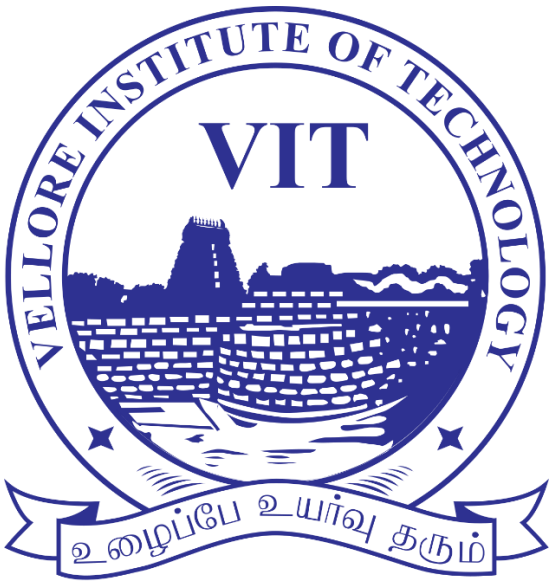


Crime Prediction System Using Machine Learning Algorithms

MACHINE LEARNING(CSE4020)

J-COMPONENT

FACULTY: Prof. Jaisakthi S M

SLOT: B2

TEAM MEMBERS:

Aryan Agarwal (17BCE0600)

Rohan Gupta (17BCE0717)

VIDEO LINK:

https://drive.google.com/file/d/1ozYQuqmAC8W3C4Sh4XABaWHu3ib-Zlnz/view?usp=sharing

1. ABSTRACT

In this project, we have created a Crime Prediction System. Instead of taking a dataset from the internet we have created a hypothetical data which mainly comprises of popular city areas and the different types of crimes taking place. With this dataset we have used machine learning algorithms to then analyse it and predict the next step of patterns. We have used 2 algorithms: Apriori and Naive Bayesian classifier. The Apriori is used to analyse data and predict the chance of a particular crime happening whereas naive Bayesian classifier is used to predict the crime that can happen in a particular area at a certain time period (e.g. morning, afternoon, evening, etc). The result obtained can be used to make people be aware of certain areas at a particular time period.

2. MOTIVATION

It is quite obvious that the rate of crime is increasing day by day and there is no particular way our government has come up with to stop these crimes. It has become almost impossible for a person to walk on a street without the fear of getting mugged. So, as to come up with a solution to this problem we have designed a system that will make the user aware of all the crimes that take place in that particular area so that he can go with particular protection. With this system, people can look out for themselves as they will get to know the dangerous locations and be more careful.

3. OBJECTIVES

- We aim to find the possible criminal hotspots using a hypothetical dataset of a city.
- We aim to find the type of crime that can possibly happen at a particular during a certain time period.
- We also aim to calculate the percentage chances of a crime happening.

4. INTRODUCTION

Crimes are something that disturbs the daily life of an individual and sometimes has small or a big impact on the society depending on the type of crimes. It is a key factor that people usually consider when choosing new cities or areas to move to or to visit other places. With increase in criminal activities all around the world, law and police forces are starting to request progressed geographic data frameworks and are using data mining techniques to examine different crimes and protect their areas.

Regardless of the fact that crimes can happen anywhere, criminals usually do crimes where there is less crowd and at place where it’s usually hard to find out. By using a data mining way to decide and choose criminal hotspots and then sort them according to the places, time and crime type, we want to raise the people’s awareness about the

dangerous areas at a specific time. This can help people a lot to avoid these dangerous areas when choosing them to visit or to move to. Furthermore, having this kind of learning would help people with improving their living area choices. On the other hand, the police can use this solution to increase the no. of personnel to deploy at that area to counter the criminal activities. Furthermore, this might prove to be very resourceful to other law enforcement organizations. It can be helpful in resource allocation of police at a certain area with high crime rate at a time than an area with low or no crime at all. We can have all this information open, so that we can create a more secured community than ever for the habitants and those who are going to move there.

5. LITERATURE SURVEY

There has been incalculable of work done regarding crime. Extensive datasets have been reviewed and data such as area and the kind of crime have been analysed to enable individuals to pursue law authorizations. Existing techniques have utilized these databases to recognize crime hotspots depending on areas. Despite the fact that crime areas have been recognized, there is no data accessible that incorporates the crime event date and time alongside systems that can precisely anticipate what crimes will happen later on.
We analysed some of the past work and the research papers regarding crime prediction. Below is a brief description of some of the past works.

An overview on crime prediction methods

2017 6th ICT International Student Project Conference (ICT-ISPC)

This paper has introduced various crime prediction methods. It discusses about "Support Vector Machine", "Fuzzy Theory", "Artificial Neural Network" and "Multivariate Time Series". This research work focuses on reviewing a crime prediction analysis tool for many scenarios using particulars crime prediction methods that can help law enforcement to efficiently handle crime incidents. *Support Vector machine* (SVM) has performed well in prediction of time series crimes because they can model nonlinear relations in a stable and efficient way. SVM is applied to predict crime hotspots resulting in a global solution. *Fuzzy Theory modelling* requires numerical inputs and uses IF-THEN regulations to form vague prediction and quadratic combination of the presumptive variables. It requires less computational complexity and has well in learning abilities. The characteristic of the fuzzy modelling is used to improve the prediction efficiency. *Artificial Neural Network* (ANN) depends on the prediction by keenly investigating the pattern from an effectively existing voluminous historical set of data. *Multivariate Time Series* is one of the statistical tools to study the behaviour of time-dependent data and predict the future values based on history of variations in the data. This method was applied in testing of statistical significance of multivariate time series analysis technique [17].

Crime prediction and forecasting in TamilNadu using clustering approaches

2016 International Conference on Emerging Technological Trends (ICETT)

This research implements *KNN classification* that searches through the data to highlight similar instance when an input is given to it. The paper discusses about *K-Means Clustering* method to provide a large criminal data and simplify the records and ease in handling, searching and retrieving. *Agglomerative Hierarchical Clustering* assigns each object to its cluster and then integrates these clusters to form a larger cluster. DBSCAN is based on density clustering method. The algorithm develops areas with appropriately high density into clusters and finds clusters of arbitrary shape in spatial databases with noise. It defines a cluster as a maximal set of density-connected points [16].

Crime tracer: Activity space-based crime location prediction

2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)

The research concludes through Crime Pattern Theory that offenders, rather than venture into unknown territory, frequently commit opportunistic crimes and serial violent crimes by exploiting openings they experience in spots they are most comfortable with as a feature of their activity space, which includes the most frequently visited places as determined by a person's daily routine activities, such as commuting patterns. *Random walk* over a graph shows how offenders encounter criminal opportunities, given that the behaviour of a random walk model is local. *Starting Probabilities* provides anchor locations of all the offenders' location, linking them and thus compute probability of each anchor location of two primary factors. Offender-based CF. The intuition behind the offender-based CF approach (OCF) is that offenders who had similar behaviour in the past will have similar behaviour in the future [15].

A novel serial crime prediction model based on Bayesian learning theory

2010 International Conference on Machine Learning and Cybernetics

Geographic Profile of a Specific Factor uses Discrete Distance Decay Function and Gaussian model to calculate its probability distribution in the region. *Dynamic Prediction Model based on Bayesian Learning Method* for a specific offender, the hypothesis above may not stand. Therefore, it needs to adjust the effect function dynamically based on the dynamical known crime data, especially data of crime sites. The paper brings forward an adaptive adjustment algorithm and put it in a Bayesian learning framework [14].

Classify interval range of crime forecasting for crime prevention decision making

2016 11th International Conference on Knowledge, Information and Creativity Support Systems (KICSS)

The Decision Support System (DSS) is intuitive, computer-based frameworks and subsystems intended to help decision-makers in utilizing communication technologies, information, archives, learning or potentially models to finish decision process tasks, explain in. They also stated that DSS can improve the effectiveness and quality of decision making by processing a lot of data and providing alternative solutions to various problems. Crime analysis is done first by identifying crime incidents, victim profile and potential risk, monitoring crime threshold, and examine crime trends. The research develops and application tool using Input Data module, Statistic Module and Crime prevention Decision Module and uses If-Then rules to forecast and associate qualitative attributes of crime parameters [13].

6. PROPOSED METHODOLOGY

For this project we have used two machine learning algorithms, the algorithms are used to find the frequent crime patterns and predicting dangerous hotspots. The methodology that we have used focuses mainly on three elements from the crime datasets which are, the type of crime, the time period at which crime happened and the location of the crime. Starting with the first algorithm i.e. Apriori, we are using it to predict the probability of the crime happening. Second algorithm i.e. Naïve Bayes Classifier, is used to predict the crime type by using the time period and location.

Apriori Algorithm

It is one of the basic machine learning algorithms for finding frequent patterns. This algorithm scans the whole database to find the item sets that satisfy a predefined minimum support. In this project, the support is taken as two meaning that itemsets with count value more than 2 are the only ones to be considered. The probability is also limited to 25% meaning only when the probability is more than 25% it will be shown.

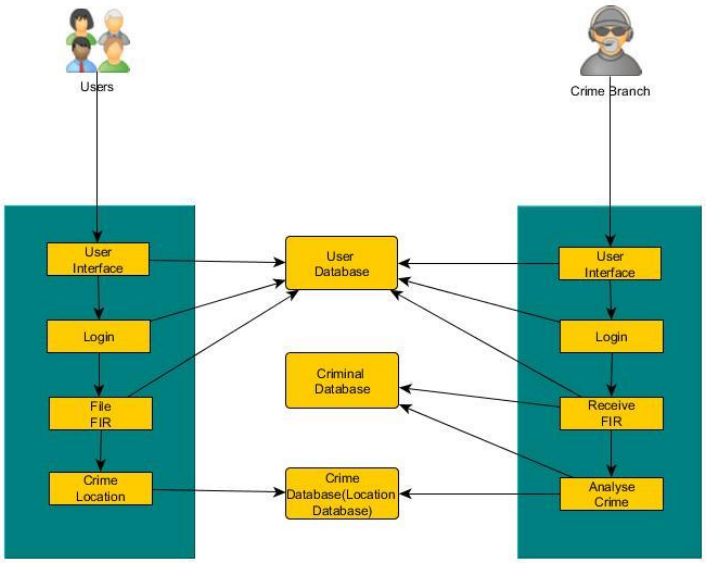
Naïve Bayes Classifier

It is an effective and widely used algorithm. It predicts class membership probabilities using Bayes theorem. The algorithm assumes all features to be conditionally independent which not always is the case so the prediction might not be hundred percent accurate.

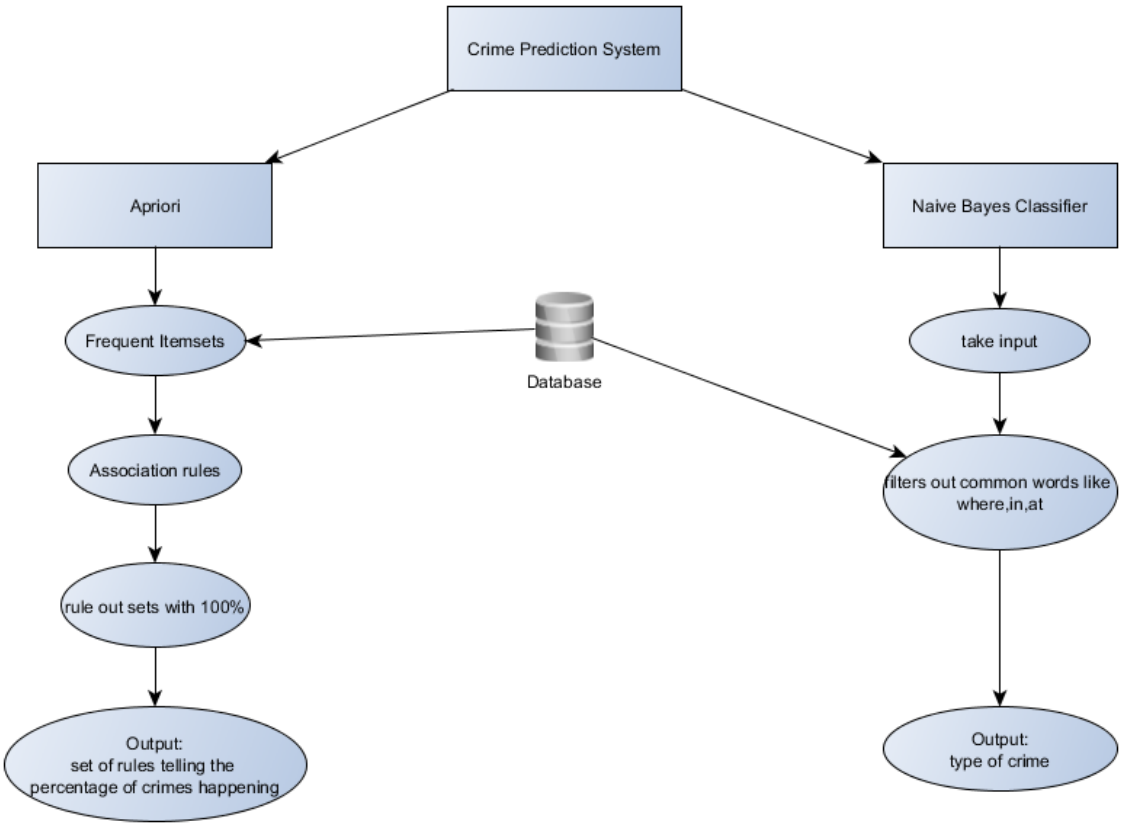
6.1 DATASET DESCRIPTION

Instead of taking dataset from the internet, we are taking hypothetical data manually, i.e. The user can add data to the system.

6.2 ARCHITECTURE DIAGRAM

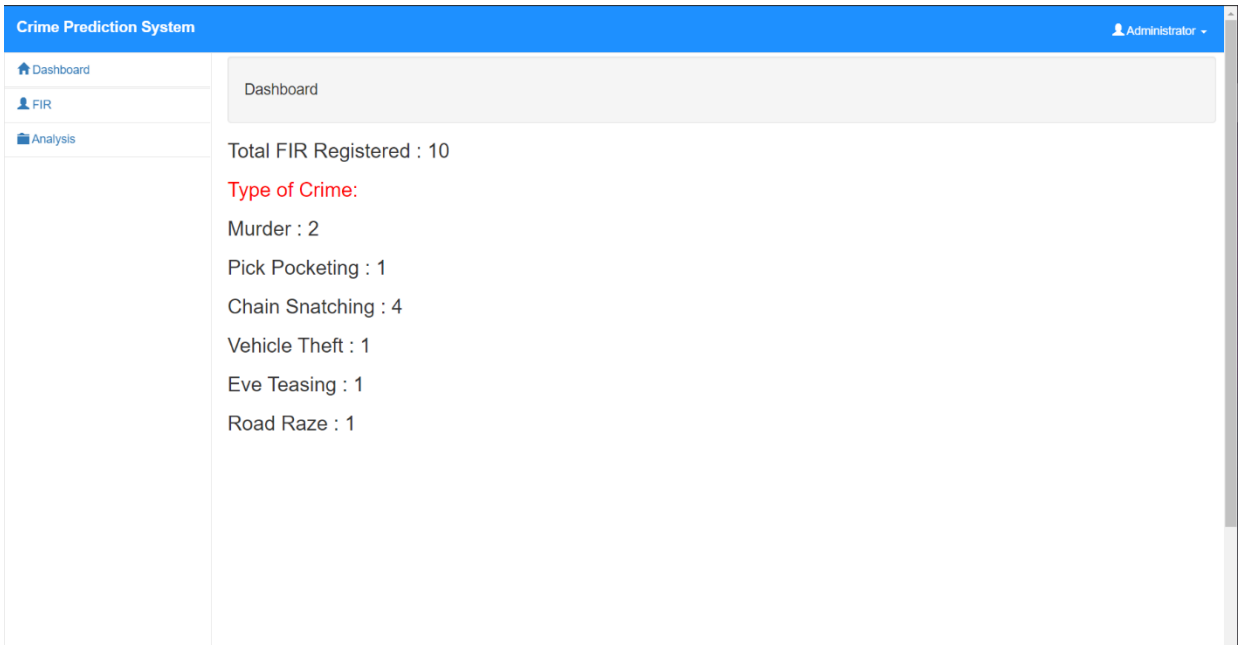


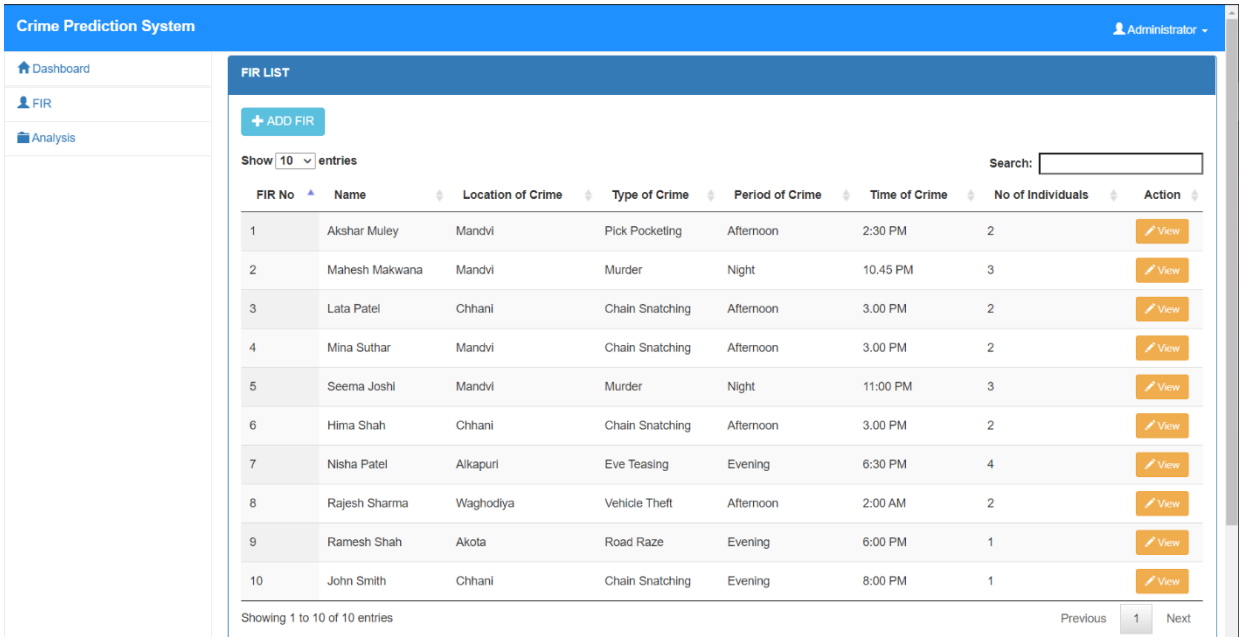
6.3 MODULE DIAGRAM



6.4 EXPERIMENTAL SETUP

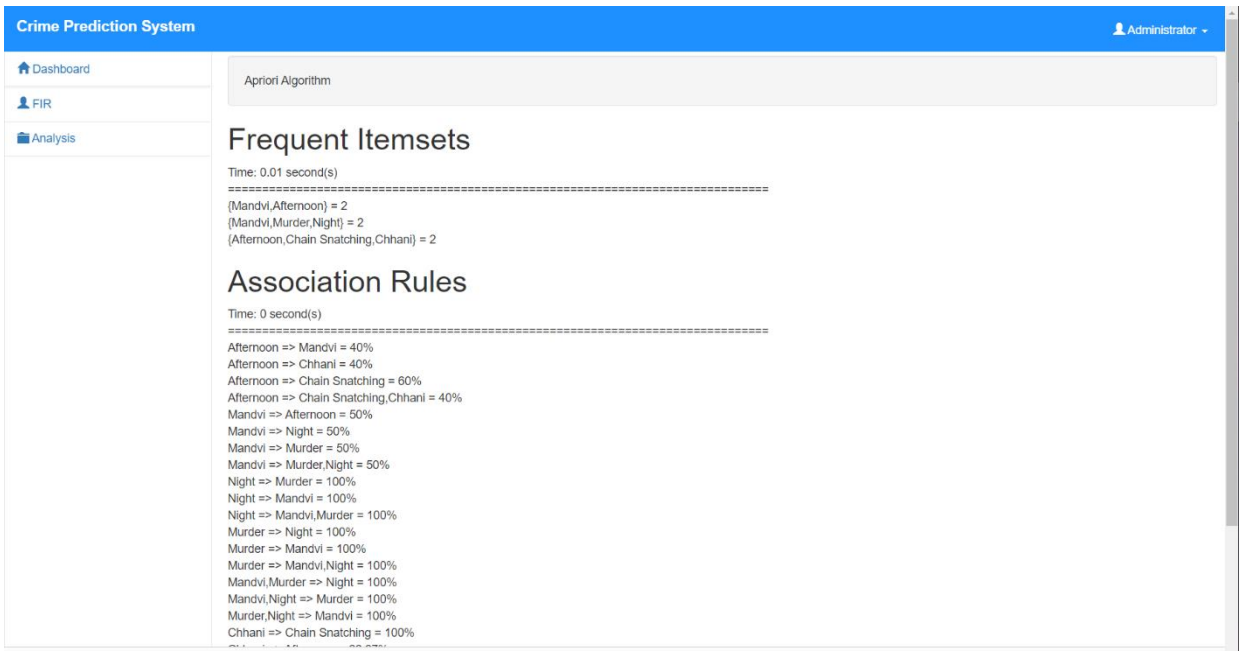
Starting up the project, we're introduced to a login page. After logging in we are taken to the Dashboard page where the total no. of FIRs recorded and the Different types of crimes are displayed. The FIR tab shows the database with all the registered records, here you can also add a new FIR.



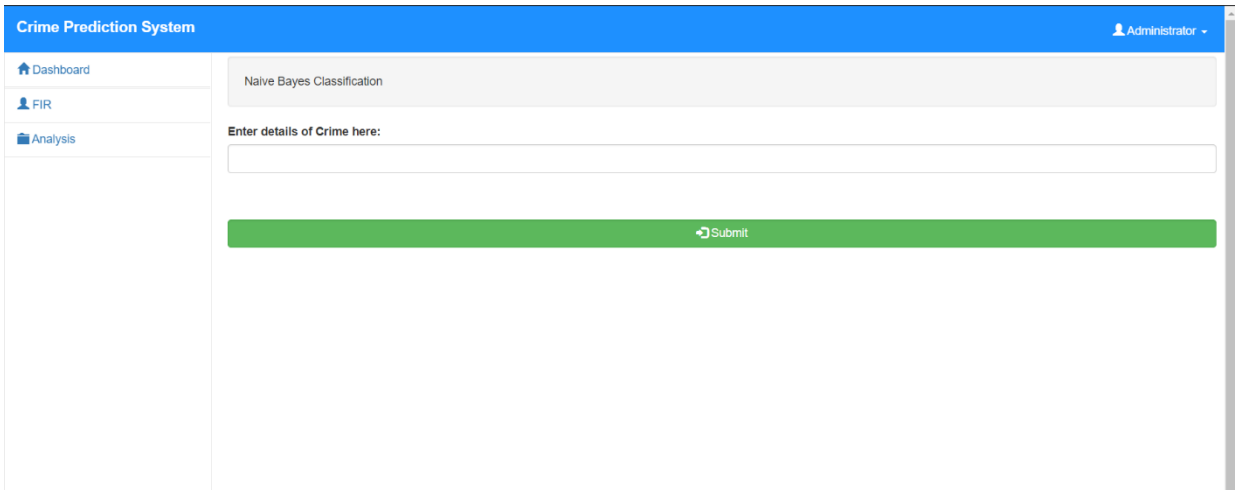


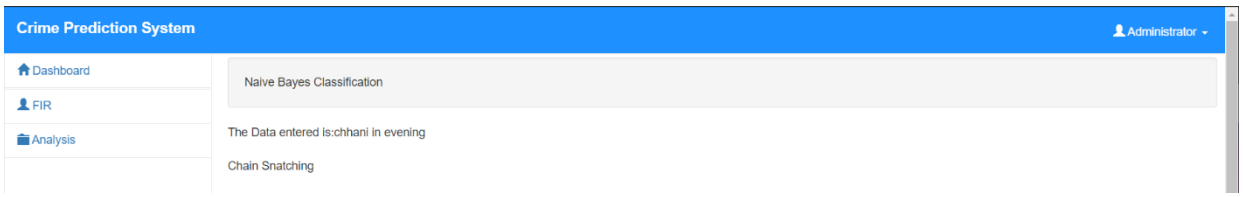
After this we can click the Analysis tab and select the type of algorithm that we want to use. Here we have two algorithms, 1. Apriori, 2. Naïve Bayes Classifier.

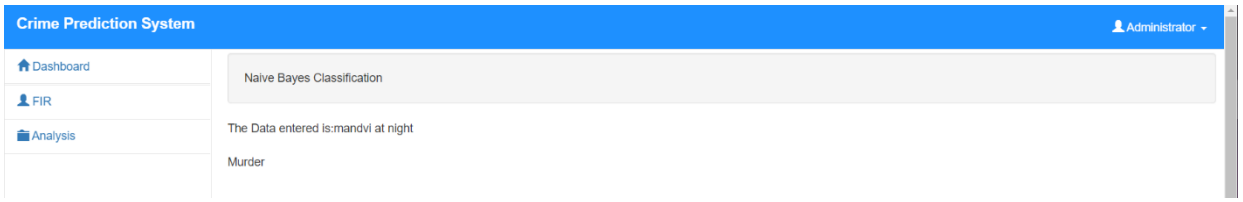
By selecting the Apriori algorithm we are presented with a page showing the Frequent Itemsets and the Association Rules. The Frequent Itemsets are calculated by using the most common pairs from the database. And as for the Association Rules, it shows the frequent patterns and the correlation of crimes. As seen that some of the Association Rules shows 100% probability of the crime happening, these rules are generally ignored as there's no proof that the probability of a particular crime happening is 100%.



And as for Naïve Bayes Classifier, we have used this algorithm as a searching algorithm but instead of it returning the count of a crime happening it returns the name of that crime. Example: by searching "Channi in evening" we are presented with result "chain snatching" and by searching "Mandvi at night" we get "Murder"







6.5 RESULT AND DISCUSSION

While we have used 2 different algorithms, we also learned that while they are effective, they are not fully reliable. First, we used the Apriori algorithm to find the frequent crime patterns meaning we analyse the database to get the possibilities of a crime happening. But the algorithm is not 100% accurate since in some of the association rules, the chances of a particular crime happening is not always 100%. The next algorithm, naïve Bayes classifier is used find the type of crime that can happen in the area at a certain time period by providing the location and time. However, this algorithm is used to provide only the crime name and not its probability of happening. And, another disadvantage of this is that if only one type of crime has happened in that location, the system will show only that name since it uses the whole database to give output. We came to the result that among these two, naïve Bayes is more accurate that Apriori in this project and also that the performance of each algorithm depends on the type of application it is used in.

7. CONCLUSION

In today's world when the crime percentage is increasing every day, a major challenge faced by law enforcement is predicting crime to protect the citizens. With the new technologies and techniques being created almost every day, these tools are now easily accessible to law enforcements.

The reason we took Apriori and naive Bayes classifier is when compared to other similar algorithms, we found that while FP-growth can be used instead of Apriori since the time complexity of Apriori is much higher than FP-growth, Apriori is much easier to understand and implement. And while comparing naive Bayes to KNN the F-measure (harmonic mean of precision) of naive Bayes was higher than KNN meaning that it's better.

This project was focused on creating a crime prediction system for the general public using different algorithms. This project can help law enforcements to categorize and analyse the crime to predict the possible future crimes at different areas.

As for the future of this project, the scope is much larger than we can think of. The project can be remade on a much larger scale with better technologies and more classification algorithms to increase its predicting accuracy leading to increase in its overall performance.

8. REFERENCES

[1] Almanie, T., Mirza, R., & Lor, E. (2015). Crime prediction based on crime types and using spatial and temporal criminal hotspots. *arXiv preprint arXiv:1508.02050*.

[2] Bogomolov, A., Lepri, B., Staiano, J., Oliver, N., Pianesi, F., & Pentland, A. (2014, November). Once upon a crime: towards crime prediction from demographics and mobile data. In *Proceedings of the 16th international conference on multimodal interaction* (pp. 427-434). ACM.

[3] Nasridinov, A., Ihm, S. Y., & Park, Y. H. (2013). A decision tree-based classification model for crime prediction. In *Information Technology Convergence* (pp. 531-538). Springer, Dordrecht.

[4] Liao, R., Wang, X., Li, L., & Qin, Z. (2010, July). A novel serial crime prediction model based on Bayesian learning theory. In *Machine Learning and Cybernetics (ICMLC), 2010 International Conference on* (Vol. 4, pp. 1757-1762). IEEE.

[5] Laalmanac.com, 'City of Los Angeles Planning Areas Map', 2015. [Online]. Available: http://www.laalmanac.com/LA/lamap3.htm. [Accessed: 20- May- 2015].

[6] Varunon9, naïve-bayes-classifier, (2018), GithubGist, https://github.com/varunon9/naive-bayes-classifier.

[7] Dave Smith, PHP Apriori Algorithm Data Miner, (2015), https://www.phpclasses.org/browse/file/61953.html.

[8] VTwo-Group, Apriori-Algorithm (2014, November), https://github.com/VTwo-Group/Apriori-Algorithm.

[9] Julian Finkler, PHP Decision Tree Classifier: Compose decision trees and evaluate subjects, (2017, July), https://www.phpclasses.org/package/10385-PHP-Compose-decision-trees-and-evaluate-subjects.html.

[10] S. Ahmad, S. P. Simonovic, "An Intelligent Decision support system for Management of Floods", Water Resources Management, vol. 20, pp. 391-410, 2006

[11] 2016 11th International Conference on Knowledge, Information and Creativity Support Systems (KICSS)

[12] 2010 International Conference on Machine Learning and Cybernetics

[13] 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)

[14] 2016 International Conference on Emerging Technological Trends (ICETT)

[15] 2017 6th ICT International Student Project Conference (ICT-ISPC)

9. APPENDIX

Apriori

```
<?php

class Apriori {
    private $delimiter   = ',';
    private $minSup      = 2;
    private $minConf     = 75;

    private $rules       = array();
    private $table       = array();
    private $allthings   = array();
    private $allsups     = array();
    private $keys        = array();
    private $freqItmsts  = array();
    private $phase       = 1;

    //maxPhase>=2
    private $maxPhase    = 20;

    private $fiTime      = 0;
    private $arTime      = 0;

    public function setDelimiter($char)
    {
      $this->delimiter = $char;
    }

    public function setMinSup($int)
    {
      $this->minSup = $int;
    }

    public function setMinConf($int)
    {
      $this->minConf = $int;
    }

    public function setMaxScan($int)
    {
      $this->maxPhase = $int;
    }
```

```
public function getDelimiter()
{
  return $this->delimiter;
}

public function getMinSup()
{
  return $this->minSup;
}

public function getMinConf()
{
  return $this->minConf;
}

public function getMaxScan()
{
  return $this->maxPhase;
}

private function makeTable($db)
{
  $table   = array();
  $arr   = array();
  $counter = 1;

  if(!is_array($db))
    {
      $db = file($db);
    }

  $num = count($db);
  for($i=0; $i<$num; $i++)
    {
      $tmp  = explode($this->delimiter, $db[$i]);
      $num1 = count($tmp);
      $x    = array();
      for($j=0; $j<$num1; $j++)
        {
          $x = trim($tmp[$j]);
          if($x==='')
            {
              continue;
            }
```

```
            if(!isset($this->keys['v->k'][$x]))
              {
                $this->keys['v->k'][$x]         = $counter;
                $this->keys['k->v'][$counter]   = $x;
                $counter++;
              }

            if(!isset($array[$this->keys['v->k'][$x]]))
              {
                $array[$this->keys['v->k'][$x]] = 1;
                $this->allsups[$this->keys['v->k'][$x]] = 1;
              }
            else
              {
                $array[$this->keys['v->k'][$x]]++;
                $this->allsups[$this->keys['v->k'][$x]]++;
              }

            $table[$i][$this->keys['v->k'][$x]] = 1;
          }
      }

    $tmp = array();
    foreach($array as $item => $sup)
      {
        if($sup>=$this->minSup)
          {

            $tmp[] = array($item);
          }
      }

    $this->allthings[$this->phase] = $tmp;
    $this->table = $table;
}

private function scan($arr, $implodeArr = '')
{
    $cr = 0;

    if($implodeArr)
      {
        if(isset($this->allsups[$implodeArr]))
          {
```

```
            return $this->allsups[$implodeArr];
          }
      }
    else
      {
        sort($arr);
        $implodeArr = implode($this->delimiter, $arr);
        if(isset($this->allsups[$implodeArr]))
          {
           return $this->allsups[$implodeArr];
          }
      }

    $num  = count($this->table);
    $num1 = count($arr);
    for($i=0; $i<$num; $i++)
      {
        $bool = true;
        for($j=0; $j<$num1; $j++)
          {
            if(!isset($this->table[$i][$arr[$j]]))
              {
                $bool = false;
                break;
              }
          }

        if($bool)
          {
            $cr++;
          }
      }

    $this->allsups[$implodeArr] = $cr;

   return $cr;
}

private function combine($arr1, $arr2)
{
    $result = array();

    $num  = count($arr1);
    $num1 = count($arr2);
    for($i=0; $i<$num; $i++)
```

```
      {
        if(!isset($result['k'][$arr1[$i]]))
          {
            $result['v'][] = $arr1[$i];
            $result['k'][$arr1[$i]] = 1;
          }
      }
    for($i=0; $i<$num1; $i++)
      {
        if(!isset($result['k'][$arr2[$i]]))
          {
            $result['v'][] = $arr2[$i];
            $result['k'][$arr2[$i]] = 1;
          }
      }

   return $result['v'];
}

private function realName($arr)
{
    $result = '';

    $num = count($arr);
    for($j=0; $j<$num; $j++)
      {
        if($j)
          {
            $result .= $this->delimiter;
          }

        $result .= $this->keys['k->v'][$arr[$j]];
      }

   return $result;
}
//1-2=>2-3 : false
//1-2=>5-6 : true
private function checkRule($a, $b)
{
    $a_num = count($a);
    $b_num = count($b);
    for($i=0; $i<$a_num; $i++)
      {
        for($j=0; $j<$b_num; $j++)
```

```
            {
              if($a[$i]==$b[$j])
                 {
                   return false;
                 }
            }
      }
  return true;
}
private function confidence($sup_a, $sup_ab)
{
    return round(($sup_ab / $sup_a) * 100, 2);
}

private function subsets($items)
{
   $result  = array();
   $num     = count($items);
   $members = pow(2, $num);
   for($i=0; $i<$members; $i++)
      {
        $b   = sprintf("%0".$num."b", $i);
        $tmp = array();
        for($j=0; $j<$num; $j++)
           {
             if($b[$j]=='1')
                {
                  $tmp[] = $items[$j];
                }
           }

        if($tmp)
           {
             sort($tmp);
             $result[] = $tmp;
           }
      }

  return $result;
}

private function freqItemsets($db)
{
   $this->fiTime = $this->startTimer();
   $this->makeTable($db);
```

```
    while(1)
      {
        if($this->phase>=$this->maxPhase)
          {
            break;
          }

        $num = count($this->allthings[$this->phase]);
        $cr  = 0;
        for($i=0; $i<$num; $i++)
          {
            for($j=$i; $j<$num; $j++)
              {
                if($i==$j)
                  {
                    continue;
                  }

                $item = $this->combine($this->allthings[$this->phase][$i],
$this->allthings[$this->phase][$j]);
                sort($item);
                $implodeArr = implode($this->delimiter, $item);
                if(!isset($this->freqItmsts[$implodeArr]))
                  {
                    $sup = $this->scan($item, $implodeArr);
                    if($sup>=$this->minSup)
                      {
                        $this->allthings[$this->phase+1][] = $item;
                        $this->freqItmsts[$implodeArr] = 1;
                        $cr++;
                      }
                  }
              }
          }

        if($cr<=1)
          {
            break;
          }

        $this->phase++;
      }

    foreach($this->freqItmsts as $k => $v)
      {
```

```
        $arr = explode($this->delimiter, $k);
        $num = count($arr);
        if($num>=3)
          {
            $subsets = $this->subsets($arr);
            $num1    = count($subsets);
            for($i=0; $i<$num1; $i++)
              {
                if(count($subsets[$i])<$num)
                  {
                    unset($this->freqItmsts[implode($this->delimiter, $subsets[$i])]);
                  }
                else
                  {
                    break;
                  }
              }
          }
      }

    $this->fiTime = $this->stopTimer($this->fiTime);
}

public function process($db)
{
    $checked = $result = array();

    $this->freqItemsets($db);
    $this->arTime = $this->startTimer();

    foreach($this->freqItmsts as $k => $v)
      {
        $arr     = explode($this->delimiter, $k);
        $subsets = $this->subsets($arr);
        $num     = count($subsets);
        for($i=0; $i<$num; $i++)
          {
            for($j=0; $j<$num; $j++)
              {
                if($this->checkRule($subsets[$i], $subsets[$j]))
                  {
                    $n1 = $this->realName($subsets[$i]);
                    $n2 = $this->realName($subsets[$j]);

                    $scan = $this->scan($this->combine($subsets[$i], $subsets[$j]));
```

```
                    $c1   = $this->confidence($this->scan($subsets[$i]), $scan);
                    $c2   = $this->confidence($this->scan($subsets[$j]), $scan);

                    if($c1>=$this->minConf)
                      {
                        $result[$n1][$n2] = $c1;
                      }

                    if($c2>=$this->minConf)
                      {
                        $result[$n2][$n1] = $c2;
                      }

                    $checked[$n1.$this->delimiter.$n2] = 1;
                    $checked[$n2.$this->delimiter.$n1] = 1;
                  }
                }
              }
            }

      $this->arTime = $this->stopTimer($this->arTime);

     return $this->rules = $result;
    }

    public function printFreqItemsets()
    {
      echo                  'Time:                  '.$this->fiTime.'                  second(s)<br
/>==========================================================================
======================<br />';

      foreach($this->freqItmsts as $k => $v)
        {
          $tmp  = '';
          $tmp1 = '';
          $k    = explode($this->delimiter, $k);
          $num  = count($k);
          for($i=0; $i<$num; $i++)
            {
              if($i)
                {
                  $tmp  .= $this->delimiter.$this->realName($k[$i]);
                  $tmp1 .= $this->delimiter.$k[$i];
                }
              else
```

```
                {
                  $tmp  = $this->realName($k[$i]);
                  $tmp1 = $k[$i];
                }
            }

        echo '{'.$tmp.'} = '.$this->allsups[$tmp1].'<br />';
      }
}

public function saveFreqItemsets($filename)
{
  $content = '';

  foreach($this->freqItmsts as $k => $v)
    {
      $tmp  = '';
      $tmp1 = '';
      $k    = explode($this->delimiter, $k);
      $num  = count($k);
      for($i=0; $i<$num; $i++)
        {
          if($i)
            {
              $tmp  .= $this->delimiter.$this->realName($k[$i]);
              $tmp1 .= $this->delimiter.$k[$i];
            }
          else
            {
              $tmp  = $this->realName($k[$i]);
              $tmp1 = $k[$i];
            }
        }

      $content .= '{'.$tmp.'} = '.$this->allsups[$tmp1]."\n";
    }

   file_put_contents($filename, $content);
}

public function getFreqItemsets()
{
  $result = array();

  foreach($this->freqItmsts as $k => $v)
```

```
        {
          $tmp        = array();
          $tmp['sup'] = $this->allsups[$k];
          $k          = explode($this->delimiter, $k);
          $num        = count($k);
          for($i=0; $i<$num; $i++)
            {
              $tmp[] = $this->realName($k[$i]);
            }

          $result[] = $tmp;
        }

     return $result;
    }

    public function printAssociationRules()
    {
      echo                    'Time:                    '.$this->arTime.'                    second(s)<br
/>==========================================================================
====================<br />';

      foreach($this->rules as $a => $arr)
        {
          foreach($arr as $b => $conf)
            {
              echo "$a => $b = $conf%<br />";
            }
        }
    }
    public function saveAssociationRules($filename)
    {
       $content = '';

      foreach($this->rules as $a => $arr)
        {
          foreach($arr as $b => $conf)
            {
              $content .= "$a => $b = $conf%\n";
            }
        }

       file_put_contents ($filename, $content);
    }
```

```
    public function getAssociationRules()
    {
       return $this->rules;
    }

    private function startTimer()
    {
      list($usec, $sec) = explode(" ", microtime());
      return ((float)$usec + (float)$sec);
    }

    private function stopTimer($start, $round=2)
    {
      $endtime = $this->startTimer()-$start;
      $round   = pow(10, $round);
      return round($endtime*$round)/$round;
    }
}
?>
```

Naïve Bayes

```
<?php

    require_once('Category.php');


    class NaiveBayesClassifier {

        public function __construct() {
        }

        public function classify($sentence) {

                // extracting keywords from input text/sentence
                $keywordsArray = $this -> tokenize($sentence);

                // classifying the category
                $category = $this -> decide($keywordsArray);

                return $category;
        }
```

```
        public function train($sentence, $category) {
                $murder = Category::$Murder;
          $pp = Category::$PP;
          $vt = Category::$VT;
          $rr = Category::$RR;
          $cc = Category::$CC;
          $et = Category::$ET;

                if ($category == $murder || $category == $pp || $category == $vt || $category == $rr || $category == $cc || $category == $et) {

                  //connecting to database
                    require 'db_connect.php';

                    // inserting sentence into trainingSet with given category
                    $sql = mysqli_query($conn, "INSERT into trainingSet (document, category) values('$sentence', '$category')");

                    // extracting keywords
                    $keywordsArray = $this -> tokenize($sentence);

                    // updating wordFrequency table
                    foreach ($keywordsArray as $word) {

                        // if this word is already present with given category then update count else insert
                        $sql = mysqli_query($conn, "SELECT count(*) as total FROM wordFrequency WHERE word = '$word' and category= '$category' ");
                        $count = mysqli_fetch_assoc($sql);

                        if ($count['total'] == 0) {
                                $sql = mysqli_query($conn, "INSERT into wordFrequency (word, category, count) values('$word', '$category', 1)");
                        } else {
                                $sql = mysqli_query($conn, "UPDATE wordFrequency set count = count + 1 where word = '$word'");
                        }
                    }

                    //closing connection
                    $conn -> close();

                } else {
                        throw new Exception('Unknown category. Valid categories are: $murder, $pp, $vt, $cc, $et, $rr');
```

```
        }
    }

    /**
     * this function takes a paragraph of text as input and returns an array of keywords.
     */

    private function tokenize($sentence) {
        $stopWords = array('about','and','are','com','for','from','how',
          'that','the','this',
'was','what','when','where','who','will','with','und','the','www');

          //removing all the characters which ar not letters, numbers or space
          $sentence = preg_replace("/[^a-zA-Z 0-9]+/", "", $sentence);

          //converting to lowercase
          $sentence = strtolower($sentence);

        //an empty array
          $keywordsArray = array();

          //splitting text into array of keywords
        //http://www.w3schools.com/php/func_string_strtok.asp
          $token =  strtok($sentence, " ");
          while ($token !== false) {

                //excluding elements of length less than 3
                if (!(strlen($token) <= 2)) {

                      //excluding elements which are present in stopWords array
                  //http://www.w3schools.com/php/func_array_in_array.asp
                      if (!(in_array($token, $stopWords))) {
                            array_push($keywordsArray, $token);
                      }
                }
                $token = strtok(" ");
          }
          return $keywordsArray;
    }


    private function decide ($keywordsArray) {
        $murder = Category::$Murder;
      $pp = Category::$PP;
```

```
$vt = Category::$VT;
$rr = Category::$RR;
$cc = Category::$CC;
$et = Category::$ET;

// by default assuming category to be ham
    $category = $pp;

    // making connection to database
  require 'db_connect.php';

    $sql = mysqli_query($conn, "SELECT count(*) as total FROM trainingSet WHERE  category = '$pp' ");
    $spamCount = mysqli_fetch_assoc($sql);
    $spamCount = $spamCount['total'];

    $sql = mysqli_query($conn, "SELECT count(*) as total FROM trainingSet WHERE  category = '$murder' ");
    $hamCount = mysqli_fetch_assoc($sql);
    $hamCount = $hamCount['total'];

$sql = mysqli_query($conn, "SELECT count(*) as total FROM trainingSet WHERE  category = '$cc' ");
$ccCount = mysqli_fetch_assoc($sql);
$ccCount = $ccCount['total'];

$sql = mysqli_query($conn, "SELECT count(*) as total FROM trainingSet WHERE  category = '$rr' ");
$rrCount = mysqli_fetch_assoc($sql);
$rrCount = $rrCount['total'];

$sql = mysqli_query($conn, "SELECT count(*) as total FROM trainingSet WHERE  category = '$et' ");
$etCount = mysqli_fetch_assoc($sql);
$etCount = $etCount['total'];

$sql = mysqli_query($conn, "SELECT count(*) as total FROM trainingSet WHERE  category = '$vt' ");
$vtCount = mysqli_fetch_assoc($sql);
$vtCount = $vtCount['total'];

    $sql = mysqli_query($conn, "SELECT count(*) as total FROM trainingSet ");
    $totalCount = mysqli_fetch_assoc($sql);
    $totalCount = $totalCount['total'];
```

```
		//p(spam) pick pocketing
		$pSpam = $spamCount / $totalCount; // (no of documents classified as spam / total no of documents)

		//p(ham) Murder
		$pHam = $hamCount / $totalCount; // (no of documents classified as ham / total no of documents)

	$pcc = $ccCount / $totalCount;

	$prr = $rrCount / $totalCount;

	$pet = $etCount / $totalCount;

	$pvt = $vtCount / $totalCount;
		//echo $pSpam." ".$pHam;

	// no of distinct words (used for laplace smoothing)
		$sql = mysqli_query($conn, "SELECT count(*) as total FROM wordFrequency ");
		$distinctWords = mysqli_fetch_assoc($sql);
		$distinctWords = $distinctWords['total'];

		$bodyTextIsSpam = log($pSpam);
		foreach ($keywordsArray as $word) {
			$sql = mysqli_query($conn, "SELECT count as total FROM wordFrequency where word = '$word' and category = '$pp' ");
			$wordCount = mysqli_fetch_assoc($sql);
			$wordCount = $wordCount['total'];
			$bodyTextIsSpam += log(($wordCount + 1) / ($spamCount + $distinctWords));
		}

		$bodyTextIsHam = log($pHam);
		foreach ($keywordsArray as $word) {
			$sql = mysqli_query($conn, "SELECT count as total FROM wordFrequency where word = '$word' and category = '$murder' ");
			$wordCount = mysqli_fetch_assoc($sql);
			$wordCount = $wordCount['total'];
			$bodyTextIsHam += log(($wordCount + 1) / ($hamCount + $distinctWords));
		}

	$bodyTextIscc = log($pcc);
```

```
        foreach ($keywordsArray as $word) {
            $sql = mysqli_query($conn, "SELECT count as total FROM wordFrequency
where word = '$word' and category = '$cc' ");
            $wordCount = mysqli_fetch_assoc($sql);
            $wordCount = $wordCount['total'];
            $bodyTextIscc += log(($wordCount + 1) / ($ccCount + $distinctWords));
        }

        $bodyTextIsrr = log($prr);
        foreach ($keywordsArray as $word) {
            $sql = mysqli_query($conn, "SELECT count as total FROM wordFrequency
where word = '$word' and category = '$rr' ");
            $wordCount = mysqli_fetch_assoc($sql);
            $wordCount = $wordCount['total'];
            $bodyTextIsrr += log(($wordCount + 1) / ($rrCount + $distinctWords));
        }

        $bodyTextIset = log($pet);
        foreach ($keywordsArray as $word) {
            $sql = mysqli_query($conn, "SELECT count as total FROM wordFrequency
where word = '$word' and category = '$et' ");
            $wordCount = mysqli_fetch_assoc($sql);
            $wordCount = $wordCount['total'];
            $bodyTextIset += log(($wordCount + 1) / ($etCount + $distinctWords));
        }

        $bodyTextIsvt = log($pvt);
        foreach ($keywordsArray as $word) {
            $sql = mysqli_query($conn, "SELECT count as total FROM wordFrequency
where word = '$word' and category = '$vt' ");
            $wordCount = mysqli_fetch_assoc($sql);
            $wordCount = $wordCount['total'];
            $bodyTextIsvt += log(($wordCount + 1) / ($vtCount + $distinctWords));
        }

            if ($bodyTextIsHam >= $bodyTextIsSpam && $bodyTextIsHam >=
$bodyTextIscc && $bodyTextIsHam >= $bodyTextIsrr && $bodyTextIsHam >=
$bodyTextIsvt && $bodyTextIsHam >= $bodyTextIset) {
                $category = $murder;
            } else if ($bodyTextIsSpam >= $bodyTextIsHam &&
$bodyTextIsSpam >= $bodyTextIscc && $bodyTextIsSpam >= $bodyTextIsrr &&
$bodyTextIsSpam >= $bodyTextIsvt && $bodyTextIsSpam >= $bodyTextIset){
                $category = $pp;
            }
```

```
        else if ($bodyTextIscc >= $bodyTextIsHam && $bodyTextIscc >= $bodyTextIsSpam && $bodyTextIscc >= $bodyTextIsrr && $bodyTextIscc >= $bodyTextIsvt && $bodyTextIscc >= $bodyTextIset){
            $category = $cc;
        }
        else if ($bodyTextIsrr >= $bodyTextIsHam && $bodyTextIsrr >= $bodyTextIsSpam && $bodyTextIsrr >= $bodyTextIscc && $bodyTextIsrr >= $bodyTextIsvt && $bodyTextIsrr >= $bodyTextIset){
            $category = $rr;
        }
        else if ($bodyTextIset >= $bodyTextIsHam && $bodyTextIset >= $bodyTextIsSpam && $bodyTextIset >= $bodyTextIscc && $bodyTextIset >= $bodyTextIsvt && $bodyTextIset >= $bodyTextIsrr){
            $category = $et;
        }
        else if ($bodyTextIsvt >= $bodyTextIsHam && $bodyTextIsvt >= $bodyTextIsSpam && $bodyTextIsvt >= $bodyTextIscc && $bodyTextIsvt >= $bodyTextIset && $bodyTextIsvt >= $bodyTextIsrr){
            $category = $vt;
        }

            $conn -> close();

            return $category;
    }
}

?>
```