# PROFESSIONAL TRAINING REPORT

**at**

**Sathyabama Institute of Science and Technology**

**(Deemed to be University)**

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering

By

## SUPRIYA REPALLE

**REG.NO. 39110846**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SCHOOL OF COMPUTING**

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,**

**CHENNAI – 600119, TAMILNADU**

**APRIL 2022**

|  |  |  |
| --- | --- | --- |
|  | **SATHYABAMA**  **INSTITUTE OF SCIENCE AND TECHNOLOGY** (DEEMED TO BE UNIVERSITY) **Accredited with Grade “A” by NAAC**  (Established under Section 3 of UGC Act, 1956)  JEPPIAAR NAGAR, RAJIV GANDHI SALAI  CHENNAI– 600119  [**www.sathyabama.ac.in**](http://www.sathyabama.ac.in) |  |

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of **R.SUPRIYA(Reg.No:39110846)** who carried out the project entitled “**ANALYSIS AND PREDICTION OF CRIME IN INDORE CITY**” under my supervision from February 2022 to April 2022.

## Internal Guide

## Dr. Prayla Shyry , M.E., Ph.D.,

**Head of the Department**

**Dr. L. Lakshmanan, M.E., Ph.D.,**



## Submitted for Viva voce Examination held on

**InternalExaminer ExternalExaminer**

**DECLARATION**

I, **SUPRIYA REPALLE,** hereby declare that the project report entitled **ANALYSIS AND PREDICTION OF CRIME IN INDORE CITY**done by me under the guidance of **Dr. Prayla Shyry** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

## DATE:10-04-2022 SUPRIYA.R

**PLACE:CHENNAI SIGNATURE OF THE CANDIDATE**

**ACKNOWLEDGEMENT**

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D**, **Dean**, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of **Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. PraylaShyry, M.E., Ph.D.,** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

**INTERNSHIP CERTICATE**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER No.** | **TITLE** | **PAGE No.** |
|  | **ABSTRACT** | **i** |
| **LIST OF FIGURES** | **ii** |
| **LIST OF ABBREVATIONS** | **iii** |
| **1** | **INTRODUCTION** | **1-3** |
|  | * 1. RATIONALE   2. METHODOLOGY   3. MACHINE LEARNING   4. UNDERSTANDING THE DATASET | 2  2  2  3 |
| **2** | **AIM AND SCOPE OF THE PRESENT INVESTIGATION** | **5** |
|  | 2.1 AIM OF THE PROJECT  2.2 SCOPE OF THE PROJECT  2.3 LITERATURE REVIEW | 5 |
| **3** | **EXPERIMENTAL METHODS, MATERIALSUSED** | **6-15** |
| * 1. IMPLEMENTATION   3.2 ALGORITHMS IMPLEMENTED  3.2.1KNN ALGORITHM  3.2.2 DECISION TREE  3.2.3 RANDOM FOREST  3.3 DATA PRE-PROCESSING  3.4 IMPORTING THE PACKAGES, LIBRARIES  3.4.1 NUMPY  3.4.2 MATPLOTLIB  3.4.3 SNS PLOT  3.4.4 SKLEARN  3.4.4.1 TEST\_TRAIN SPLIT  3.4.4.2 MEAN\_SQUARED ERROR  3.4.4.3 R2\_SCORE | 7  7  8  9  10  11  12  12  12  13  14  14  14  15 |
| **4** | **RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS** | **16-26** |
|  | 4.1 RESULTS FROM ML ALGORTIHMS  4.2 RESULTS FROM VISUALIZATION  4.3 RESLUTS FROM PREDICTION  4.4 CALCULATING ERRORS | 17  17  21  26 |
| **5** | **SUMMARY AND CONCLUSION** | **21** |
|  | 5.1 SUMMARY  5.2 FUTURE SCOPE |  |
| **6** | **REFERENCES**  **APPENDIX**   1. **SCREENSHOTS** 2. **SOURCE CODE** | **22**  **23-27** |

# ABSTRACT

To be better prepared to respond to criminal activity, It is important to

understand patterns in crime. In this project, i analyzed crime data from the city of Indore

At the outset, the task is to predict which category of crime is most likely to occur given a time and place in Indore. The use of AI and machine learning to detect crime via sound or cameras currently exists, is proven to work, and expected to continue to expand.

The use of AI/ML in predicting crimes or an individual’s likelihood for committing a crime has promise but is still more of an unknown. The biggest challenge will probably be “proving” to politicians that it works.

When a system is designed to stop something from happening, it is difficult to prove the negative. Companies that are directly involved in providing governments with AI tools to monitor areas or predict crime will likely benefit from a positive feedback loop. Improvements in crime prevention technology will likely spur increased total spending on this technology. The project also attempts to make the classification task more meaningful by merging multiple classes into larger classes. Finally, the project report and reflect on the results with different classifiers, and dwell on avenues for future work

# i

# LIST OF FIGURES

## FIG.NOFIGURENAME PAGE NO.

## 

1. 1.2.1 Machine Learning Process 3
2. 3.2.1 Principle diagram of KNN 8
3. 3.2.1.1Distance Functions 9
4. 3.2.2Decision Tree10
5. 3.2.3 Random Forest11
6. 3.3 Dataset after Preprocessing 12
7. 4.1.1Results From KNN Algorithm 16
8. 4.1.2Results from Decision Tree 16
9. 4.1.3 Results from Random Forest 17
10. 4.2.1 Heat map 17
11. 4.2.2 Act 379 vs Hour 18
12. 4.2.3Act 13 vs Hour18
13. 4.2.4. Act 323 vs Hour19
14. 4.2.5 Act 363 vs Hour 19
15. 4.2.6 Comparision between columns 20
16. 4.3 Results from Prediction 21
17. 4.4MAE vs K 21

ii

# LIST OF ABBREVIATIONS

ABBREVIATION EXPANSION

KNN K Nearest Neighbours

iii

**CHAPTER 1**

**INTRODUCTION**

Criminal activity is gradually rising in India and has a significant and negative social impact. The recent spurt in the nation has put everyone wondering as to what will happen in the future. Cases of murder, abduction, rape, and fatal accidents have skyrocketed. The need of the hour is to make people of the nation realize the issue. Machine learning advancements and deep learning algorithms can discover hidden patterns in unstructured data sets and reveal new information.

Crime prediction and criminal identification are the major problems to the police department because there is a tremendous amount of data related to crime that exists. There is a need for technology through which the case-solving could be faster. The idea behind this project is that crimes can be easily predicted once we are able to sort through a huge amount of data to find patterns that are useful to configuring what is required. The recent development in machine learning makes this task possible.

We will give date, time, location (longitude, latitude) as input and the output will be generated which will give us information about which crime is likely to happen in that area. It basically gives us the hotspots of crime. The data is taken considering the time and type of crime that happened in the past. KNN algorithm then uses its approach which assumes that similar things exist in close proximity and classifies new cases based on similarity measures.

Classes of Crime are:

● Act 379 – Robbery

● Act 13 - Gambling

● Act 279 - Accident

● Act 323 - Violence

● Act 302 – Murder

● Act 363 – Kidnap

This prediction, if put to good use, is of great help in investigating cases that have happened. It can be used to suppress the crimes by installing some measures if we know what type of crime is going to happen beforehand. This will indirectly help reduce the rates of crimes and can help to improve security in such required areas.

* 1. **RATIONALE:**

Madhya Pradesh's commercial capital Indore has topped the crime record in the country in 2008 followed by Bhopal and Jaipur. Crime rate of Indore was 941.4, which is the highest in the country, according to National Crime Record Bureau's (NCRB) report - "Crime in India 2008".

With the rapid urbanization and development of big cities and towns, the graph of crimes is also on the increase. This phenomenal rise in offences and crime in cities is a matter of great concern and alarm to all of us.

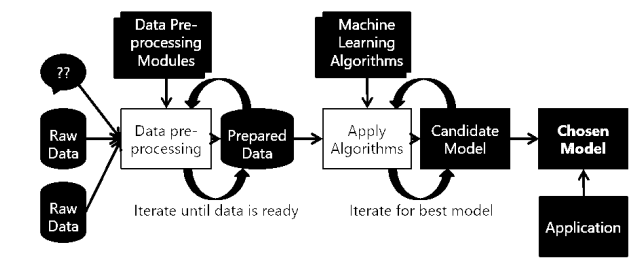
There are robberies, murders, rapes and what not. The frequent and repeated thefts, burglaries, robberies, murders, killings, rapes, shoplifting, pick pocketing, drug- abuse, illegal trafficking, smuggling, theft of vehicles etc., have made the common citizens to have sleepless nights and restless days.

They feel very insecure and vulnerable in the presence of anti-social and evil elements. The criminals have been operating in an organized way and sometimes even have nationwide and international connections and links

* 1. **METHODOLOGY:** 
     1. **MACHINE LEARNING:**

The term machine learning refers to the automated detection of meaningful patterns in data. In the past couple of decades it has become a common tool in almost any task that requires information extraction from large data sets.We are surrounded by a machine learning based technology: search engines learn how to bring us the best results placing , anti-spam software learns to filter our email messages, and credit card transactions are secured by a software that learns how to detect frauds. Digital cameras learn to detect faces and intelligent personal assistance applications on smart-phones learn to recognize voice commands. Cars are equipped with accident prevention systems that are built using machine learning algorithms.

Machine learning is also widely used in scientific applications such as bioinformatics, medicine, and astronomy. One common feature of all of these applications is that, in contrast to more traditional uses of computers, in these cases, due to the complexity of the patterns that need to be detected, a human programmer cannot provide an explicit, finedetailed specification of how such tasks should be executed. Taking example from intelligent beings, many of our skills are acquired or re\_ned through learning from our experience (rather than following explicit instructions given to us). Machine learning tools are concerned with endowing programs with the ability to learn and adapt.

****

**FIG 1.2.1 MACHINE LEARNING PROCESS**

**1.2.2 UNDERSTANDING THE DATASET:**

Beforeimplementing the machine learning algorithms on our data, we went through a series of preprocessing steps with our classification task in mind. The inputs to our algorithms are time (hour, day, month, year), place (latitude and longitude), class of crime

• Act 379-Robbery

• Act 13-Gambling

• Act 279-Accident

• Act 323-Violence

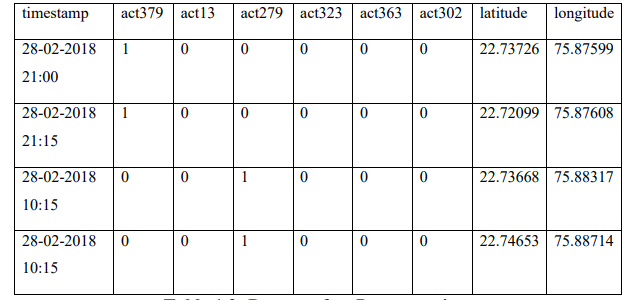
• Act 302-Murder

• Act 363-Kidnapping

The output is the class of crime that is likely to have occurred. We try out multiple classification algorithms, such as KNN (K-Nearest Neighbors), Decision Trees, and Random Forests.

The timestamp contained the year, date and time of occurrence of each crime. This was decomposed into five features: Year (2018), Month (1-12), Date (1-31), Hour (0- 23) and Minute (0-59).

Following these preprocessing steps, some out-of-the box learning algorithms as a part of our initial exploratory steps. Our new feature set consisted of 9 features, all of which were now numeric in nature.



***TABLE 1.2.2 DATASET***

**CHAPTER 2**

**AIM AND SCOPE OF THE PROJECT**

**2.1 AIM OF THE PROJECT:**

The main aim of the project is to create the best machine learning model to accurately predict the crimeand hotspots of the crime with the given time and place.The project also aims at bringing out the meaningful insights from the given dataset using visualization.Understanding patterns of criminal behavior that could help in solving criminal investigations.

**2.2 SCOPE OF THE PROJECT:**

The scope of the project is to in prredicting crime before it takes place and predicting hotspots of crime..The project also helps us to understand crime pattern and criminal behaviour according to the areas and classify crime based on location. Analysis of crime in Indore can be done.

**2.3 LITERATURE REVIEW:**

It is observed that many machine learning models are implemented on datasets of different cities having unique features so predictions are different in all cases. Classification models have been implemented on various other applications like prediction of weather, in banking, finances and also in security.

In identification of criminals by using classification techniques and crime prediction was done using data set of six cities of Tamil Nadu by using KNN classification, K-Means clustering, Agglomerative hierarchical clustering, and DBSCAN clustering algorithms. In , they used a model whose purpose was to use a database in which the data points were separated into several classes to predict the classification of a new sample point. Using features Day, Date, Year of the crime using KNN it is found to be 40% accuracy.

Their model used techniques like Logistic Regression, Decision Trees, Bayesian Methods and Support Vector Machine. Python was used to explore training data, make regression analysis and predict categories for test data, in order to get the best correlation between the features (Date, Pd-District, Address, Day of the Week, Description, Resolution, X and Y) and the target value (Category of Crime).

All nominal values were converted into binary values by converting the values of the attributes into separate new attributes and give them values of either a 0 or 1. Several trials of different Regression methods were used on the training data by splitting it into two sets; training and validation, both validation and cross-validation were conducted, the method with the least Log loss was applied to predict the results for the test data.

The idea behind this project is that crimes are relatively predictable; it just requires being able to sort through a massive volume of data to find patterns that are useful to law enforcement.This kind of data analysis was technologically impossible a few decades ago, but the hope is that recent developments in machine learning are up to the task.

**CHAPTER 3**

**ALGORITHMS,METHODS IMPLEMENTED**

**3.1IMPLEMENTATION:**

. The implementation of the project is done with the help of python language. To be particular, for the purpose of machine learning Anaconda is being used. Anaconda is one of several Python distributions. Anaconda is a new distribution of the Python. It was formerly known as Continuum Analytics. Anaconda has more than 100 new packages. Anaconda is used for scientific computing, data science, statistical analysis, and machine learning. On Python technology, we found out Anaconda to be easier. Since it helps with the following problems: Installing Python on multiple platforms.

• Separating out different environments.

• Dealing with not having correct privileges

.• Getting up and running with specific packages and libraries. This data was scraped from the publically available data from Indore police website which had been made by people in police station of different areas. Implementation of the idea started from the Indore city itself so as to limit an area for the prediction and making it less complex. The data was sorted and converted into a new format of timestamp, longitude, latitude, which was the input that machine would be taking so as to predict the crime rate in particular location or city.

**3.2 ALGORITHMS IMPLEMENTED:**

After the preprocessing described in the previous sections, we had three different classifications problems to solve, which we proceeded to attack with an assortment of classification algorithms. The following are the algorithms which we are using:

•KNN( K- Nearest neighbors)

• Decision Tree

• Random Forest

**3.2.1 KNN(K-NEAREST NEIGHBOURS) ALGORITHM:**

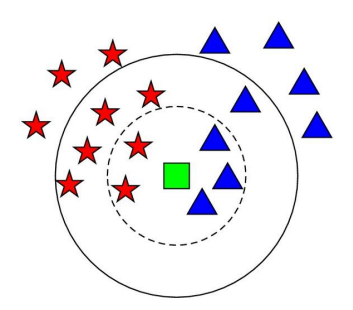
A powerful classification algorithm used in pattern recognition K nearest neighbors stores all available cases and classifies new cases based on a similarity measure (e.g. distance function).One of the top data mining algorithms used today. A non-parametric lazy learning algorithm (An Instance based Learning method).

KNN: Classification Approach

• An object (a new instance) is classified by a majority votes for its neighbor classes

.• The object is assigned to the most common class amongst its K nearest

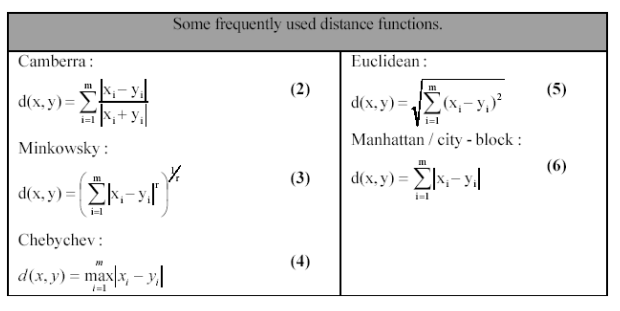
neighbors.(measured by distance function)

:

***FIG 3.2.1: PRINCIPLE DIAGRAM OF KNN***

K Nearest Neighbor Classifier is a supervised machine learning algorithm useful for classification problems. It works by finding the distances between a query and all the examples in the data, selecting the specified examples that are closest to the query, then votes for the most frequent label.

It is non-parametric which means that it does not make any assumptions on the underlying data distribution. In other words, the model structure is determined by the data. It's pretty useful because in reality, most of the data does not obey the typical theoretical assumptions made.



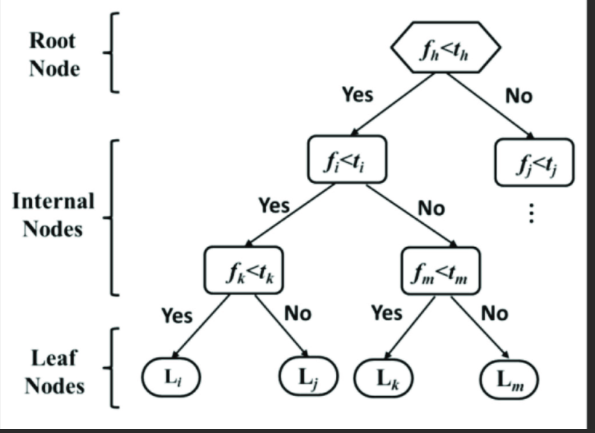
***FIG 3.2.1.1 DISTANCE FUNCTIONS***

**3.2.2 DESICION TREE:**

As the name says all about it, it is a tree which helps us by assisting us in decision-making. Used for both classification and regression, it is a very basic and important predictive learning algorithm. It is different from others because it works intuitively i.e., taking decisions one-by-one.

• Non-parametric: Fast and efficient.

• It consists of nodes which have parent-child relationships.



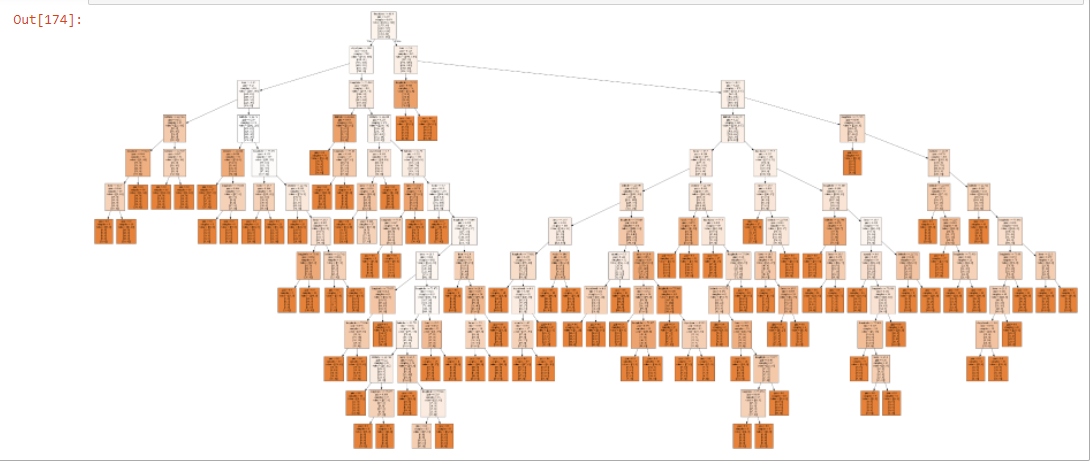
***FIG 3.2.2 DECISION TREE***

Decision tree considers the most important variable using some fancy criterion and splits dataset based on it. It is done to reach a stage where we have homogenous subsets that are giving predictions with utmost surety.

**3.2.3 RANDOM FOREST:**

Random Forests is a very popular ensemble learning method which builds a number of classifiers on the training data and combines all their outputs to make the best predictions on the test data.

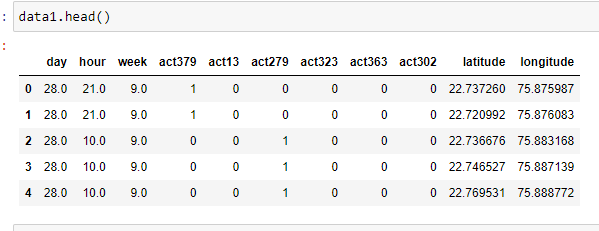
Thus, the Random Forests algorithm is a variance minimizing algorithm that uses randomness when making split decision to help avoid overfitting on the training data. A random forests classifier is an ensemble classifier, which aggregates a family of classifiers h(x|θ1),h(x|θ2),..h(x|θk). Each member of the family, h(x|θ), is a classification tree and k is the number of trees chosen from a model random vector. Also, each θk is a randomly chosen parameter vector. If D(x,y) denotes the training dataset, each classification tree in the ensemble is built using a different subset Dθk(x,y) ⊂ D(x,y) of the training dataset.



***FIG 3.2.3 RANDOM FOREST***

**3.3DATA PREPROCESSING:**

The dataset undergoes preprocessing work to see whether all the columns oare of the same data type whether any missing values present incase we have to replace them here we re-construt the dataset by splitting the timestamp column



***FIG3.3 DATASET AFTER PRE-PROCESSING***

Here we have splitted the timestamp column into different columns as day,hour ,week, to clearly analyze what is at what day and wek the crime has happened

**3.4 IMPORTING THE PACKAGES ,LIBRARIES:**

**3.4.1NUMPY:**NumPy is the fundamental package for scientific computing in python. NumPy stands for**Numerical Python**. It is open-source and we can use it freely. NumPy is a python library that is used for working with arrays.

NumPy, which stands for**Numerical Python**, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. In this project the numpy has been imported as np

Import numpy as np .

**3.4.2MATPLOTLIB:**

Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPythonotTkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers also.

Matplotlib is written in Python and makes use of NumPy, the numerical mathematics

extension of Python. Matplotlib was originally written by John D. Hunter in 2003. The

current stable version is 2.2.0 released in January 2018.Matplotlib and its dependency packages are available in the form of wheel packages on the standard Python package repositories and can be installed on Windows, Linux as well as MacOS systems using

the pip package manager.

pip3 install matplotlib

import mathplotlib.pyplot as plt.

In this project we have used various plots like bar plots ,pie charts ,scatterplots to understand and for the comparative studies

We have plotted various graphs of scatter plot model of training and testing data.

**3.4.3SNS PLOT:**

 Sea born is a library mostly used for statistical plotting in Python. It is built on top

of Matplotlib and provides beautiful default styles and color palettes to make

statistical plots more attractive. Seaborn can be installed using the pip. Type the

below command in the terminal.

pip install seaborn

After the installation is completed you will get a successfully installed message

at the end of the terminal as shown below.

**Note:**Seaborn has the following dependencies –

* Python 2.7 or 3.4+
* numpy
* scipy
* pandas
* matplotlib

We have to import seaborn as sns variable ,

import seaborn as sns.

After importing we perform various visualizations to actually understand each column in detail like to

**3.4.4SKLEARN:**

Scikit learn is the most useful and robust library in Python it is a free software machine learning library for Python programming language .

It is simple and efficient tool for predictive data analytics it is accessible to everybody end uses in various contexts this built on numpy ,sciPy and mathplot.In this project we will import linear regression ,mean squared error and r2score from this library .

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

**from**sklearn.linear\_model**import**LinearRegression

**from**sklearn.metrics**import**mean\_squared\_error,r2\_score

based on this the regression model and train \_test split is also done.

**3.4.4.1 TEST\_TRAIN\_SPLIT:**

It is used to estimate the performance of the model of machine learning algorithms when they are used to make predictions on data not used to train the model.in this project the testing data is 20% and training data is 80%.Here again the training data and testing data is further divided based on the dataset dependent and independent

**3.5.4.3 MEAN \_SQUARED ERROR:**

The Mean Squared Error (MSE) or Mean Squared Deviation (MSD) of an estimator measures the average of error squares i.e. the average squared difference between the estimated values and true value. It is a risk function, corresponding to the expected value of the squared error loss.Themean\_squared error is imported from sklearn.metrics import mean\_squared\_error,

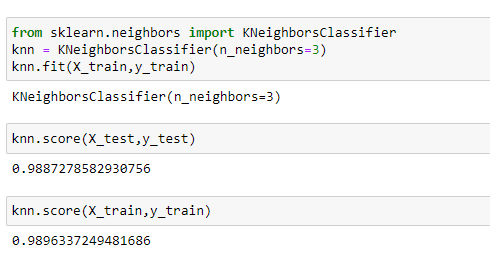
**3.5.4.4 R2 SCORE:**

R 2 (coefficient of determination) regression score function. Best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse). A constant model that always predicts the expected value of y, disregarding the input features, would get a R 2 score of 0.0.The r2 score is imported from sklearn.metrics import r2\_score.Here the r2\_score is calculated on both the training and testing dataset to see how well the training and testing data have been fitted .

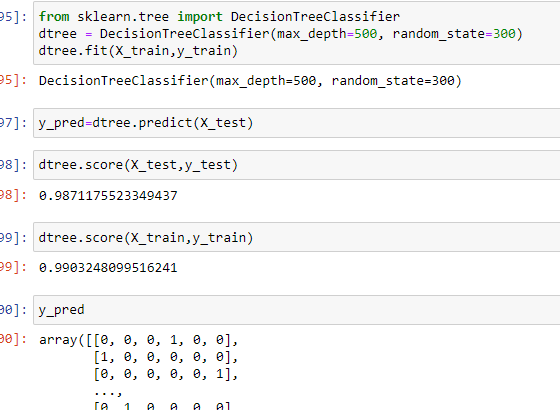
**CHAPTER 4**

**RESULTS AND DISCUSSIONS**

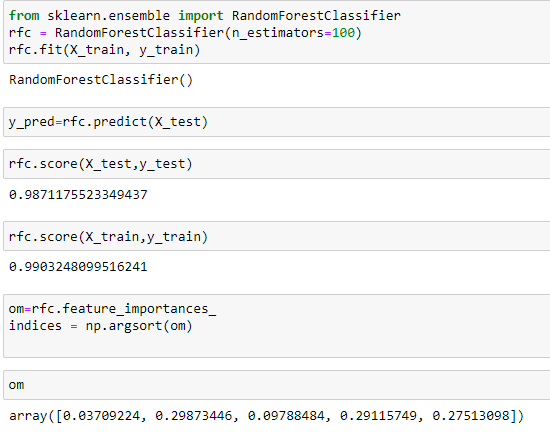
**4.1 RESULTS FROM ML ALGORITHMS:**

****

***FIG4.1.1 RESULT FROM KNN ALGORITHM:***

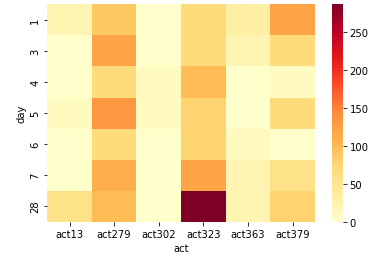
****

***4.1.2 RESULTS FROM DECISION TREE ALGORITHM***

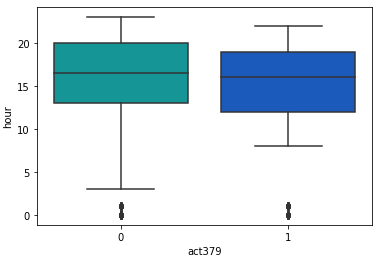


***FIG 4.1.3 RESULT FROM RANDOMFOREST ALGORITHM***

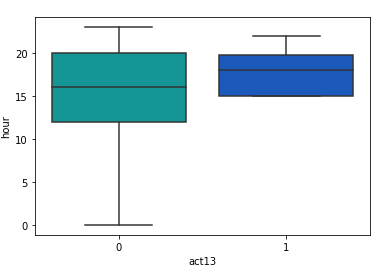
**4.2 RESULTS FROM VISUALIZATION:**

****

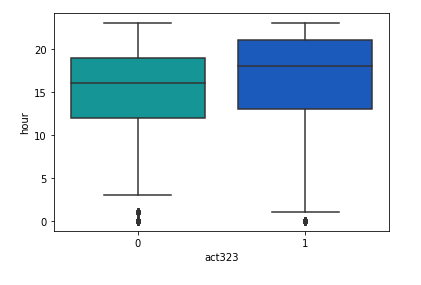
***FIG 4.2.1HEATMAP***

****

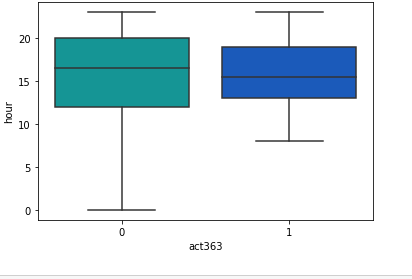
***FIG 4.2.2 ACT379 VS HOUR***

****

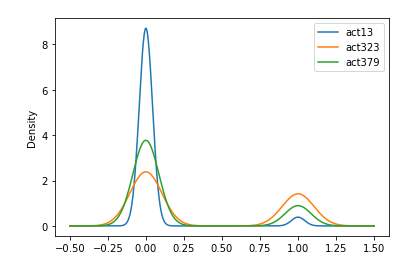
***FIG 4.2.3ACT13 VS HOUR***

****

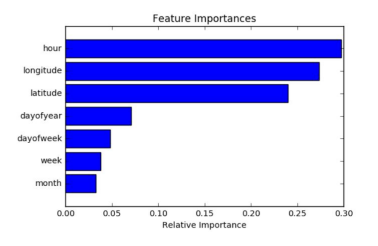
***FIG 4.2.4 ACT 323 VS HOUR***

****

***FIG 4.2.5 ACT 363 VS HOUR***

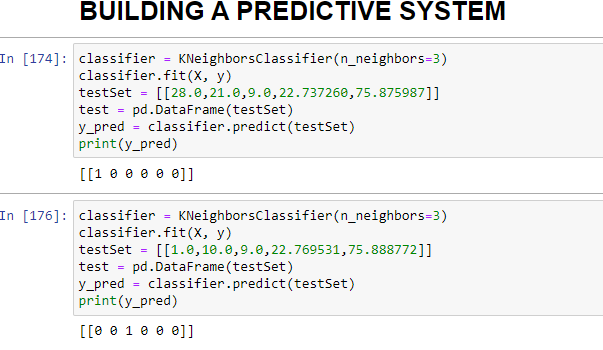
****

***FIG 4.2.6 COMPARING THE DENSITY BETWEEN COLUMNS***

****

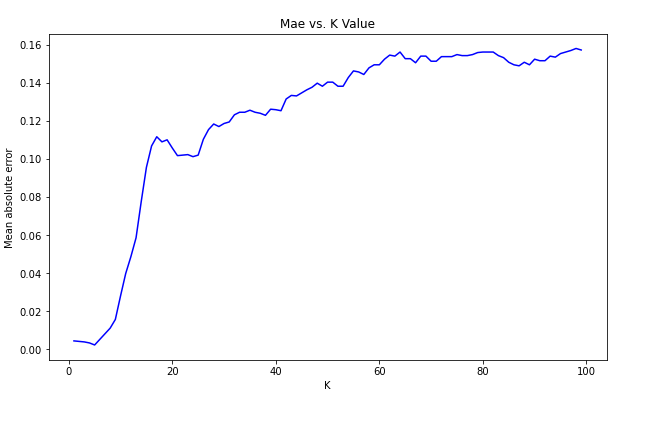
***FIG 4.2.7 FEATURE IMPORTANCE***

**4.3 RESULTS FROM PRDECTION:**



***FIG 4.3 PREDICTION RESULT***

**4.4 CALCULATING ERRORS**

****

***FIG 4.4 MAE VS K***

**CHAPTER 5**

**SUMMARY AND FUTURE SCOPE**

**5.1 SUMMARY:**

The dataset was successfully modeled by the the three machine learning classification algorithms. The results of the visualization shows the heatmap by which we can understand which type of crime was happening the most at what day .The ml model from knn is successful in predicting the type of crime when we give a new input to the model as we observed in the prediction result.

This project presents a method to predict and forecast crimes within a city. It focuses on having a crime prediction tool that can be helpful to law enforcement. Hence I tried to increase the prediction accuracy as much as possible. Along the way, we got to know the patterns of criminal activities in various areas which will be helpful for criminal investigation. This pattern has much greater importance than we realize. The KNN system assists law enforcement agencies for improved and accurate crime analysis. By seeing through the crime data we have to identify new factors that lead to crime. Since we are considering only some limited factors full accuracy cannot be achieved. For getting better results in prediction we have to find more crime attributes of places instead of fixing certain attributes.

**5.2 FUTURE SCOPE:**

The future scope of this project is to bulid a website that can show digitally the crime spots and type of crime with included attributes and also The goal of any society shouldn’t be to just catch criminals but to prevent crimes from happening in the first place.

**•Predicting Future Crime Spots:** By using historical data and observing where recent crimes took place we can predict where future crimes will likely happen. For example a rash of burglaries in one area could correlate with more burglaries in surrounding areas in the near future. System highlights possible hotspots on a map the police should consider patrolling more heavily

**•Predicting Who Will Commit a Crime**: Using Face Recognition to predict if a individual will commit a crime before it happens. The system will detect if there are any suspicious changes in their behavior or unusual movements. For example if an individual seems tobe walking back and forth in a certain area over and over indicating they might be a pickpocket or casing the area for a future crime. It will also track individual over time.

**•Pretrial Release and Parole**: After being charged with a crime, most individuals are released until they actually stand trial. In the past deciding who should be released pretrial or what an individual’s bail should be set at is mainly now done by judges using their best judgment. In just a few minutes, judges had to attempt to determine if someone is a flight risk, a serious danger to society, or at risk to harm a witness if released. It is an imperfect system open to bias. The media organization’s analysis indicated the system might indirectly contain a strong racial bias. They found, “That black defendants who did not recidivate over a two-year period were nearly twice as likely to be misclassified as higher risk compared to their white counterparts (45 percent vs. 23 percent).” The report raises the question of whether better AI/ML can eventually produce more accurate predictions or if it would reinforce existing problems. Any system will be based off of real world data, but if the real world data is generated by biased police officers, it can make the AI/ML biased

**REFERENCES:**

[1] Kim, Suhong, Param Joshi, Parminder Singh Kalsi, and Pooya Taheri. "Crime Analysis Through Machine Learning." In 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), pp. 415-420. IEEE, 2018.

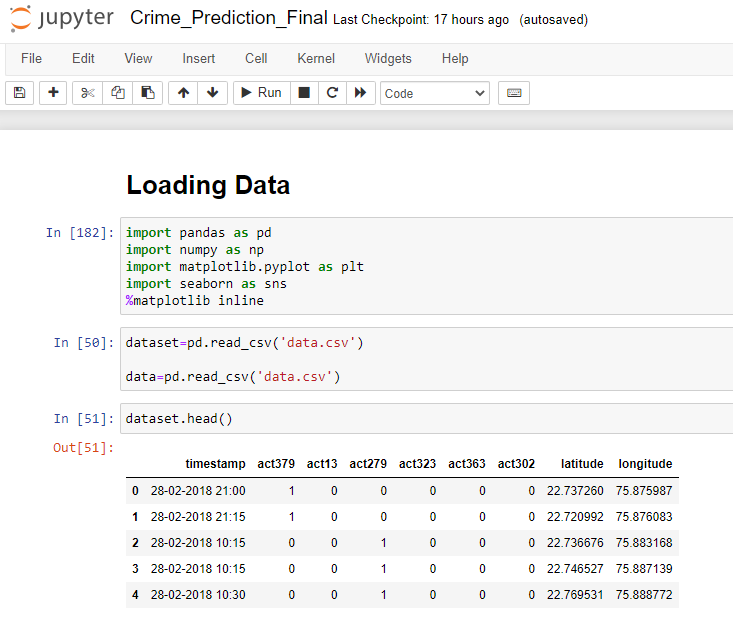
[2] Bogomolov, Andrey and Lepri, Bruno and Staiano, Jacopo and Oliver, Nuria and Pianesi, Fabio and Pentland, Alex.2014. Once upon a crime: Towards crime prediction from demographics and mobile data, Proceedings of the 16th International Conference on Multimodal Interaction.

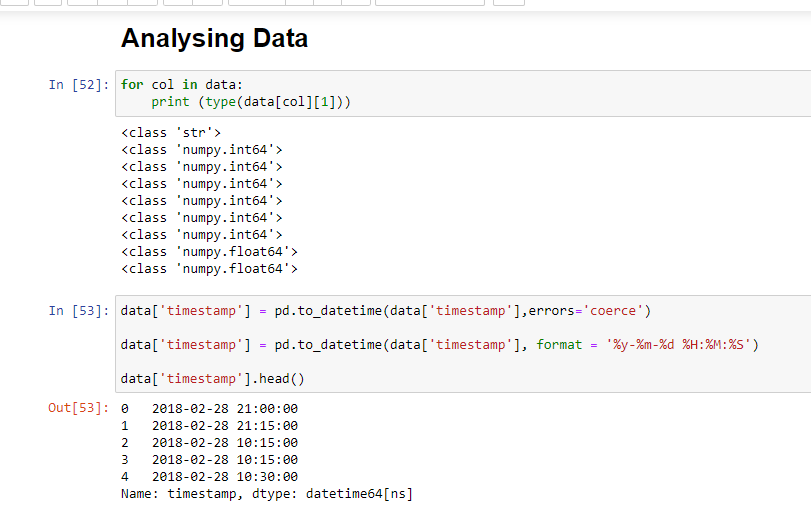
[3] Shah, Riya Rahul. "Crime Prediction Using Machine Learning." (2003).

[4]LeoBreiman, Random Forests, Machine Learning, 2001,Volume 45, Number 1, Page5

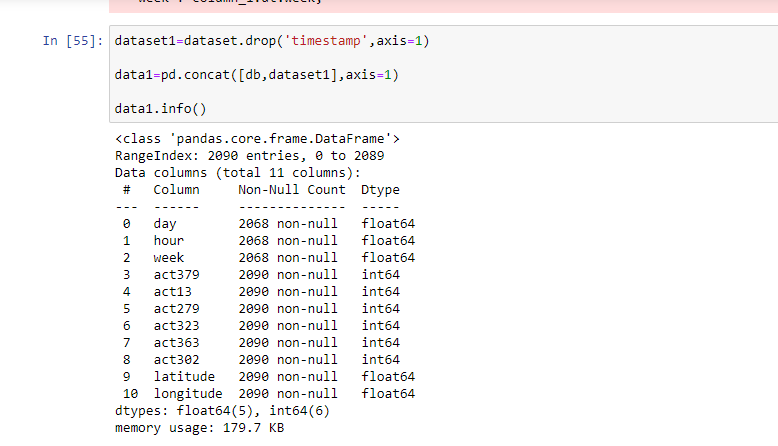
**APPENDIX:**

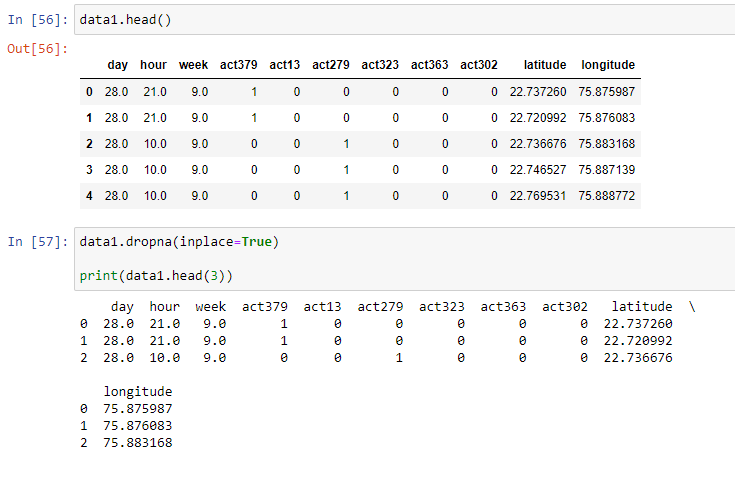
**A.SCREENSHOTS:**

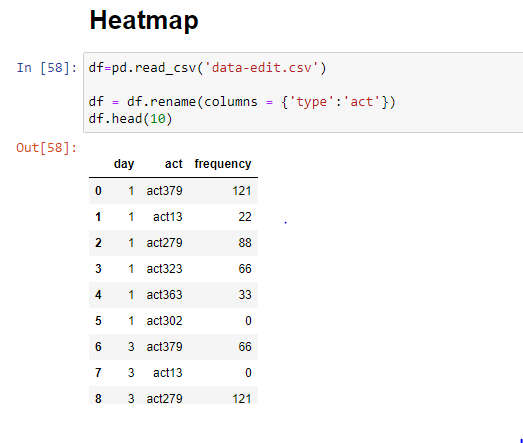
****

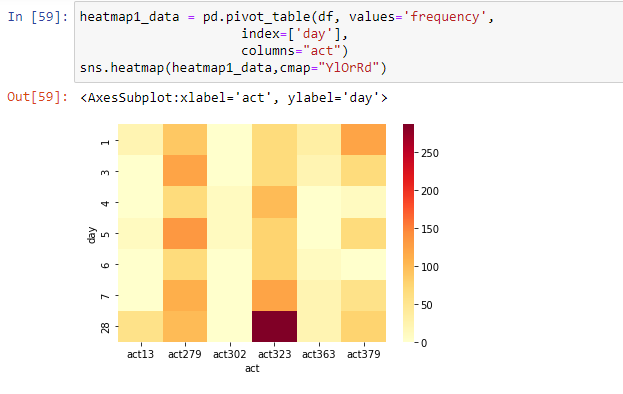
****

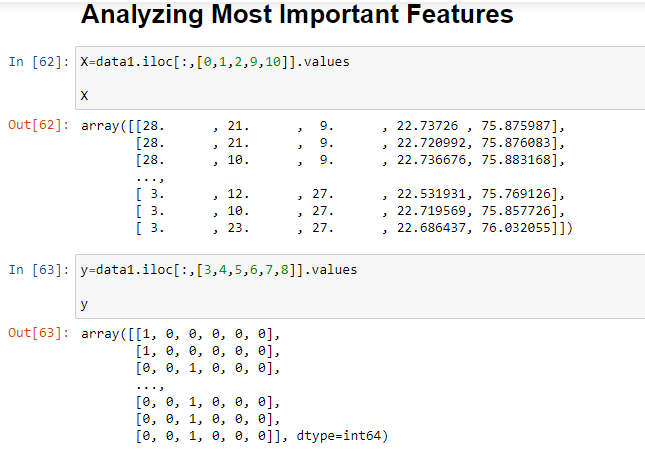
****

****

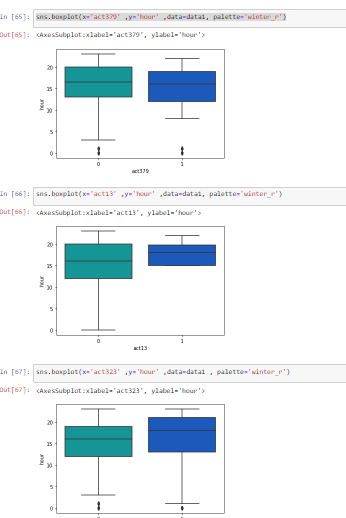
****

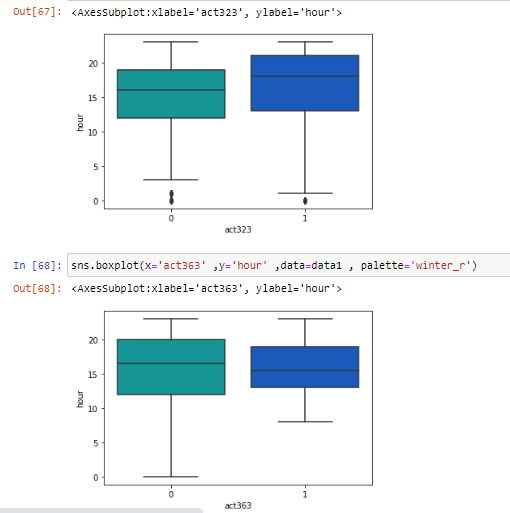
****

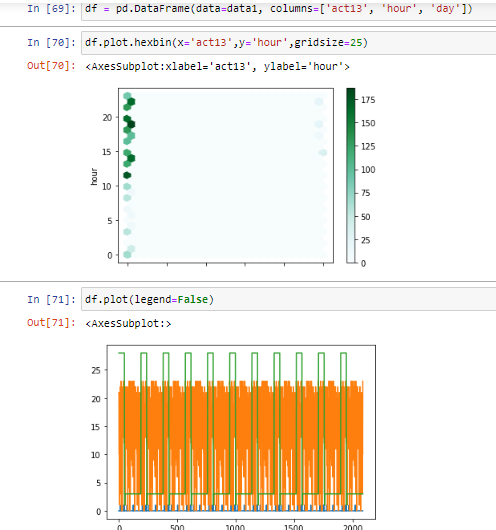
****

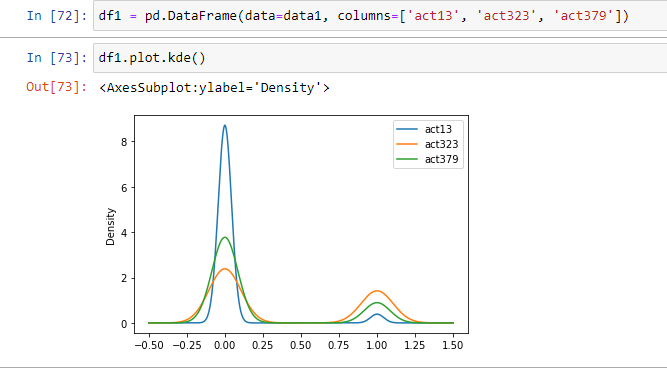
****

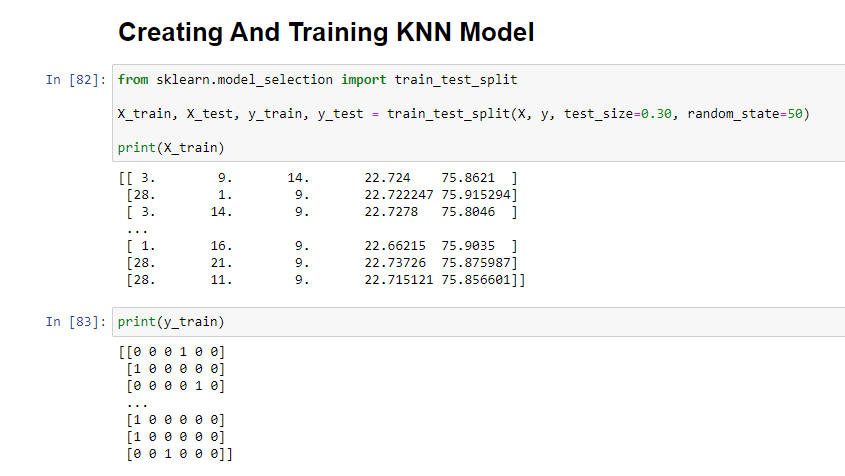
****

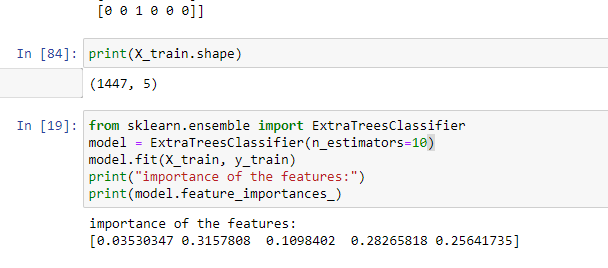
****

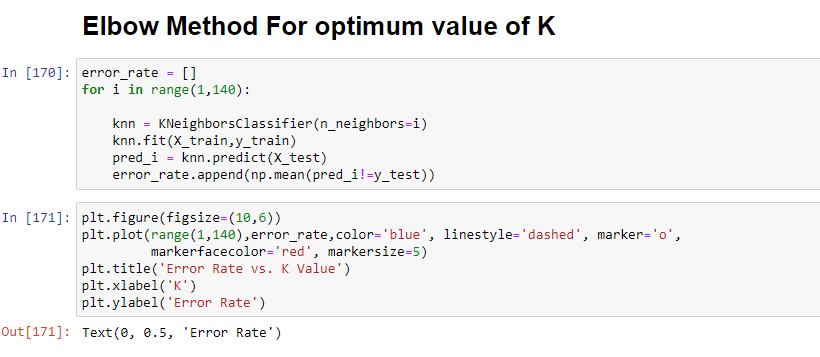
****

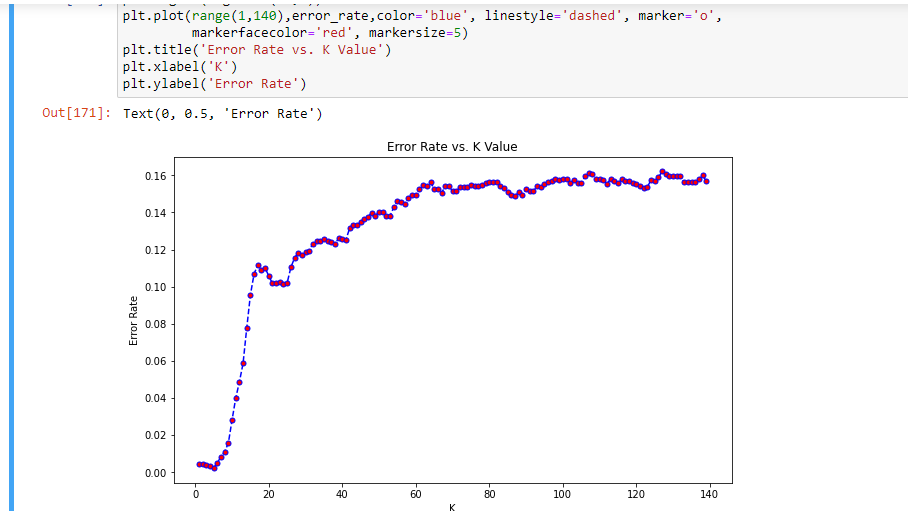
****

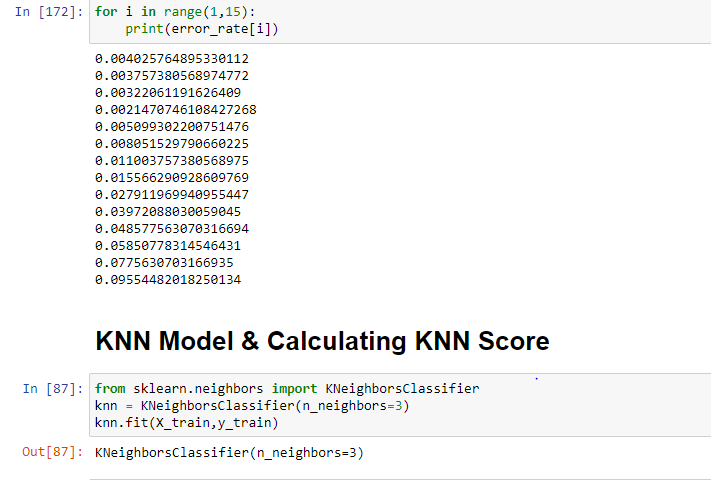
****

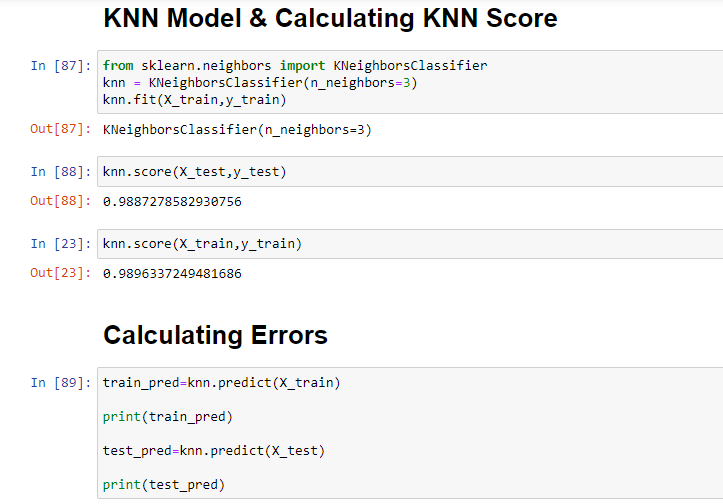
****

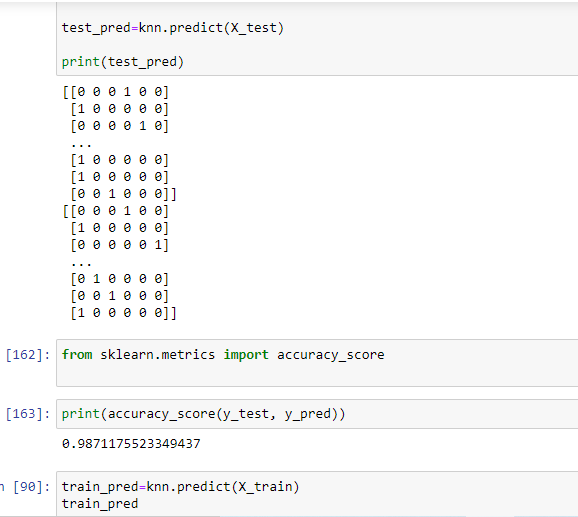
****

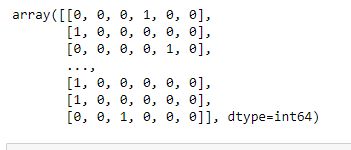
****

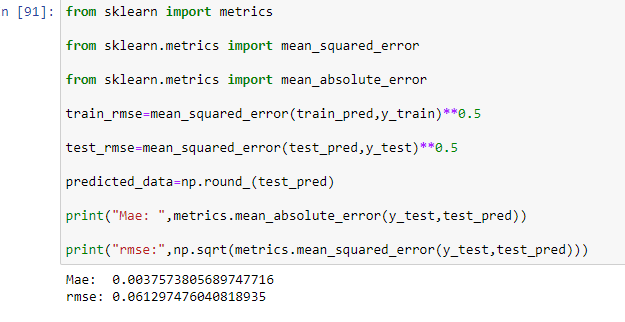
****

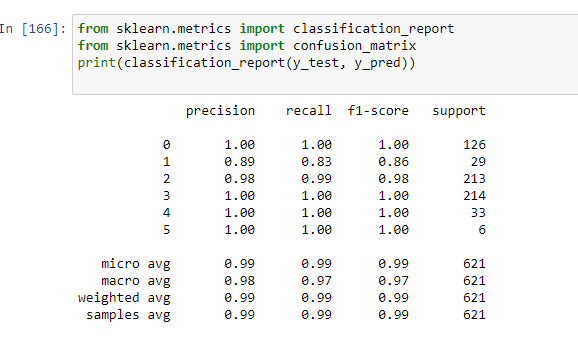
****

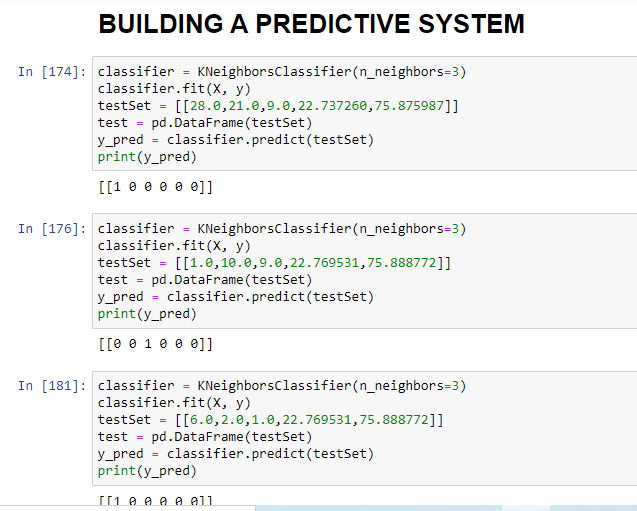
****

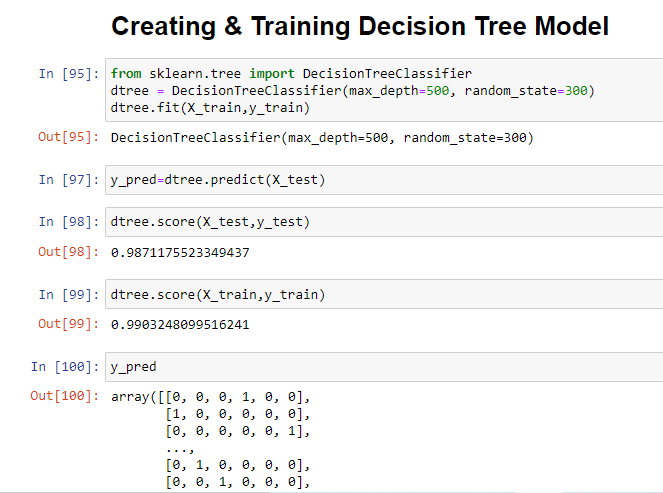
****

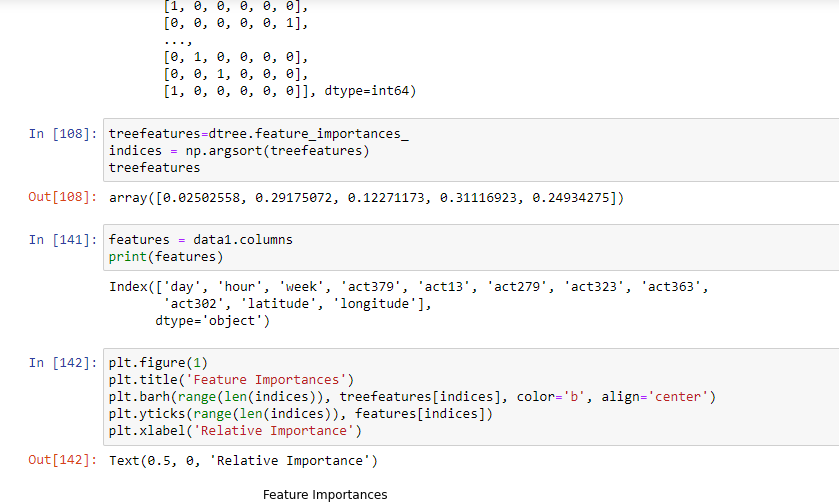
****

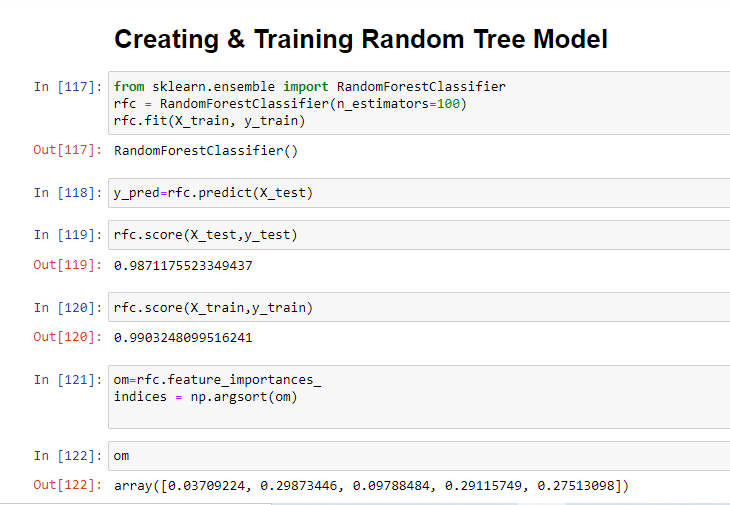
****

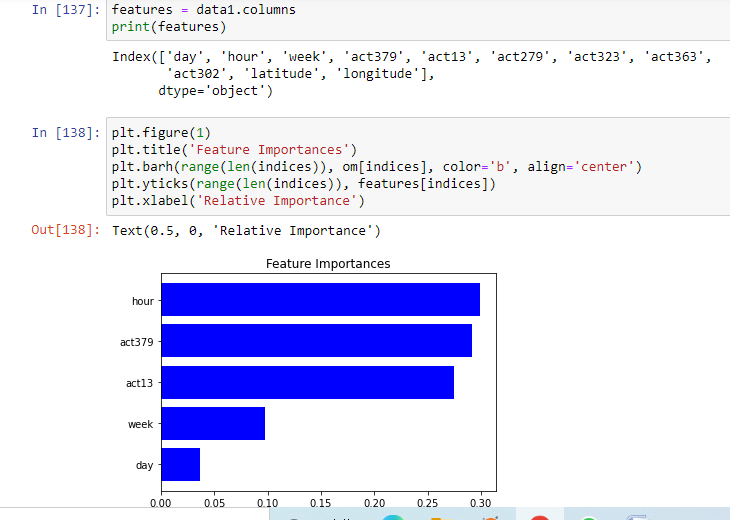
****

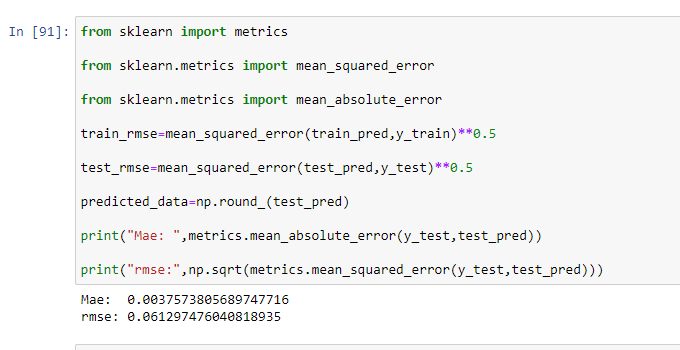
****

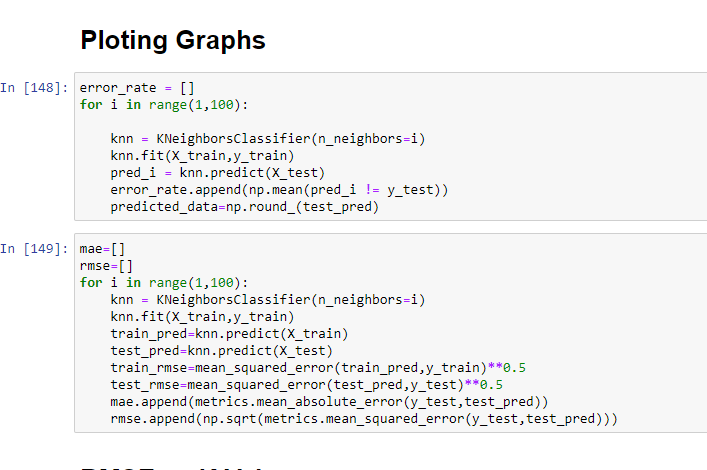
****

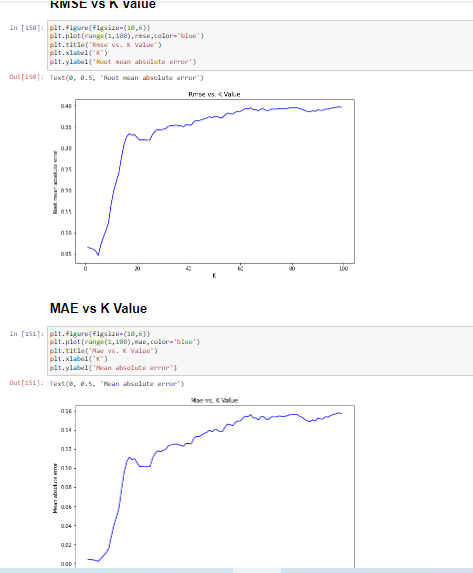
****

****

****

****

****

****

**SOURCE CODE:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

dataset=pd.read\_csv('data.csv')

data=pd.read\_csv('data.csv')

dataset.head()

for col in data:

print (type(data[col][1]))

data['timestamp'] = pd.to\_datetime(data['timestamp'],errors='coerce')

data['timestamp'] = pd.to\_datetime(data['timestamp'], format = '%y-%m-%d %H:%M:%S')

data['timestamp'].head()

column\_1 = data.iloc[:,0]

db=pd.DataFrame({#"year": column\_1.dt.year,

#"month": column\_1.dt.month,

"day": column\_1.dt.day,

"hour": column\_1.dt.hour,

#"dayofyear": column\_1.dt.dayofyear,

"week": column\_1.dt.week,

#"weekofyear": column\_1.dt.weekofyear,

#"dayofweek": column\_1.dt.dayofweek,

#"weekday": column\_1.dt.weekday,

#"quarter": column\_1.dt.quarter,

})

dataset1=dataset.drop('timestamp',axis=1)

data1=pd.concat([db,dataset1],axis=1)

data1.info()

ata1.dropna(inplace=True)

print(data1.head(3))

df=pd.read\_csv('data-edit.csv')

df = df.rename(columns = {'type':'act'})

df.head(10)

heatmap1\_data = pd.pivot\_table(df, values='frequency',

index=['day'],

columns="act")

sns.heatmap(heatmap1\_data,cmap="YlOrRd")

X=data1.iloc[:,[0,1,2,9,10]].values

X

y=data1.iloc[:,[3,4,5,6,7,8]].values

y

sns.pairplot(data1,hue='act363')

sns.boxplot(x='act13' ,y='hour' ,data=data1, palette='winter\_r')

sns.boxplot(x='act379' ,y='hour' ,data=data1, palette='winter\_r')

sns.boxplot(x='act323' ,y='hour' ,data=data1, palette='winter\_r')

sns.boxplot(x='act363' ,y='hour' ,data=data1, palette='winter\_r')

df = pd.DataFrame(data=data1, columns=['act13', 'hour', 'day'])

df.plot.hexbin(x='act13',y='hour',gridsize=25)

df1 = pd.DataFrame(data=data1, columns=['act13', 'act323', 'act379'])

df1.plot.kde()

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.30, random\_state=50)

print(X\_train)

print(y\_train)

from sklearn.ensemble import ExtraTreesClassifier

model = ExtraTreesClassifier(n\_estimators=10)

model.fit(X\_train, y\_train)

print("importance of the features:")

print(model.feature\_importances\_)

error\_rate = []

for i in range(1,140):

knn = KNeighborsClassifier(n\_neighbors=i)

knn.fit(X\_train,y\_train)

pred\_i = knn.predict(X\_test)

error\_rate.append(np.mean(pred\_i!=y\_test))

plt.figure(figsize=(10,6))

plt.plot(range(1,140),error\_rate,color='blue', linestyle='dashed', marker='o',

markerfacecolor='red', markersize=5)

plt.title('Error Rate vs. K Value')

plt.xlabel('K')

plt.ylabel('Error Rate')

for i in range(1,15):

print(error\_rate[i])

from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n\_neighbors=3)

knn.fit(X\_train,y\_train)

knn.score(X\_test,y\_test)

knn.score(X\_train,y\_train)

train\_pred=knn.predict(X\_train)

print(train\_pred)

test\_pred=knn.predict(X\_test)

print(test\_pred)

from sklearn.metrics import accuracy\_score

print(accuracy\_score(y\_test, y\_pred))

train\_pred=knn.predict(X\_train)

train\_pred

train\_pred=knn.predict(X\_train)

train\_pred

from sklearn import metrics

from sklearn.metrics import mean\_squared\_error

from sklearn.metrics import mean\_absolute\_error

train\_rmse=mean\_squared\_error(train\_pred,y\_train)\*\*0.5

test\_rmse=mean\_squared\_error(test\_pred,y\_test)\*\*0.5

predicted\_data=np.round\_(test\_pred)

print("Mae: ",metrics.mean\_absolute\_error(y\_test,test\_pred))

print("rmse:",np.sqrt(metrics.mean\_squared\_error(y\_test,test\_pred)))

from sklearn.metrics import classification\_report

from sklearn.metrics import confusion\_matrix

print(classification\_report(y\_test, y\_pred))

classifier = KNeighborsClassifier(n\_neighbors=3)

classifier.fit(X, y)

testSet = [[28.0,21.0,9.0,22.737260,75.875987]]

test = pd.DataFrame(testSet)

y\_pred = classifier.predict(testSet)

print(y\_pred)

from sklearn.tree import DecisionTreeClassifier

dtree = DecisionTreeClassifier(max\_depth=500, random\_state=300)

dtree.fit(X\_train,y\_train)

y\_pred=dtree.predict(X\_test)

dtree.score(X\_test,y\_test)

dtree.score(X\_train,y\_train)

y\_pred

treefeatures=dtree.feature\_importances\_

indices = np.argsort(treefeatures)

treefeatures

lt.figure(1)

plt.title('Feature Importances')

plt.barh(range(len(indices)), treefeatures[indices], color='b', align='center')

plt.yticks(range(len(indices)), features[indices])

plt.xlabel('Relative Importance')

from sklearn.ensemble import RandomForestClassifier

rfc = RandomForestClassifier(n\_estimators=100)

rfc.fit(X\_train, y\_train)

y\_pred=rfc.predict(X\_test)

rfc.score(X\_test,y\_test)

rfc.score(X\_train,y\_train)

om=rfc.feature\_importances\_

indices = np.argsort(om)

om

error\_rate = []

for i in range(1,100):

knn = KNeighborsClassifier(n\_neighbors=i)

knn.fit(X\_train,y\_train)

pred\_i = knn.predict(X\_test)

error\_rate.append(np.mean(pred\_i != y\_test))

predicted\_data=np.round\_(test\_pred)

mae=[]

rmse=[]

for i in range(1,100):

knn = KNeighborsClassifier(n\_neighbors=i)

knn.fit(X\_train,y\_train)

train\_pred=knn.predict(X\_train)

test\_pred=knn.predict(X\_test)

train\_rmse=mean\_squared\_error(train\_pred,y\_train)\*\*0.5

test\_rmse=mean\_squared\_error(test\_pred,y\_test)\*\*0.5

mae.append(metrics.mean\_absolute\_error(y\_test,test\_pred))

rmse.append(np.sqrt(metrics.mean\_squared\_error(y\_test,test\_pred)))

plt.figure(figsize=(10,6))

plt.plot(range(1,100),rmse,color='blue')

plt.title('Rmse vs. K Value')

plt.xlabel('K')

plt.ylabel('Root mean absolute error')

plt.figure(figsize=(10,6))

plt.plot(range(1,100),mae,color='blue')

plt.title('Mae vs. K Value')

plt.xlabel('K')

plt.ylabel('Mean absolute error')