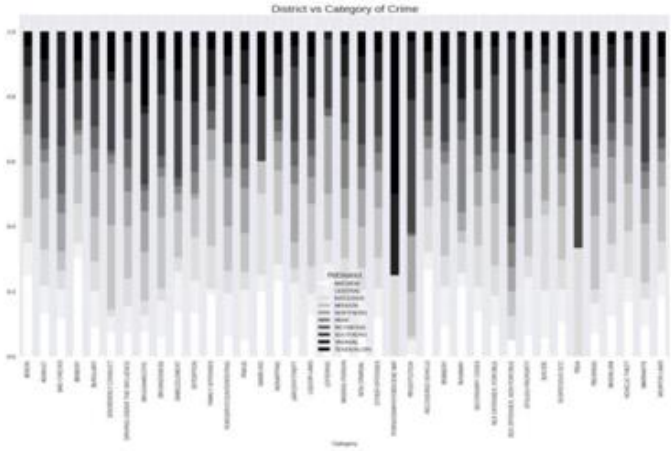
V. VISUALIZATION











VI. IMPLEMENTATION

We used the Python library Scikit-learn, which is commonly used for machine learning algorithms. Scikit-learn facilitates splitting data into two parts - training, and testing data. In our project, we have split out dataset in 8:2 fractions, where 80% of the dataset is used as training data, or the data from which our classifier model is built, and the remaining 20% of our data is the testing data, which we pass in, as inputs to determine the performance metrics & accuracy of our classifier model.

With the help of scikit-learn, we can implement k-Nearest Neighbours algorithm. We first initialise a Classifier object with parameter k, which determines the number of neighbours that the model needs to take into consideration to determine the unique class an entity would belong to. In our scenario, we have chosen k=4 randomly.

To determine the accuracy of our model, we pass in the testing dataset, and compare the detected values with actual values.

As we can see, k=4 gave us an accuracy of 79%, which is sufficient, but can be improved. We use an iterative approach to determine the best k value.

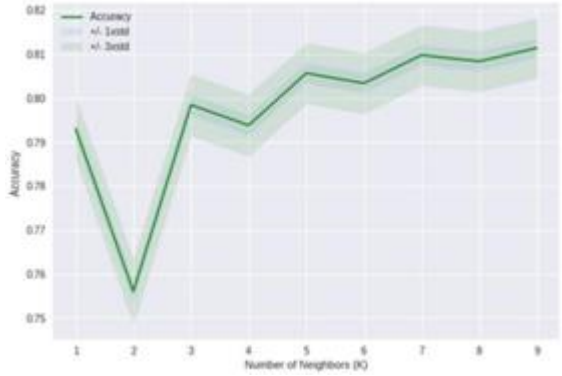


Figure. k-Nearest Neighbours Accuracy Graph

We set our iterations to 10, and train our model 10 times, to determine the optimum k value. As we can observe in the figure above, the highest accuracy is when k=9. So, using k=9, we obtained an accuracy of 83%. Now implementing the web application using REST APIs with the help of Flask, a lightweight Python library used to make static webpages. We define the RESTful routes. As we can see, the server sends a POST request, which contains all the input values, and then the server converts the input into the format which our classifier model can understand. Depending on the output, we will render the result value in the next page.

6.2. SCREENSHOTS

Figure. Home Page, where crime details are entered

Depending upon the inputs, the classifier model outputs a result.

Figure. Result

VII. CONCLUSION & FUTUREENHANCEMENTS

The web application is built using Flask, which is not reliable on the long run. We are also running the REST API, as well as our trained model on the same server, which is usually not a good option as it is regarded as bad system design and may cause latency. Due to the rise in the number of smartphone users, it is important to make our application native. Also, to improvescaling, using modern JavaScript libraries like React and Angular is the better choice. Network stability might not be constant in all geographical locations, so it is essential to convert our apps into Progressive Web Applications (PWA), which, offloads the web application, and can be accessed when the network is down. Using a Heroku cloud service over advanced web services like Amazon Web Services (AWS) would be absurd. The optimal solution would be to dockerise the application, publish it through a pipeline to an AWS instance, which ensures that the web application is scalable. A minor change that every user would desire in our web application would be an option to point the co-ordinates on the map, which we are working on right now, using the Google Maps API.

REFERENCES

[1] S. A. Gokte, “Most Popular Distance Metrics Used inKNN and When to Use Them,”2020.

[2] S. a. J. P. a. K. P. S. a. T. P. Kim, “Crime AnalysisThrough Machine Learning,” 2018 IEEE 9th

Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), pp. 415-420,2018.

[3] B. B. a. N. S. A. Mary Shermila, "Crime Data Analysis and Prediction of Perpetrator Identity Using MachineLearning Approach," 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), pp. 107-114,2018.

[4] G. U. S. D. a. L. C. Pratibha, "Crime Prediction and Analysis," 2nd International Conference on Data,Engineering and Applications (IDEA), pp. 1-6, 2020.