**BIKE THEFT DETECTION SYSTEM**

**A PROJECT REPORT**

***Submitted by***

|  |  |
| --- | --- |
| **DIWAKAR N** | **(Reg. No. 201304029)** |
| **KARTHIKEYAN R R** | **(Reg. No. 201304055)** |
| **MADHU PERKIN T** | **(Reg. No. 201304059)** |

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

**in**

COMPUTER SCIENCE AND ENGINEERING

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**MEPCO SCHLENK ENGINEERING COLLEGE, SIVAKASI**

**(An Autonomous Institution affiliated to Anna University Chennai)**

**April 2017**

**BONAFIDE CERTIFICATE**

Certified that this project report titled **BIKE THEFT DETECTION SYSTEM** is the bonafide work of **Mr. DIWAKAR.N, Mr. KARTHIKEYAN.R.R, Mr. MADHU PERKIN .T** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Mrs. M. Nagajothi, M.E., Dr. K.Muneeswaran, M.E., Ph.D.,

**Internal guide Head of the Department**

**Assistant professor (Sr. Grade), Senior Professor,**

**Department of Computer Science Department of Computer Science and Engineering, and Engineering,**

**Mepco Schlenk Engineering College, Mepco Schlenk Engineering College,**

**Sivakasi. Sivakasi.**

Submitted for viva-voce Examination held at **MEPCO SCHLENK ENGINEERING COLLEGE, SIVAKASI (AUTONOMOUS)** on **……………..............**

**Internal Examiner External Examiner**

|  |  |  |  |
| --- | --- | --- | --- |
| **ACKNOWLEDGEMENT**  **ACKNOWLEDGEMENT**    First and foremost, we thank the Almighty for his abundant blessings that is showered upon our past, present, future successful endeavors.  At the appellation, we would like to extend our sincere gratitude to our reverent principal, **Dr. S. Arivazhagan, M.E., Ph.D.,** for providing sufficient working environment such as system and library facilities. We also thank him very much for providing us with adequate lab facilities, which enable us to complete our project.  We are grateful to our charismatic Head of the Computer Science Engineering and Department **Dr. K. Muneeswaran, M.E., Ph.D.,** for giving us this golden opportunity to undertake a project of this nature and for his most valuable guidance given at every phase of our work.  We would also like to extend our gratitude **Mr. C. Bala Subramanian, M.Tech., (Ph.D)** Assistant professor (Sr. Grade), Department of Computer Science and Engineering, Mepco Schlenk Engineering College for being our project coordinator and directing us through our project.  We thank **Mrs. M. Nagajothi, M.E.,** Assistant professor, Department of Computer Science and Engineering, Mepco Schlenk Engineering College for his valuable guidance and encouragement.  We acknowledge with the sense of gratitude, the help given to us by all the staffs and technicians in the department.  We are very grateful to our beloved parents and friends who afforded the necessary help for us at the right time for making our project a grand success.  **TABLE OF CONTENTS**  **TABLE OF CONTENTS** | | | |
| **CONTENTS** | |  | **PAGE NO** |
|  | |  |  |
| **ABSTRACT** | |  | Iii |
| **LIST OF FIGURES** | |  | Vii |
| **LIST OF TABLES** | |  | Viii |
| **LIST OF ABBREVIATIONS** | |  | Ix |
| **CHAPTER 1** | | **INTRODUCTION** |  |
|  | |  |  |
|  | 1.1 | GENERAL | 1 |
|  | 1.2 | OBJECTIVE OF THE PROJECT | 4 |
|  | 1.3 | PROBLEM STATEMENT | 4 |
|  | 1.4 | PROPOSED SYSTEM | 5 |
|  | 1.5 | APPLICABLE AREAS | 5 |
| **CHAPTER 2** |  | **LITERATURE SURVEY** |  |
|  | 2.1 | AFFINITY ANALYSIS AND ASSOCIATION RULE MINING USING APRIORI ALGORITHM IN MARKET BASKET ANALYSIS | 6 |
|  | 2.2 | DATA MINING BASED STORE LAYOUT ARCHITECTURE FOR SUPERMARKET | 17 |
|  | 2.3 | INFORMATION SYSTEM ON MARKET BASKET ANALYSIS | 20 |
|  | 2.4 | ALIGNMENT AND INTEGRATION OF COMPLEX NETWORKS BY HYPERGRAPH- BASED SPECTRAL CLUSTERING | 22 |
| **CHAPTER 3** | | **SYSTEM DESIGN** |  |
|  | 3.1 | DATA FLOW DESIGN | 37 |
|  |  | 3.3.1 Transaction REPOSITORY |  |
|  |  | 3.3.2 fILTER SINGLE ITEM ORDER |  |
|  |  | 3.3.3 FREQUENT ITEM SET MINING |  |
|  |  | 3.3.4 HYPER GRAPH CONSTRUCTION |  |
|  |  | 3.3.5 HYPER GRAPH CLUSTERING |  |
| **CHAPTER 4** | | **SYSTEM IMPLEMENTATION** |  |
|  |  |  |  |
|  | 4.1 | FILTER SINGLE ITEM ORDER IMPLEMENTATION |  |
|  | 4.2 | FREQUENT ITEM SET MINING IMPLEMENTATION |  |
|  | 4.3 | HYPER GRAPH CONSTRUCTION |  |
|  | 4.4 | HYPER GRAPH CLUSTERING IMPLEMENTATION |  |
| **CHAPTER 5** | | **RESULTS AND DISCUSSION** |  |
|  |  |  |  |
|  | 5.1 | TEST CASES AND RESULTS |  |
|  |  | 5.1.1 INPUT DATA |  |
|  |  | 5.1.2 FILTER SINGLE ITEM ORDER TESTING AND RESULTS |  |
|  |  | 5.1.3 FREQUENT ITEM SET MINING TESTING AND RESULTs |  |
|  |  | 5.1.4 HYPER GRAPH CONSTRUCTION TESTING AND RESULTS |  |
|  |  | 5.1.5 HYPER GRAPH CLUSTERING TESTING AND RESULTS |  |
|  |  | 5.1.6 NAMING THE PRODUCT ID |  |
|  |  | 5.1.7 representation of clusters |  |
|  |  | 5.1.8 Positive predictive value (ppv) |  |
| **CHAPTER 6** | | **conclusion and future enhancement** |  |
|  |  |  |  |
| **APPENDIX A**  **APPENDIX B**  **APPENDIX C** |  |  | 55 |
|  |  |  |  |

**ABSTRACT**

**ABSTRACT**

In today’s era everyone is using motor bikes for transportation. Bike theft started increasing now a days. In an attempt to prevent this, we propose a GPS and GSM based bike tracking and controlling system to help the users locate their bikes using GPS. The proposed technology allows people to get alert and can track real-time information about their bike when stolen. The system is equipped with GPS and GSM modules along with Arduino microcontroller and relay that is installed in bikes. The integration of the system with the bike is also simple without any major modifications to the bike design. During bike theft or when a bike is started without key, user receives an alert message from the system installed on bike and the bike is controlled based on the reply message received from the owner. This location information will be sent to registered mobile number. The request to the tracking system can be made through SMS from a registered mobile number or through website after authentication and the bike can be stopped based on the user’s command. It is completely integrated in bikes making it easy to track them any time. The user is also provided service from the nearest service centre through our website automatically.

**LIST OF FIGURES**

**LIST OF FIGURES**

**S. No. FIGURE No. FIGURE NAME PAGE No.**

1. 3.1 Data Flow Design of Market Basket Analysis 6

2. 3.2 Construction of FP Tree from the transaction 13

3. 3.3 Representation of Hyper Graph 14

4. 5.1 Transaction of data 15

5. 5.2 Product ID and its corresponding name 17

6. 5.3 Frequent pattern tree 35

7. 5.4 FP Growth tree 36

8. 5.5 Hyper Graph edges 37

9. 5.6 Hyper Graph clustering 38

10. 5.7 Naming with product id 39

11. 5.8 Representation of clusters 40

12. 5.9 PPV vs number of items 41

13. 5.10 PPV vs number of transactions 42

**LIST OF TABLES**

**LIST OF TABLES**

**S. No. TABLE No. TABLE NAME PAGE No.**

1. 5.1 Test Case for Filter Single Item Order 12
2. 5.2 Test Case for Frequent Item Set Mining 22
3. 5.3 Test Case for Hyper Graph Construction 24
4. 5.4 Test Case for Hyper Graph Clustering 25
5. 5.5 Test Case for Name the Product ID 26
6. 5.6 Representation of Clusters 29

**LIST OF ABBREVIATIONS**

**ABBREVIATION**

MBA : Market Basket Analysis

FISM: Frequent Item Set Mining

FIS : Frequent Item Set

FP : Frequent Pattern

PPV : Positive Predictive Value

KDD : Knowledge Discovery in Database

**INTRODUCTION**

**CHAPTER 1**

**INTRODUCTION**

* 1. **GENERAL**

During the last few years there has been a near exponential increase in the information generated. A study by EMC2 shows that the annual amount of data generated and copied would reach a stunning 44 trillion gigabytes by the year 2020. This rise is due to the increase in storage capability, network speed, embedded systems, sensors and smart devices. The data that can be used for analysis is also expected to grow from 22% in the year 2013 to more than 35% in 2020.

The abundant data available today can be efficiently tapped to obtain knowledge which can be used in many areas such as marketing, heath care, sports, research and security.

Data Mining is the process of discovering interesting patterns in large datasets. Data mining involves techniques from artificial intelligence, machine learning, statistics and database systems. Some data mining techniques like decision trees and rules, non-linear regression and classification methods and example based methods.

Market Basket Analysis (MBA) is a very prominent technique of data mining which is widely used in identifying products. Now a day, there is a boom in online shopping of products due to rapid increase in e-commerce websites, and the ease in using them. This has made an enormous increase in transactional datasets. The availability of such datasets has promoted the researches to analyze the data and use it for productivity of the retailers. The increasing number of e-commerce websites has led to development of competition between the retailers, thus this analysis of consumer purchase behavior has become a point of prime importance for them. This kind of analysis has helped them to gain the competitive advantage. To adapt to the needs of consumers retailers need to know the demands and expectations, which can be very well known by performing affinity analysis. This also helps us to know who the consumers are, understand why they make certain purchases, and gain insight about its merchandise.

* 1. **OBJECTIVE OF THE PROJECT**
* To identify group of products that are likely to be bought together so that they can be placed in the same warehouse allowing multi-item orders to be shipped together.
* To analyze the true positive results for the product items and the transactions.

* 1. **PROBLEM STATEMENT**

Identification of group of products bought together by customers from the online retailers with better accuracy is a tedious process. Placing the identified product in the same warehouse to allow multi-item orders to be shipped together.

**1.4 PROPOSED SYSTEM**

The proposed design has 4 phases which are Filtering Single Item Order, Frequent Item set Mining, Hyper Graph Construction and Hyper Graph Clustering. The frequent items that are bought together are identified using Frequent Item Set Mining (FISM) algorithms such as FP-Growth algorithm. This relationship is then mapped to a hyper graph. The actual product clusters are then identified by employing hyper graph clustering algorithms. These clusters contains the products which are to that are likely to be bought together so that they can be placed in the same warehouse allowing multi-item orders to be shipped together.

* 1. **APPLICABLE AREAS**

Online retailers such as

* Flipkart
* Amazon
* Voonik

**1.6 ORGANIZATION OF REPORT:**

The overviews of the subjects described in the underlying chapters are given below

* **Chapter 2:** Covers discussion about the related works carried out before formulating the proposed system. It also briefs about the methods adopted by existing system and its disadvantages.
* **Chapter 3:** Deals with the overview design of the proposed system and the formulations used in the implementation of the system.
* **Chapter 4:** Outlines the design and explains the methodology of the proposed system.
* **Chapter 5:** Deals with the implementation of the proposed system, how the modules interact and how the method executes to come with the required result.
* **Chapter 6:** Explains about the experiments carried out in proving the efficiency of the system.
* **Chapter 7:** Concludes about the system and discusses about any future enhancements.

**LITERATURE SURVEY**

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 AFFINITY ANALYSIS AND ASSOCIATION RULE MINING USING APRIORI ALGORITHM IN MARKET BASKET ANALYSIS**

R. Karthiyayini and Dr. R. Balasubramanian focused on Affinity analysis and association rule mining encompasses a broad set of analytics techniques aimed at revealing the associations and correlation between specific objects. Association Rule Mining is a powerful tool in Data Mining. In large databases, it is used to identifying correlation or pattern between objects. Market basket analysis is one of the ways to derive associations by examining the buying habits of the customers in their baskets. Market Basket Analysis is a mathematical modeling technique based upon the theory that if you buy a certain group of items, you are likely to buy another group of items. It is used to analyze the customer purchasing behavior and helps in increasing the sales and maintain inventory by focusing on the point of sale transaction data.

For example, In a retail shop 400 customers had visited in last month to buy products. It was observed that out of 400 customers, 200 of them bought Product A, 160 of them bought Product B and 100 of them buy both Product A and Product B; we can say 50%(200 out of 400) of the customer buy Product A, 40%(160 out of 400) customers buy Product B and 25% (100 out of 400) buy Product A and B. some terminologies to be discussed,

1) Items (Products)

Items are the objects that we are identifying associations between. For an online retailer, each item is a product in the shop. A group of items is an item sets( set of products.)

2) Support

The support of a product or set of products is the fraction of transactions in our data set that contain that product or set of products. In our example,

1. Support(Product A)=50%

2. Support(Product B)=40%

3. Support(Product A and B)=25%

3) Confidence

Confidence is a conditional probability that customer buy product A will also buy product B. Out of 200 customers who bought Product A, 100 bought Product B too. Confidence (Product A, Product B) = 100/200=50% It implies if someone buys product A, they are 50% likely to buy Product B too. Confidence (A==>B)=Support( A and B)/Support (A)

4) Lift:

If someone buys product A, what % of chance of buying product B would increase. A lift greater than 1 indicates that the presence of A has increased the probability that the product B will occur on this transaction. A lift smaller than 1 indicates that the presence of A has decreased the probability that the product B will occur on this transaction. Lift (A==>B)= Confidence(A==>B)/Support(B) % increase of chance of buying other product(s)=(Lift-1)\*100 A lift value of 1.25 implies that chance of buying product B(on the right hand side) would increase by 25%.

5) Desired outcome:

In market basket analysis, we pick rules with a lift of more than one because the presence of one product increases the probability of the other product(s) on the same transaction. Rules with higher confidence are ones where the probability of an item appearing on the RHS is high given the presence of the items on the LHS. The formulas to calculate them are:

Support(A=>B) = P(AUB)-------------------------(1)

Confidence(A=>B) = P(B|A)---------------------------(2)

= P(AUB)/P(A)------------------(3)

Lift(A=>B) = Confidence(A=>B)/P(B)-----(4)

= P(AUB)/P(A)P(B) ------------(5)

Where P(A) is the percentage( or Probability) of cases containing A. In this proposed system, we analyzed the transactional dataset with summarized information on buying status according to item details of ID, Products, Quality, Status. To make it suitable for association rule mining, we reconstruct the raw data as new.raw, where each row represents a product.

**LIMITATIONS**

The Apriori algorithm used for the market basket analysis requires many database scans which in turn consumes more time for generating frequent item sets. The Apriori algorithm assumes that the transaction database resides in the memory which occupies more memory space while computing more data.

**2.2 DATA MINING BASED STORE LAYOUT ARCHITECTURE FOR SUPERMARKET**

Aishwarya Madan Mirajkar, Aishwarya Prafulla Sankpal, Priyanka Shashikant Koli , Rupali Anandrao Patil and Ajit Ratnakar Pradnyavant designed a system to find the most frequent combinations of items. It is based on developing an efficient algorithm that outperforms the best available Frequent Pattern algorithms on a number of typical data sets. The system contains the following models :

1) Data Input.

2) Market Basket Analysis.

3) Applying Association Rule.

4) Graphical representation.

**Data input**

In data input model we are using database of the supermarket .In that we are storing information about categories of the products, price of the products, quantity of products and names of the products .

**Market basket understanding**

Market Basket Analysis is used to determine which products sell together, the input data to a Market Basket Analysis is normally a list of sales transactions, where each has two dimensions, one represents a product and the other represents a customer, depending on whether the goal of the analysis is to find which items sell together to the same person. Apriori algorithm is one of the most widely used and famous techniques for finding association rules. This algorithm was chosen primarily due to the speed of application. Simulation models, online analytical processing and data visualization. On the other hand, a data warehouse is the newest form of decision support system.

**Applying Association Rule**

The applications of data mining in retail trade enterprises are mainly concentrated in association rules mining. Association rule mining is an initial data exploration approach that is often applied to extremely large data set. An example is grocery store market basket data. Association rules mining provides valuable information in assessing significant correlations. By mining association rules, marketing analysts try to find sets of products that are frequently bought together. They have been applied to a variety of fields. Market Basket Analysis is used to determine which products sell together, the input data to a Market Basket analysis .We use the buying association measure to create a category correlation matrix and we apply the multi-dimensional scale technique to display the set of products in the store space.

**Graphical representation**

Data modeling is where the data mining software is used to generate results for various situations. The data processing in Clementine is done through the use of nodes, which are then connected together to form a stream frame. In addition, data visualization can be presented to users after the mining process has been done in the graphical format.

From these models the frequent data sets are generated using the Apriori algorithm and Brute Force approach which uses bottom up approach.

**LIMITATIONS**

A brute-force algorithm is not necessarily a one that requires a lot of memory but it generally requires a lot of processing time.  For many problems, brute force is only feasible for small problem dimensions. Similarly Apriori algorithm have their own disadvantages.

**2.3 INFORMATION SYSTEM ON MARKET BASKET ANALYSIS**

The authors Shruti Kawale, Rajni Tetwar, Nirmala Suryavanshi, Prachi Wankhede and

Prof. Neha Titarmare used Apriori algorithm to identify the frequent patterns present in the transactions repository. The most basic principles for apriori to follow are:

1) All subset of a frequent itemset must be frequent.

2) No super set of any infrequent itemset should be tested or generated.

3) If an itemset is infrequent, then all of its super sets must also be infrequent.

**Steps of apriori algorithm:**

**1) Generating frequent item list**

Here first we have to generate a candidate set before generation of frequent item list. The database is scanned and support of items is counted from the transaction this is called the candidate set generation. After generation of candidate set we have to make use of it to obtain frequent itemset list. So for this we calculate a k-itemset candidate list and then a k-itemset frequent list, this list contains all the items which are above or equal to the minimum support count and should be present all last frequent itemset [3].Now this frequent k-itemset is used to generate a k+1-itemset candidate list. And then we again use this k+1- candidate list to generate k+1 frequent list. This is the nature of apriori algorithm to generate frequent items from the data set. This goes on till we get an empty set of candidate or frequent items. The most important point here is that the final set as well the set of frequent list which were gathered from the candidate set previously should have support count greater than or equal to minimum support threshold value.

**2) Associating frequent items**

This step completely deals with the set of items which are frequent, and this step has nothing to deal with the candidate sets used previously as they do not necessarily have items above support count. Here we use the final frequent list to make non-empty subset from individual items. The following example explains this, F= {i2,i4,i7} be the set of frequent item set, therefore the non-empty subsets are {i2,i4}, {i2,i7}, {i4,i7}, {i2}, {i4} , {i7}. The next step is to associate all of them together for example relation between {i2,i4} → {i2} “in press” [5]. Using these non-empty subsets we associate them together one by one and then calculate the confidence for them. If this calculated value comes above or equal to the minimum confidence value then the pair of items is considered to be profitable if used together for marketing purpose. So this is how apriori algorithm helps us to know the association between the items or more specifically between the frequent items. But if the calculated confidence is less than the threshold value then the pair of items is neglected. This value of confidence is calculated for every association possible with the subsets irrespective whether the confidence will be above or below threshold. This is how we get the sets of beneficial pairs of items which were purchased together by the consumers. Thus this is a step by step process to be followed to get the final results regarding the motive of market basket analysis.

**LIMITATION**

Candidate generation is extremely slow in apriori algorithm. Runtime of apriori algorithm increases exponentially depending on the number of different items.

**2.4 ALIGNMENT AND INTEGRATION OF COMPLEX NETWORKS BY HYPERGRAPH- BASED SPECTRAL CLUSTERING**

Tom Michoel and Bruno Nachtergaele proposed that the complex networks possess a rich, multi-scale structure reflecting the dynamical and functional organization of the systems they model. Often there is a need to analyze multiple networks simultaneously, to model a system by more than one type of interaction or to go beyond simple pairwise interactions, but currently there is a lack of theoretical and computational methods to address these problems. Here we introduce a framework for clustering and community detection in such systems using hypergraph representations. Our main result is a generalization of the Perron-Frobenius theorem from which we derive spectral clustering algorithms for directed and undirected hypergraphs. We illustrate our approach with applications for local and global alignment of protein-protein interaction networks between multiple species, for tripartite community detection in folksonomies, and for detecting clusters of overlapping regulatory pathways in directed networks.

**Graphs and Hyper Graph**

A graph G is defined as a pair (V, E) of vertices V and edges (pairs of vertices) E, which may be directed or not. In a weighted graph, a number is assigned to each edge which may represent, e.g., the cost, length or reliability of an edge. A hyper graph is a generalization of a graph where an edge, called hyper edge in this case, can connect any number of vertices, i.e., E is a set of arbitrarily sized subsets of V. A particular class of hyper graphs are so called k-uniform hyper graphs where each hyper edge has the same cardinality k. Algebraically, a graph can be represented by an adjacency matrix A of dimension N × N, with N the number of vertices, such that Aij = 1 if {i, j} ∈ E and 0 otherwise. For undirected graphs, A is a symmetric matrix and for weighted graphs, Aij is defined to be the weight of the edge {i, j}. For k uniform hyper graphs, the notion of adjacency matrix can be generalized to an adjacency multi-array or tensor T, with Ti1...ik = 1 if {i1, . . . , ik} ∈ E and 0 otherwise. For a general hyper graph, we define a function w on the set of subsets of V such that w(E) = 1 for E ∈ E and 0 otherwise. In general, we allow weighted hyper graphs where w can be any non-negative function.

**Spectral Clustering and Bi-Clustering in Hyper Graph**

Having a generalization of the Perron-Frobenius theorem, it is straightforward to also generalize the spectral clustering method. Define for X ⊂ V,

with x the dominant eigenvector and uX ∈ SNp now de-fined by uX,i = |X|−1/p for i ∈ X and 0 otherwise. The parameter p balances cluster size versus edge density. For p = 1, Sp is the ratio of edges to nodes in X. Taking p > 1 diminishes the influence of the denominator and progressively favors to have a high number of edges rather than a high number of edges per node in high-scoring clusters (further details in Section VII). The spectral clustering algorithm becomes:

1. Calculate the maximizer x of Rp.
2. Find the cluster Xmax which solves the restricted variational problem Xmax = argmax c>0 Sp(Xc) with Xc = {i ∈ V : xi > c}.
3. Store Xmax, remove all hyperedges between nodes in Xmax from the edge set E, and repeat the procedure until no more hyperedges remain.

**LIMITATION**

Spectral clustering cares about connectivity instead of geometrical proximity. So if the data isn't well geometrically separated, but clusters aren't connected, spectral clustering will work well. In other cases the k-means algorithm works well.

**SYSTEM STUDY**

**CHAPTER 3**

**3 SYSTEM STUDY**

The core idea of this system is to track and control the bike remotely. Initially the system is installed in the bike near engine. Whenever the bike is started without the key, the SMS is sent to the owner mobile through GSM. The location of the bike is determined using GPS and sent to the owner on request through SMS.

First of all, a detailed study has to be made on Arduino microcontrollers, GSM, GPS, DC Motor, 5V Regulator (KA7805) and Ignition Lock.

**3.1 ARDUINO:**

Arduino is an open-source computer hardware and software company, project and user community that designs and manufactures microcontroller-based kits for building digital devices and interactive objects that can sense and control objects in the physical world.

The work is based on microcontroller board designs, manufactured by several vendors, using various microcontrollers. These systems provide sets of digital and analog I/O pins that can be interfaced to various expansion boards ("shields") and other circuits. The boards feature serial communications interfaces, including USB on some models, for loading programs from personal computers. For programming the microcontrollers, the Arduino project provides an integrated development environment (IDE) based on the Processing project, which includes support for the C and C++ programming languages.

The first Arduino was introduced in 2005, aiming to provide an inexpensive and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.

An Arduino board historically consists of an Atmel 8-, 16- or 32-bit AVR microcontroller (although since 2015 other makers' microcontrollers have been used) with complementary components that facilitate programming and incorporation into other circuits. An important aspect of the Arduino is its standard connectors, which lets users connect the CPU board to a variety of interchangeable add-on modules known as shields.

Some shields communicate with the Arduino board directly over various pins, but many shields are individually addressable via an I²C serial bus—so many shields can be stacked and used in parallel. Prior to 2015 Official Arduinos had used the Atmel mega AVR series of chips, specifically the ATmega8, ATmega168, ATmega328, ATmega1280, and ATmega2560 and in 2015 units by other manufacturers were added. A handful of other processors have also been used by Arduino compatible devices.

Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator (or ceramic resonator in some variants), although some designs such as the LilyPad run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions.

An Arduino's microcontroller is also pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory, compared with other devices that typically need an external programmer. This makes using an Arduino more straightforward by allowing the use of an ordinary computer as the programmer.

At a conceptual level, when using the Arduino integrated development environment, all boards are programmed over a serial connection. Its implementation varies with the hardware version. Some serial Arduino boards contain a level shifter circuit to convert between RS-232 logic levels and TTL-level signals.

Current Arduino boards are programmed via Universal Serial Bus (USB), implemented using USB-to-serial adapter chips such as the FTDI FT232. Some boards, such as later-model Uno boards, substitute the FTDI chip with a separate AVR chip containing USB-to-serial firmware, which is reprogrammable via its own ICSP header.

Other variants, such as the Arduino Mini and the unofficial Boarduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods, when used with traditional microcontroller tools instead of the Arduino IDE, standard AVR ISP programming is used.

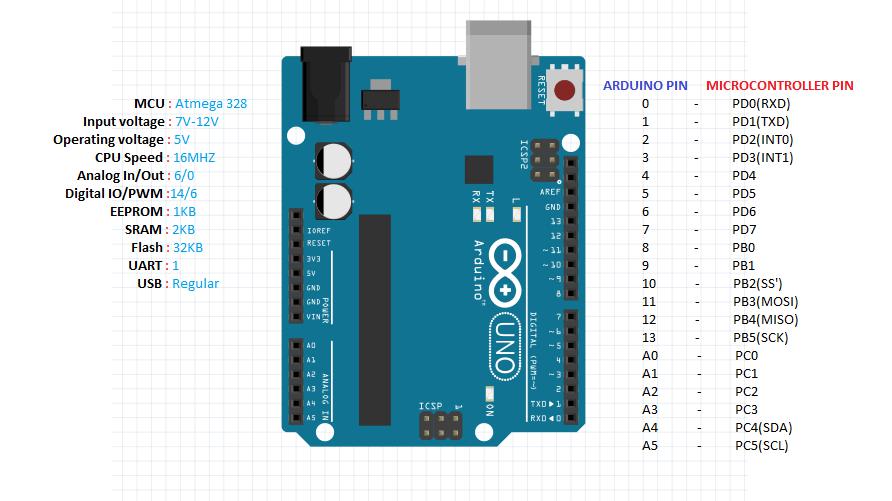
The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The Diecimila, Duemilanove, and current Uno provide 14 digital I/O pins, six of which can produce pulse-width modulated signals, and six analog inputs, which can also be used as six digital I/O pins.

These pins are on the top of the board, via female 0.10-inch (2.5 mm) headers. Several plug-in application shields are also commercially available. The Arduino Nano, and Arduino-compatible Bare Bones Board and Boarduino boards may provide male header pins on the underside of the board that can plug into solder less breadboards.

There are many Arduino-compatible and Arduino-derived boards. Some are functionally equivalent to an Arduino and can be used interchangeably. Many enhance the basic Arduino by adding output drivers, often for use in school-level education to simplify the construction of buggies and small robots.

Others are electrically equivalent but change the form factor, sometimes retaining compatibility with shields, sometimes not. Some variants use completely different processors, with varying levels of compatibility.

**3.1.1 ARDUINO PIN DIAGRAMS:**

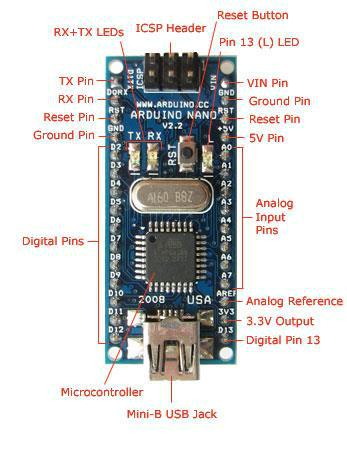
** ARDUINO UNO R3**

**Figure 3.1: Arduino UNO R3 pin diagram**

**Table 3.1 Pin configuration of Arduino Uno**

|  |  |
| --- | --- |
| Pin 0 to Pin 13 | DIGITAL MODE PINS |
| Pin A0 to Pin A5 | ANALOG MODE PINS |
| Pin 0 | Tx Pin |
| Pin 1 | Rx Pin |

**ARDUINO NANO**

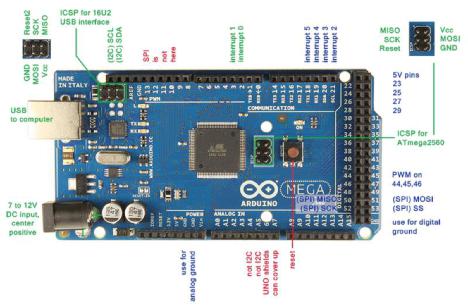
****

**Figure 3.2 Arduino Nano Pin Diagram**

**Table 3.2 Pin configuration of Arduino Nano**

|  |  |
| --- | --- |
| Pin D2 to pin D12 | DIGITAL MODE PINS |
| Pin A0 to Pin A7 | ANALOG MODE PINS |
| GND , RST Pin | GROUND AND RESET PIN |
| AREF | ANALOG REFERENCE |

**ARDUINO MEGA 2560**

****

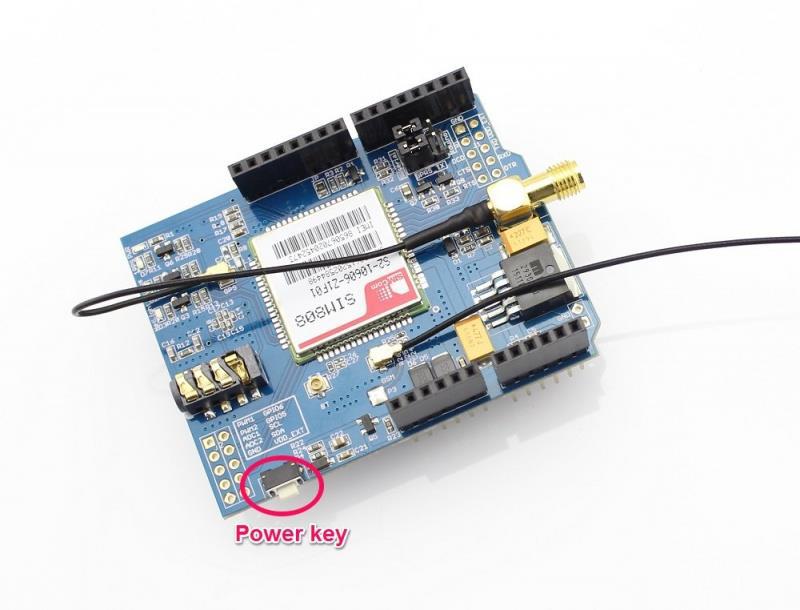
**Figure 3.3 Arduino Mega 2560 Pin diagram**

**Table 3.3 Pin configuration of Arduino Mega 2560**

|  |  |
| --- | --- |
| Pin 0 to Pin 53 | DIGITAL MODE PINS |
| Pin A0 to Pin A15 | ANALOG MODE PINS |
| Pin 18 to Pin 21 | INTERRUPT PINS |
| Pins 23,25,27,29 | 5V PINS |

**3.2 GSM/GPRS MODEM (GLOBAL SYSTEM FOR MOBILE**

**COMMUNICATION/GENRAL PACKET RADIO SERVICE):**

****

**Figure 3.4 GSM/GPRS SIM800A**

GSM/GPRS MODEM is a class of wireless MODEM devices that are designed for communication of a computer with the GSM and GPRS network. It requires a **SIM (Subscriber Identity Module)** card just like mobile phones to activate communication with the network. Also they have **IMEI** (International Mobile Equipment Identity) number similar to mobile phones for their identification. A GSM/GPRS MODEM can perform the following operations:

1. Receive, send or delete SMS messages in a SIM.
2. Read, add, search phonebook entries of the SIM.
3. Make, Receive, or reject a voice call.

The MODEM needs **AT commands**, for interacting with processor or controller, which are communicated through serial communication. These commands are sent by the controller/processor. The MODEM sends back a result after it receives a command. Different AT commands supported by the MODEM can be sent by the processor/controller/computer to interact with the **GSM and GPRS cellular network**.

A **GSM modem** is a specialized type of modem which accepts a SIM card, and operates over a subscription to a mobile operator, just like a mobile phone. From the mobile operator perspective, a GSM modem looks just like a mobile phone.

When a GSM modem is connected to a computer, this allows the computer to use the GSM modem to communicate over the mobile network. While these GSM modems are most frequently used to provide mobile internet connectivity, many of them can also be used for sending and receiving SMS and MMS messages.

A GSM modem can be a dedicated modem device with a serial, USB or Bluetooth connection, or it can be a mobile phone that provides GSM modem capabilities.

For the purpose of this document, the term GSM modem is used as a generic term to refer to any modem that supports one or more of the protocols in the GSM evolutionary family, including the 2.5G technologies GPRS and EDGE, as well as the 3G technologies WCDMA, UMTS, HSDPA and HSUPA.

GSM modems can be a quick and efficient way to get started with SMS, because a special subscription to an SMS service provider is not required. In most parts of the world, GSM modems are a cost effective solution for receiving SMS messages, because the sender is paying for the message delivery.

Due to some compatibility issues that can exist with mobile phones, using a dedicated GSM modem is usually preferable to a GSM mobile phone. This is more of an issue with MMS messaging, where if you wish to be able to receive inbound MMS messages with the gateway, the modem interface on most GSM phones will only allow you to send MMS messages. This is because the mobile phone automatically processes received MMS message notifications without forwarding them via the modem interface.

It should also be noted that not all phones support the modem interface for sending and receiving SMS messages. In particular, most smart phones, including Blackberries, iPhone, and Windows Mobile devices, do not support this GSM modem interface for sending and receiving SMS messages at all at all. Additionally, Nokia phones that use the S60 (Series 60) interface, which is Symbian based, only support sending SMS messages via the modem interface, and do not support receiving SMS via the modem interface.

**3.3 GPS (GLOBAL POSITIONING SYSTEM):**

****

**Figure 3.5 GPS Module**

The **Global Positioning System** (**GPS**) shown in Fig 3.5 is a space-based radio navigation system owned by the United States government and operated by the United States Air Force. It is a [global navigation satellite system](https://en.wikipedia.org/wiki/Satellite_navigation) that provides [geolocation](https://en.wikipedia.org/wiki/Geolocation) and time information to a [GPS receiver](https://en.wikipedia.org/wiki/GPS_receiver) anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites.

A GPS device can retrieve from the GPS system **location and time information** in all weather conditions, anywhere on or near the Earth. A GPS reception requires an unobstructed line of sight to four or more GPS satellites, and is subject to poor satellite signal conditions. In exceptionally poor signal conditions, for example in urban areas, satellite signals may exhibit multipath propagation where signals skip off structures, or are weakened by meteorological conditions. Obstructed lines of sight may arise from a tree canopy or inside a structure, such as in a building, garage or tunnel.

The GPS module continuously produces a set of data regarding the position of the earth surface where it is situated which includes the current position with respect to the equator of the earth in terms of Latitude and Longitude. This data can be decoded and printed into the readable format with the help of a microcontroller only. In this project the data regarding the geographical coordinate is extracted from the **GPS**output with the help of the **Arduino**. The Arduino can be used as a stand-alone board of which the output or inputs can be taken from the boards or given to the board. They can communicate using standard communication ports like USART, TWI, SPI etc. which enables them to be connected with various kinds of devices. The Arduino board is designed for easy prototyping and the IDE used for coding is very simple and provides so many libraries for interfacing with common external devices.

Using the function **Serial.read()** the Arduino continuously reads the data from the GPS module, looking for Latitude-Longitude details. The GPS send the data in standard NMEA format which consist of the real time data regarding the current position. The format includes so many sentences and among them one particular sentence referred to as “Global Positioning System Fix Data” is extracted to read the Latitude Longitude.

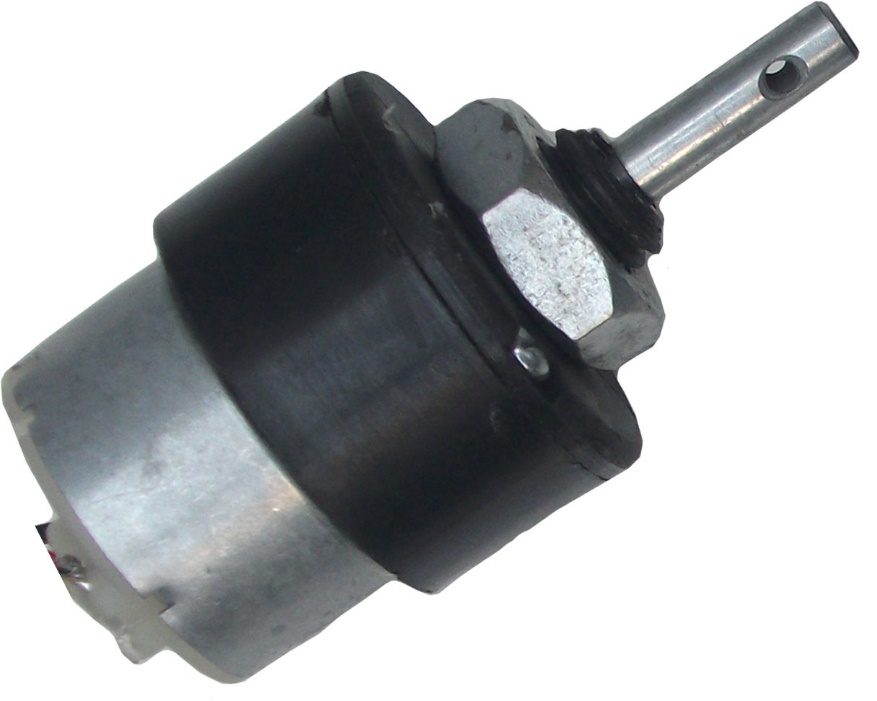
**Example:**

$GPRMC,092204.999,A,4250.5589,S,14718.5084,E,0.00,89.68,211200,,\*25**`**

**Table 3.4 Sample GPS Data and Sentence Format**

|  |  |  |
| --- | --- | --- |
| **Field** | **Example** | **Comments** |
| Sentence ID | $GPRMC |  |
| UTC Time | 092204.999 | hhmmss.sss |
| Status | A | A = Valid, V = Invalid |
| Latitude | 4250.5589 | ddmm.mmmm |
| N/S Indicator | S | N = North, S = South |
| Longitude | 14718.5084 | dddmm.mmmm |
| E/W Indicator | E | E = East, W = West |
| Speed over ground | 0.00 | Knots |
| Course over ground | 0.00 | Degrees |
| UTC Date | 211200 | DDMMYY |
| Magnetic variation |  | Degrees |
| Magnetic variation |  | E = East, W = West |
| Checksum | \*25 |  |

**3.4 12V DC MOTOR:**

****

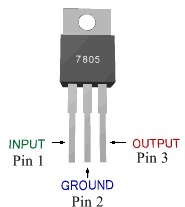
**Fig 3.6 12V DC Motor**

A **DC motor** (Fig 3.6) is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current flow in part of the motor.

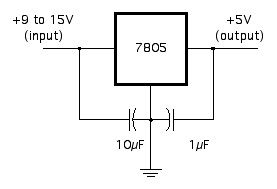
Here, the **DC motor** is used to emulate the motorcycle engine. The **Motorcycle engine** is similar to the DC motor. A motorcycle engine is an engine that powers a motorcycle. Motorcycle engines are typically two-stroke or four-stroke internal combustion engines, but other engine types (such as Wankels and electric motors) have been used in small numbers.

It is a variable speed motor. The speed is low at high Page on torque at light or no load, the motor speed attains dangerously high speed. The motor has a high starting torque.

**3.5 5V VOLTAGE REGULATOR (KA7805):**

** **

**Fig 3.7 Regulator module Fig 3.8 Pin Diagram of 7805**

****

**Fig 3.9 Circuit Connection of 7805**

**7805** shown in Fig 3.7 is a voltage regulator integrated circuit. It is a member of 78xx series of fixed linear voltage regulator ICs. The voltage source in a circuit may have fluctuations and would not give the fixed voltage output. The voltage regulator IC maintains the output voltage at a constant value. The xx in 78xx indicates the fixed output voltage it is designed to provide. 7805 provides +5V regulated power supply. Capacitors of suitable values can be connected at input and output pins depending upon the respective voltage levels.

**Table 3.5 Pin Description of 7805**

|  |  |  |
| --- | --- | --- |
| **Pin No** | **Function** | **Name** |
| 1 | Input voltage (5V-18V) | Input |
| 2 | Ground (0V) | Ground |
| 3 | Regulated output; 5V (4.8V-5.2V) | Output |

A regulated power supply is very much essential for several electronic devices due to the semiconductor material employed in them have a fixed rate of current as well as voltage. The device may get damaged if there is any deviation from the fixed rate. The AC power supply gets converted into constant DC by this circuit. By the help of a voltage regulator DC, unregulated output will be fixed to a constant voltage. The circuit is made up of linear voltage regulator 7805 along with capacitors and resistors with bridge rectifier made up from diodes. From giving an unchanging voltage supply to building confident that output reaches uninterrupted to the appliance, the diodes along with capacitors handle elevated efficient signal conveyal.

**3.6 MOTOR BIKE IGNITION LOCK:**

****

**Fig 3.10 Motor Bike Ignition Lock**

An **ignition lock** or **starter switch** shown in Fig 3.10 is a switch in the control system of an internal combustion engine motor vehicle that activates the main electrical systems for the vehicle. Besides providing power to the starter solenoid and the ignition system components (including the engine control unit and ignition coil) it also usually switches on power to many "accessories" (radio, power windows, etc.). The ignition switch usually requires a key be inserted that works a lock built into the switch mechanism. It is frequently combined with the starter switch which activates the starter motor. The ignition locking system may be bypassed by disconnecting the wiring to the switch and manipulating it directly; this is known as **hotwiring**.

The [ignition system](https://en.wikipedia.org/wiki/Ignition_system) is used to ignite the fuel-air mixture in the engine. The starter system is the ignition system, plus the [battery](https://en.wikipedia.org/wiki/Automotive_battery), and starter switch and DC [starter motor](https://en.wikipedia.org/wiki/Starter_motor). The ignition lock is key component for all the systems.

**SYSTEM DESIGN**

**CHAPTER 4**

**4 SYSTEM DESIGN**

This chapter will provide the system flow diagram and also explain the design of the various modules of the parking system. This system consists of some major phases which are described below.

**4.1 INTRODUCTION**

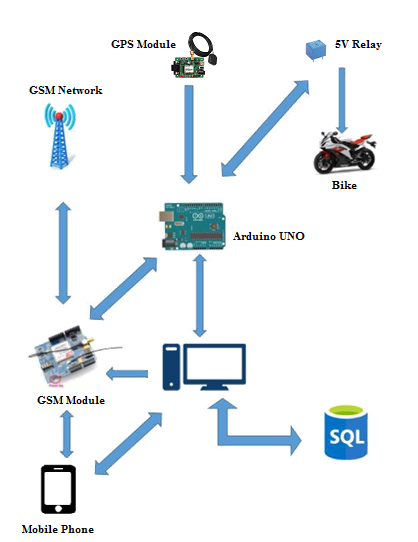
A sequence of modules are used in the system design. Each object embedded with Arduino Uno R3 board, GSM/GPRS sim800A modem, GPS module, 5V Relay and 5V Regulator (KA7805).

Programs can be loaded on to it from the easy-to-use Arduino computer program. The Arduino has an extensive support community, which makes it a very easy way to get started working with embedded electronics.

The Arduino can "talk", (transmit or receive data data) via a serial channel, so any other device with serial capabilities can communicate with an Arduino. It doesn't matter what program/programming language is driving the other device.

Serial is used for communication between the Arduino board and a computer or other devices. All Arduino boards have at least one serial port (also known as a UART or USART): **Serial**. It communicates on digital pins 0 (RX) and 1 (TX) as well as with the computer via USB. Thus, if you use these functions, you cannot also use pins 0 and 1 for digital input or output.

There is also an Arduino environment's built-in serial monitor to communicate with an Arduino board. Click the serial monitor button in the toolbar and select the same baud rate used in the call to begin ().



**Fig 4.1 Graphical View of the Bike Theft Detection System**

GSM modem depends on the service provider network. Only when the signal strength is excellent user can able to call and text. Arduino plays a vital role in this system. GSM, GPS, DC Motor through relay and Ignition Lock are serially connected to Arduino. PC communicates with Arduino through USB serial port. The GPS module fetches the current location data and send it to Arduino. The data from the Arduino is send to user mobile when the request are made. Only when bike is started without the key, the message is send to user automatically through GSM. User also can manually turn off the bike through SMS from his/her own mobile.

**4.2 FLOW CHART DIAGRAM**

**4.3 USE CASE DIAGRAM**

**4.4 DATABASE DESIGN**

**4.5 ARCHITECTURAL DIAGRAM**

**4.6 MODULES DESIGN**

* **LOCATION TRACKING AND ALERTING THE USER.**
* **CONTROLLING BIKE THROUGH GSM.**
* **NEAREST BIKE SERVICE LOCATION.**

**4.6.1 LOCATION TRACKING AND ALERTING THE USER:**

**4.6.2 CONTROLLING BIKE THROUGH GSM:**

**4.6.3 NEAREST BIKE SERVICE LOCATION:**

**4.7 ARDUINO IDE**

Arduino programs may be written in any programming language with a compiler that produces binary machine code. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio.

Arduino is an open-source hardware. The hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino website. Layout and production files for some versions of the hardware are also available. The source code for the IDE is released under the GNU General Public License, version 2.

The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It is designed to introduce programming to artists and other newcomers unfamiliar with software development.

It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and provides simple one-click mechanism to compile and load programs to an Arduino board. A program written with the IDE for Arduino is called a "sketch".

The Arduino IDE supports the languages C and C++ using special rules to organize code. The Arduino IDE supplies a software library called Wiring from the Wiring project, which provides many common input and output procedures. A typical Arduino C/C++ sketch consist of two functions that are compiled and linked with a program stub main () into an executable cyclic executive program:

**Setup ()**: a function that runs once at the start of a program and that can initialize settings.

**Loop ()**: a function called repeatedly until the board powers off.

A typical program for a beginning Arduino programmer blinks a light-emitting diode (LED) on and off. This program is usually loaded in the Arduino board by the manufacturer. In the Arduino environment, a user might write such a program as shown below.

#define LED\_PIN 13

void setup() {

pinMode(LED\_PIN, OUTPUT); // Enable pin 13 for digital output

}

void loop() {

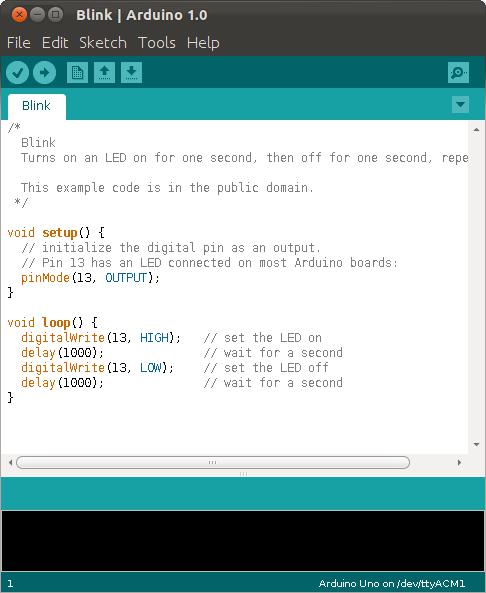
digitalWrite(LED\_PIN, HIGH); // Turn on the LED

delay(1000); // Wait one second (1000 milliseconds)

digitalWrite(LED\_PIN, LOW); // Turn off the LED

delay(1000); // Wait one second

}



**Fig 4.4 A sample program running in Arduino IDE**

After compiling and linking with the GNU toolchain, also included with the IDE distribution, the Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal coding that is loaded into the Arduino board by a loader program in the board's firmware.

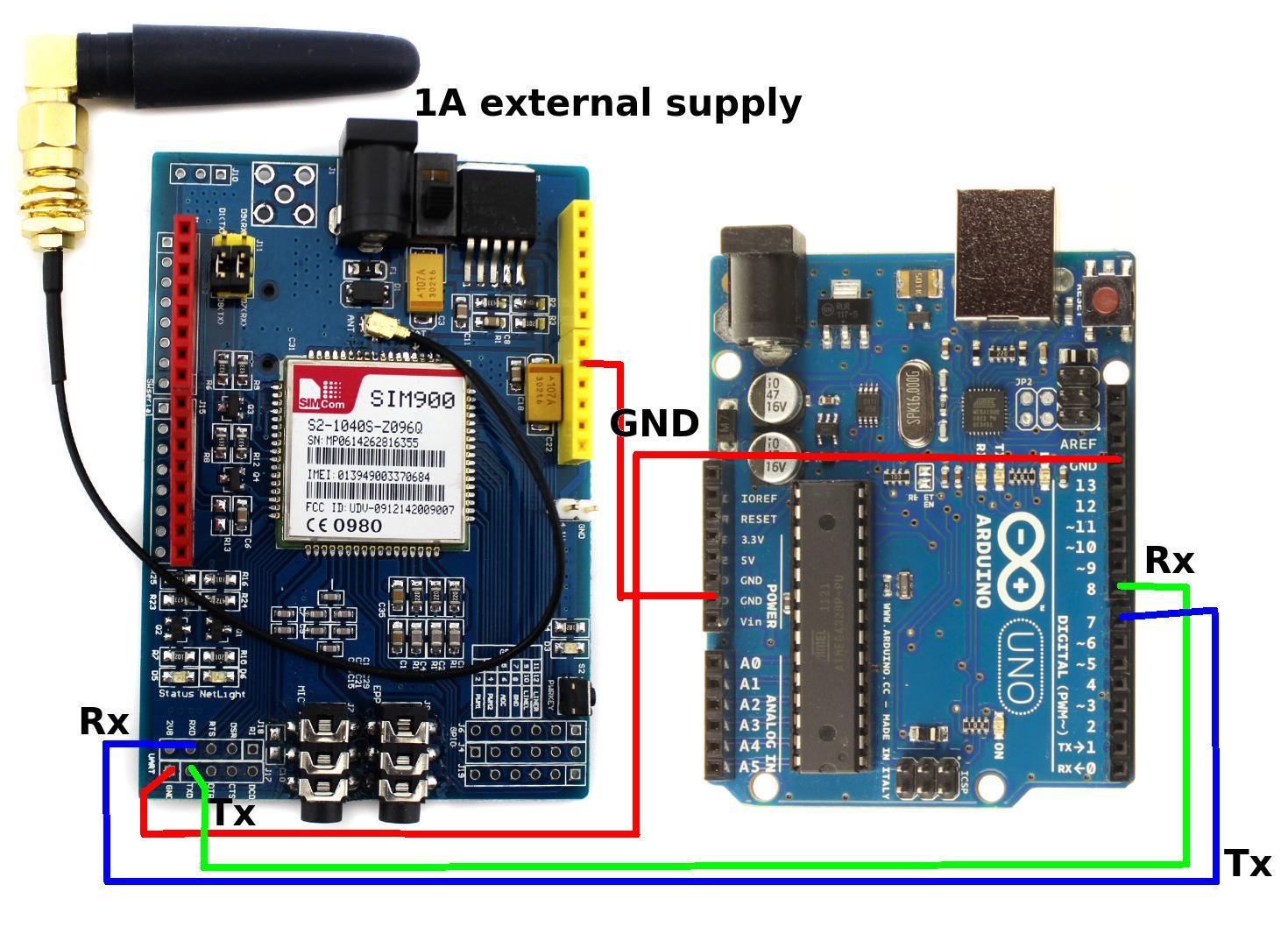
**SYSTEM IMPLEMENTATION**

**CHAPTER 5**

**5 SYSTEM IMPLEMENTATION**

A detailed Description of each of the different modules in the system design was provided in the previous chapter. As a continuation of that, this chapter will provide the information about the various functions deployed by our system and the corresponding implementation details pertaining to those functions.

**5.1 SIM900 GSM/GPRS MODEM – ARDUINO INTERFACING**



**Fig 5.1 Arduino-GSM interface**

**AT command used in GSM/GPRS Modem:**

AT commands-command used for SMS sending and receiving.

– AT-To check the modem

– AT+CMGF=1-To change to text mode

– AT+CMGS=“Mobile NO”- To send SMS to a new Number.

– AT+CMGR=”INDEX NO”- Read certain message

– AT+CNMI=2,0,0,0,0 – Receiving live messages.

– ATD “Mobile No” – Call to a number

– ATH – Hang a call

**Table 5.1: Arduino GSM Interfacing Pin Configuration**

|  |  |
| --- | --- |
| **GSM INTERFACING** | **ARDUINO INTERFACING** |
|  |  |
| Receiver Pin (Rx) | Transmitter Pin(Tx) Pin 0 |
|  |  |
| Transmitter Pin(Tx) | Receiver Pin (Rx) Pin 1 |
|  |  |
| GND | GND of Arduino. |
|  |  |
| Vcc | 5V of Arduino. |
|  |  |

**5.2 GPS MODULE-ARDUINO INTERFACING**

**5.3 DC MOTOR-IGNITION LOCK-ARDUINO INTERFACING:**

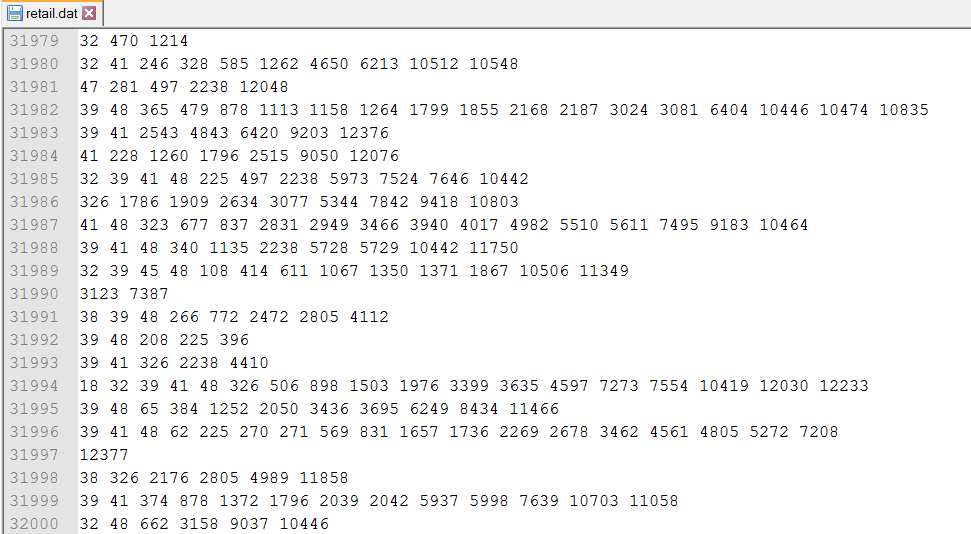


**CHAPTER 6**

**TEST CASES AND TEST RESULTS**

**6.1 TEST CASES AND RESULT**

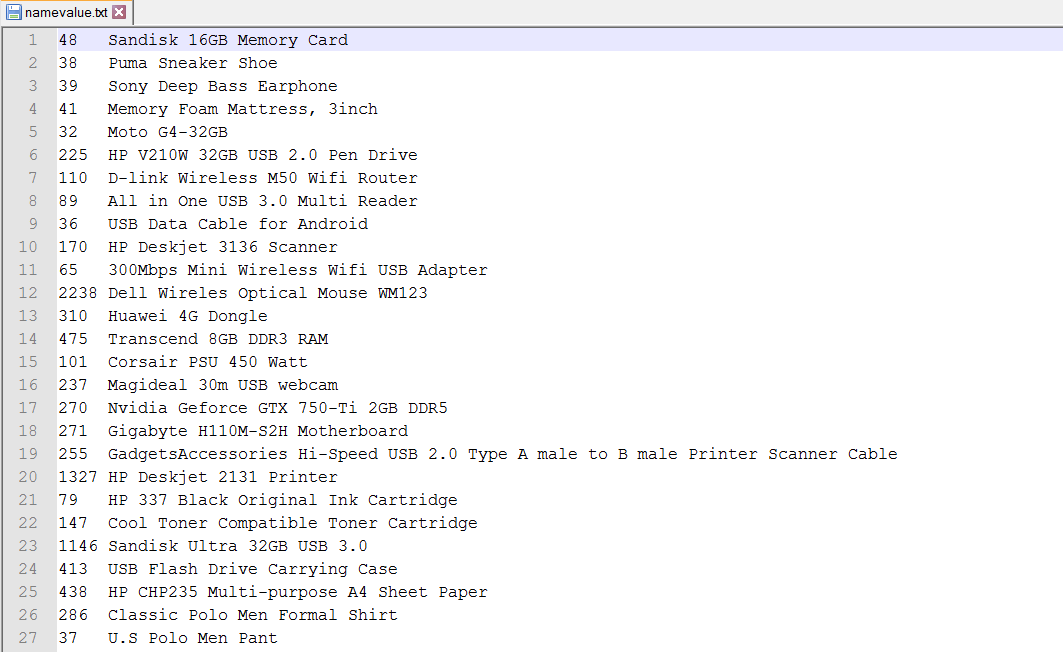
**6.1.1 Input Data**

****

**Figure 6.1:** Transaction data

Figure 6.1 shows the input data (i.e.) the online transaction details of the retailer from the Data Repository. The each line represents a single transaction and numbers in each transaction is the product id. This is the initial input data used in finding the frequent item sets.

The transaction data consists of nearly 80,000 transactions and more than 600 items. These transaction data is used for finding the cluster of frequent item sets.



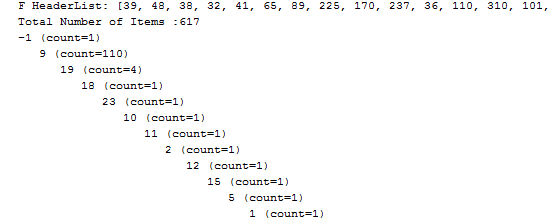
**Figure 6.2:** Product ID and its corresponding Name

Figure 6.2 shows the product id of the product and the corresponding name of the product used in the transaction data. The NamingItem class uses this file as input to name the product id with the corresponding value.

**6.1.2 Filter Single Item Order Testing and Results**

**Table 6.1:** Test case for Filter Single Item Order

|  |  |
| --- | --- |
| ***Test Case ID*** | ***T1*** |
| ***Test Case Description*** | *Construction of FP tree* |
| ***Test Data*** | *A transaction database DB and a minimum support threshold.* |
| ***Expected output*** | *FP-tree, the frequent-pattern tree of DB.* |
| ***Test Result*** | *PASS* |



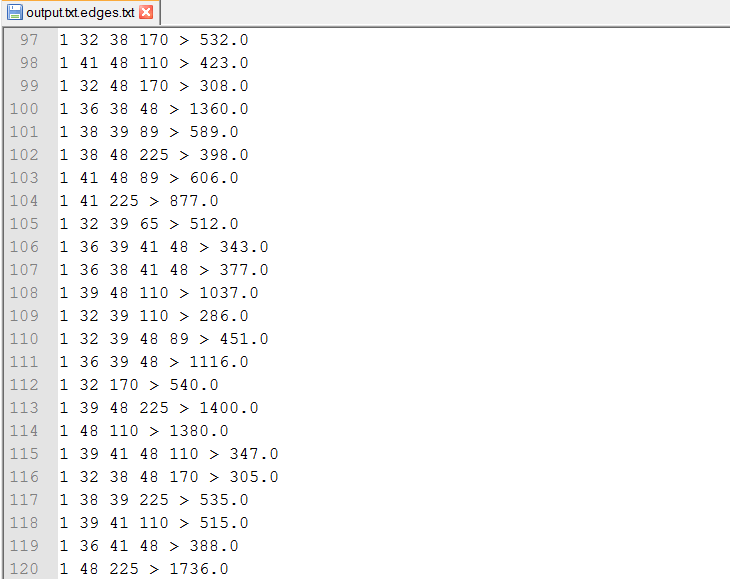
**Figure 6.3:** Frequent Pattern Tree

Figure 6.3 shows the Filtering Single Item Order by construction of the FP Tree which contains all the transaction in the transaction repository. The Frequent Pattern tree structure which contains the all the transaction data above the support threshold is constructed for the item sets.

**6.1.3 Frequent Item Set Mining Testing and Results**

**Table 6.2:** Test case for Frequent Item SetMining

|  |  |
| --- | --- |
| ***Test Case ID*** | ***T2*** |
| ***Test Case Description*** | *Construction of FP growth tree* |
| ***Test Data*** | *A database DB, represented by FP-tree constructed, and a minimum support threshold* |
| ***Expected output*** | *The complete set of frequent patterns* |
| ***Test Result*** | *PASS* |

****

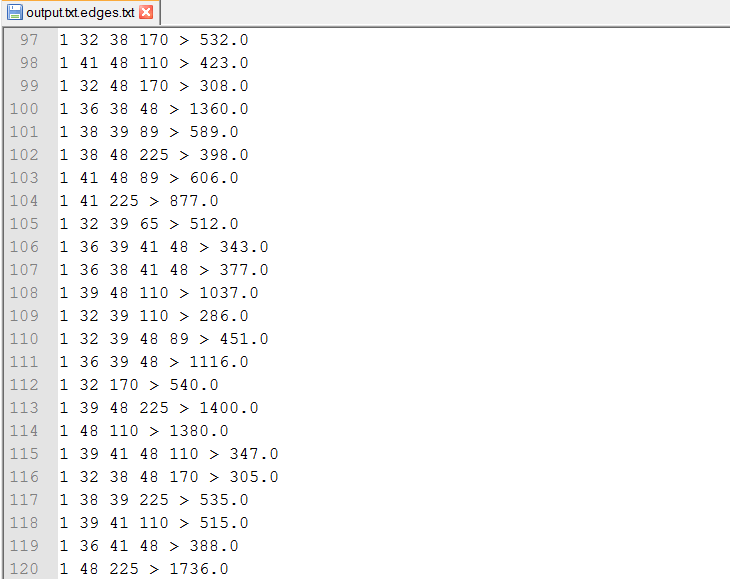
**Figure 6.4 :** FP Growth Tree

Figure 6.4 shows the frequent item set mining by the construction of the FP growth tree. The FP tree with a minimum threshold is given as input and a complete set of frequent patterns are generated with the support count at the end of each transactions which are represented after ‘>’ symbol.

**6.1.4 Hyper Graph Construction Testing and Results**

**Table 6.3:** Test case for Hyper Graph Construction

|  |  |
| --- | --- |
| ***Test Case ID*** | ***T3*** |
| ***Test Case Description*** | *Construction of Hyper Graph* |
| ***Test Data*** | *FP growth tree with weighted edges* |
| ***Expected output*** | *Hyper Graph edges* |
| ***Test Result*** | *PASS* |

****

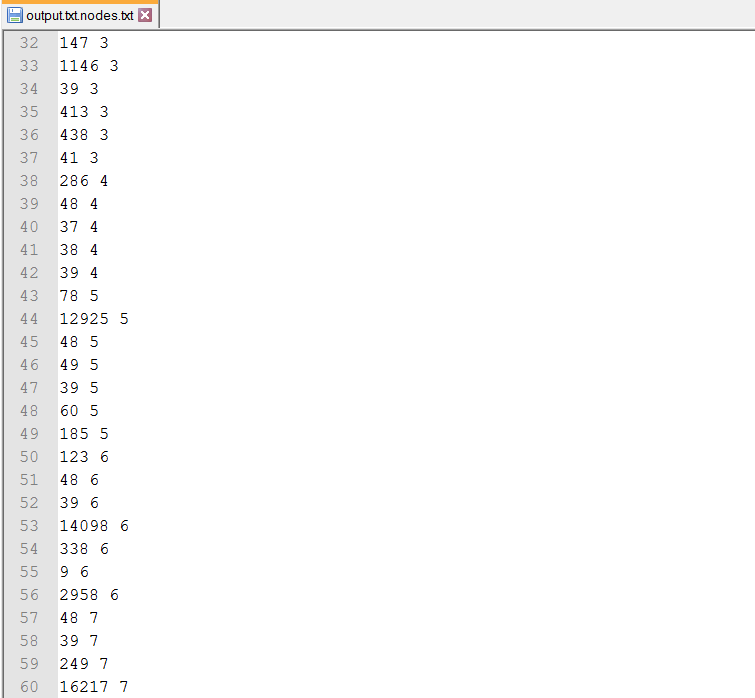
**Figure 6.5:** Hyper Graph edges

Figure 6.5 shows the hyper graph edges constructed from the FP growth tree. The single item transactions are eliminated while constructing hyper graph from the weighted graph.

**6.1.5 Hyper Graph Clustering Testing and Results**

**Table 6.4:** Test case for Hyper Graph Clustering

|  |  |
| --- | --- |
| ***Test Case ID*** | ***T4*** |
| ***Test Case Description*** | *Clustering of Frequent Items.* |
| ***Test Data*** | *Hyper graph edges* |
| ***Expected output*** | *Product Clusters* |
| ***Test Result*** | *PASS* |

****

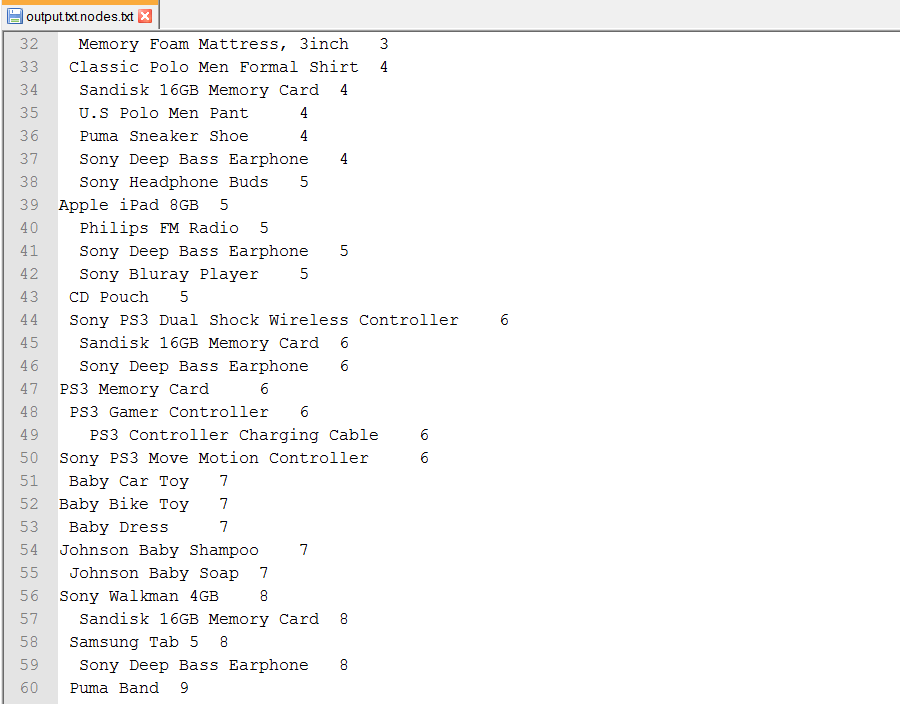
**Figure 6.6:** Hyper Graph clustering

Figure 6.6 shows the cluster of frequent item sets obtained by the clustering of hyper graph. The hyper graph is clustered using bi-partitioning algorithm to form the product clusters.

**6.1.6 Naming the Product IDs**

**Table 6.5:** Test Case for Naming the Product ID

|  |  |
| --- | --- |
| ***Test Case ID*** | ***T5*** |
| ***Test Case Description*** | *Naming the product ID with their corresponding names* |
| ***Test Data*** | *namevalue.txt* |
| ***Expected output*** | *output file with name and its clusters* |
| ***Test Result*** | *PASS* |

****

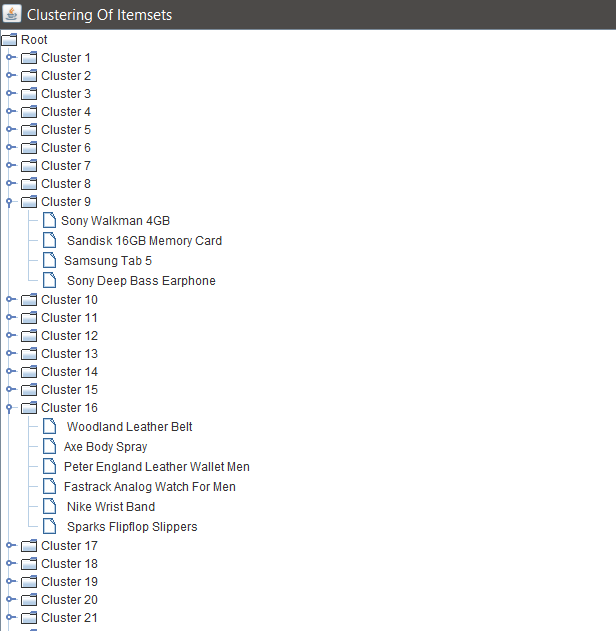
**Figure 6.7:** Naming with product id.

Figure 6.7 shows the list of product clusters whose product id is assigned with corresponding names using Hashmap.

**6.1.7 Representation of Clusters**

**Table 6.6:** Test Case for Representation of Clusters

|  |  |
| --- | --- |
| ***Test Case ID*** | ***T6*** |
| ***Test Case Description*** | *Representation of the clusters in a Jtree structure* |
| ***Test Data*** | *Output file consists of name and its clusters* |
| ***Expected output*** | *Jtree structure with clusters and its product* |
| ***Test Result*** | *PASS* |

****

**Figure 6.8:** Representation of clusters.

Figure 6.8 shows the list of clusters which are obtained after the hyper graph clustering are represented using the Jtree.

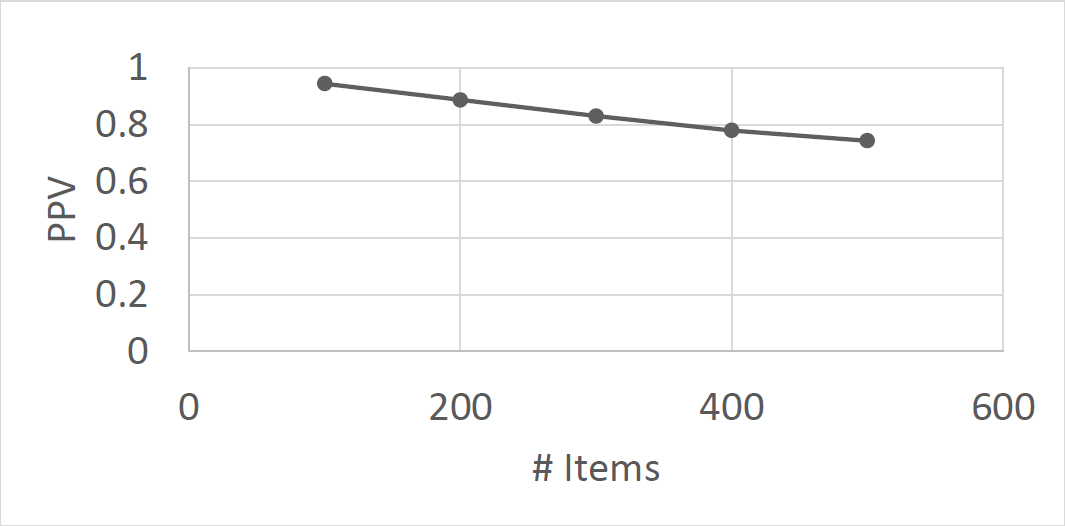
**6.1.8 Positive Predictive Value (PPV)**

Positive Predictive Value can be defined as follows

𝑃𝑃𝑉 = 𝑛𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝑡𝑟𝑢𝑒 𝑝𝑜𝑠𝑖𝑡𝑖𝑣𝑒𝑠

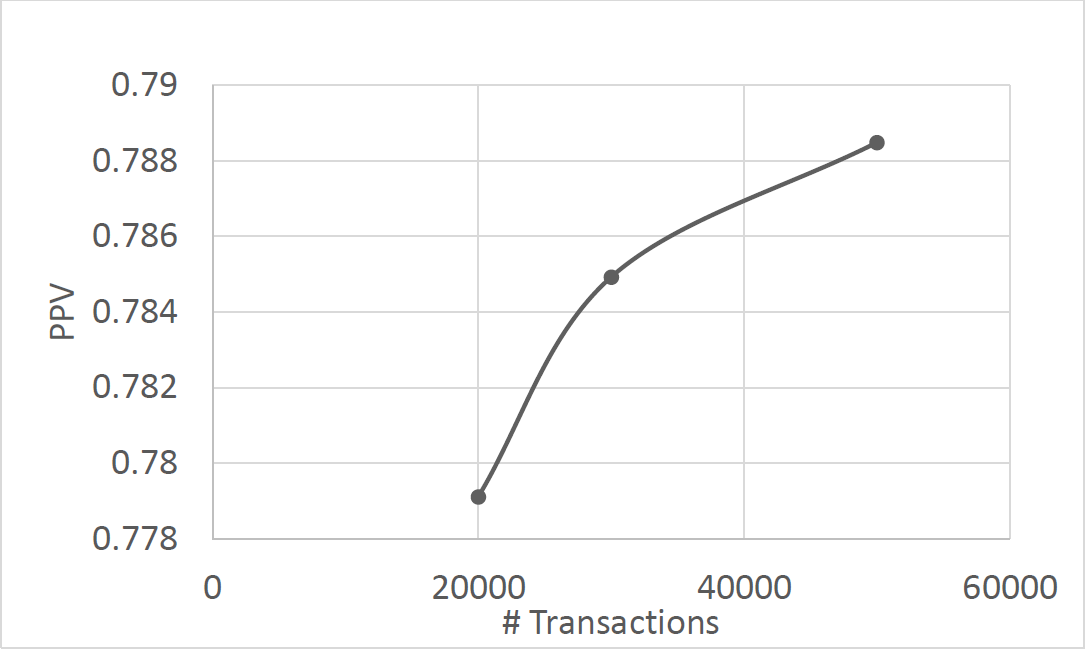
𝑛𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝑡𝑟𝑢𝑒 𝑝𝑜𝑠𝑖𝑡𝑖𝑣𝑒𝑠 + 𝑛𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝑓𝑎𝑙𝑠𝑒 𝑝𝑜𝑠𝑖𝑡𝑖𝑣𝑒𝑠

The value of PPV when the number of items is increased with fixed order count was studied. The result is shown in fig 6.9. The number of orders were fixed at 80,000. It is clear from the chart that as number of items increases the PPV decreases.



**Figure 6.9:** PPV vs number of items

Keeping the number of distinct items as 500 the effect of number of transaction is studied (Fig 6.10). When the number of transactions increase PPV increases. This is because more data is available for learning leading to better prediction. Using this data the number of transaction required to produce a particular PPV can be found given the number of distinct items.

****

**Figure 6.10:** PPV vs number of transactions

**CHAPTER 7**

**CONCLUSION**

An algorithm to group products that are likely to be bought together has been designed and implemented. This is a two stage algorithm. In the first stage frequent item sets are found. The actual clusters are then identified by mapping frequent item sets to hyper graph and using hyper graph clustering algorithms.

This algorithm can be used in making inventory placement decisions. Products that are usually bought together can be placed in same warehouse so that multi-item orders can be shipped together.

**FUTURE ENHANCEMENT**

Future optimizations to the algorithm can help the online retailers who are at different locations to sell their product by packing them in the same package which are in the same cluster.

In current methodology the relationship between orders made by same user has not been used. Making use of this may produce better clusters. Also items belonging to same product category are more likely to be bought together. Such features can also be brought into the algorithm.

Number of orders made through online portals are increasing. A framework for clustering has been proposed in this work. Scalability of the algorithm for large datasets has to be studied in detail.

**Appendix A**

**SOURCE CODE**