**RV COLLEGE OF ENGINEERING®**

**(Autonomous Institution, Affiliated to VTU, Belagavi) Bengaluru, Karnataka– 560059**


**Lab Manual for 7th Semester**

# DATA SCIENCE AND ENGINEERING

# (16IS72)

## Department of Information Science & Engineering

**Faculty In-charge**

**Prof. Smitha G R**

| | |
|---|---|
| **USN** | **1RV17IS011** |
| **NAME** | **Jayanth G** |
| **ACADEMIC YEAR** | **2020-21** |

# Vision, Mission, PEO, PO and PSO of the department

## Vision

**To be the hub for innovation in Information Science & Engineering through Teaching, Research, Development and Consultancy; thus make the department a well known resource center in advanced sustainable and inclusive technology.**

## Mission

**ISE1: To enable students to become responsible professionals, strong in fundamentals of information science and engineering through experiential learning.**

**ISE2: To bring research and entrepreneurship into class rooms by continuous design of innovative solutions through research publications and dynamic development oriented curriculum.**

**ISE3: To facilitate continuous interaction with the outside world through student internship, faculty consultancy, workshops, faculty development programmes, industry collaboration and association with the professional societies.**

**ISE4: To create a new generation of entrepreneurial problem solvers for a sustainable future through green technology with an emphasis on ethical practices, inclusive societal concerns and environment.**

**ISE5: To promote team work through inter-disciplinary projects, co-curricular and social activities.**

## Program Educational Objectives (PEOs)

**PEO1: To provide adaptive and agile skills in Information Science and Engineering needed for professional excellence / higher studies /Employment, in rapidly changing scenarios.**

**PEO2: To provide students a strong foundation in basic sciences and its applications to technology.**

**PEO3: To train students in core areas of Information science and Engineering, enabling them to analyze, design and create products and solutions for the real world problems, in the context of changing technical, financial, managerial and legal issues.**

**PEO4: To inculcate leadership, professional ethics, effective communication, team spirit, multi-disciplinary approach in students and an ability to relate Information Engineering issues to social and environmental context.**

**PEO5: To motivate students to develop passion for lifelong learning, innovation, career growth and professional achievement.**

**Program Outcomes (PO)**

1. **Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.**

2. **Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.**

3. **Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.**

4. **Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.**

5. **Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.**

6. **The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.**

7. **Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.**

8. **Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.**

9. **Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.**

10. **Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.**

11. **Project management and finance: Demonstrate knowledge and understanding of the Engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.**

12. **Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.**

**Program Specific Outcome (PSO)**

**PSO-1**

**Recognize and appreciate the principles of theoretical foundations, data organization, data communication, security and data analytical methods in the evolving technology**

**PSO-2**

**Learn the applicability of various system softwares for the development of quality products in solving real-world problems with a focus on performance optimization**

**PSO-3**

**Demonstrate the ability of team work, professional ethics, communication and documentation skills in designing and implementation of software products using the SDLC principles**

**(Autonomous Institution Affiliated to VTU, Belagavi)**

## Department of Information Science & Engineering


# CERTIFICATE

**This is to certify that Mr. SUDARSHAN( USN: 1RV17IS049) of 7th semester Information Science and Engineering has satisfactorily completed the course of experiments in practical of Data Science and Engineering (16IS72) during the academic year 2020-21.**

| LAB MARKS | |
|---|---|
| **Max Marks** | **Obtained** |
| **50** | |

**Signature of the Student** **Signature of the Faculty In–Charge**

**Signature of HOD**

# General Laboratory Instructions

| | |
|---|---|
| | **PART – A**<br>**Program Execution** |
| **1.** | **Student has to do all the experiments in the collage lab only** |
| **2.** | **Students can solve the experiments by using Python or R tool.** |
| | **PART – B**<br>**Case Study implementation** |
| | **Student shall finalize the topic in consultation with the teacher** |

## General Guidelines

**The lab manual should contain duly signed data sheets with programs and results in the respective program slot given.**

**Each program in PART A is evaluated for 10 marks (7 marks for execution + 3 marks for viva voce) and the average of 09 programs is considered for 30 marks.**

## Guidelines for Case Study

**Case Study is carried out in ISE lab with maximum of two members in a team.**

**Complexity of the casestudy will be discussed with the faculty – In charge of the lab and then allotment of the same would be confirmed.**

**Students are required to strict to the schedule without fail.**

**Acknowledgement: Prof. Poornima K, ISE, RVCE**

**Evaluation distribution table for lab record**

| PART A | PART B | Internals | Total Marks |
|---|---|---|---|
| 30 | 10 | 10 | 50 |

**Scheme of Continuous Internal Evaluation for Practical**

**CIE consists of 50 marks out of which 30 marks for PART-A, 10 marks for PART-B (40) and 10 marks for test conducted at the end of semester.**

**Scheme of Semester End Examination for Practical**

**SEE is evaluated for 50 marks which include writing correct program, execution and viva voce.**

1. **In the examination, ONE program has to be asked from Part-A for a total of 50 marks.**
2. **The Case Study impleted under Part-B not to be evaluated for SEE**

## V COLLEGE OF ENGINEERING®, BENGALURU – 560059
## (Autonomous Institution Affiliated to VTU, Belagavi)

### Data Science and Engineering – 16IS72

### Laboratory Evaluation

| Week | Program No. | Program Evaluation | Date of Execution | Marks (10) | Signature of Staff |
|---|---|---|---|---|---|
| 1 | 1 | Process the Movie dataset and visualize the correlations | | | |
| 2 | 2 | Implement data preprocessing techniques | | | |
| 3 | 3 | Implement simple linear regression and multiple linear regression using relevant datasets for prediction. | | | |
| 4 | 4 | Implement k- nearest neighbour algorithm using relevant datasets. | | | |
| 5 | 5 | Implement decision tree algorithm for classification using relevant datasets. | | | |
| 6 | 6 | Implement Naïve bayes classification using relevant datasets. | | | |
| 7 | 7 | Implement support vector machine using relevant datasets. | | | |
| 8 | 8 | Implement Association rule process using Apriori algorithm using relevant datasets. | | | |
| 9 | 9 | Implement K- means clustering to classify the clusters in a given data set | | | |
| | | **Total for program execution:** | **/90** | | |
| | | **CIE for program execution :** | **/30** | | |

### Case Study Evaluation

| Activity No. | Activity | Date of Submission | Marks (10 marks ) | Signature of Staff |
|---|---|---|---|---|
| 1 | Case study topic finalization | | - | |
| 2 | Finalization of Data Source and model planning based on empirical studies | | 2 | |
| 3 | Building the model (Ensemble model) | | 2 | |
| 4 | Evaluation of the model | | 2 | |
| 5 | Submission of Report and other deliverables | | 2 | |
| 6 | | | | |

| | |
|---|---|
| **Total for casestudy execution:** | **/10** |
| **CIE for casestudy execution :** | **/10** |
| **Internal Conduction (10 marks)**<br>1. **Program execution -05 marks - (Writeup-1m ,Execution- 3m ,Viva- 1m) -**<br>2. **Case Study Demonstration – 05 marks –** | **Final CIE marks –**<br>**/50** |

# Rubrics for Data Science and Engineering

**Each program is evaluated for 10 marks – Write up & Execution = 7 Marks, Viva Voce = 3 Marks**

| Lab Write-up and Execution rubrics (Max: 7 marks) | | | | | | |
|---|---|---|---|---|---|---|
| **Sl no** | **Criteria** | **Measuring methods** | **Excellent** | **Good** | **Poor** | **CO** |
| 1 | Understanding of problem and requirements<br><br>(2 Marks) | Observations | Student exhibits thorough understanding of program requirements and applies the concepts of Data Science to develop a model (2M) | Student has sufficient understanding of program requirements and applies concepts of Data Science to develop a model ( 1M) | Student does not have clear understanding of program requirements and is unable to apply Data science concepts learnt for development. (0M) | CO 1 |
| 2 | Execution<br><br>(3Marks) | Observations | Student demonstrates the execution of the program with optimized code with all the necessary conditions (3M) | Student demonstrates the execution of the program without optimization of the code (2M-1M) | Student has not executed the program. (0 M) | CO 4 |
| 3 | Results and Documentation<br><br>(2Marks) | Observations | Documentation with appropriate comments and output is covered in data sheets and manual. (2M) | Documentation with only few comments and only few output cases is covered in data sheets and manual. (1M) | Documentation with no comments and no output cases is covered in data sheets and manual. (0 M) | CO 2 |
| **Viva Voce rubrics (Max: 3 marks)** | | | | | | |
| 1 | Conceptual Understanding and Communication of concepts<br><br>(2 Marks) | Viva Voce | Explains Data Science concepts with Python/R and related concepts involved. (2M) | Adequately explains the Data Science concepts with Python/R and related concepts involved. (1M) | Unable to explain the concepts. (0M) | CO 1 |
| 2 | Use of appropriate Design Techniques<br><br>(1 Mark) | Viva Voce | Insightful explanation of appropriate design techniques for the given problem to derive solution. (1 M) | Sufficiently explains the use of appropriate design techniques for the given problem to derive solution. (0.5 M) | Unable to explain the design techniques for the given problem. (0 M) | CO 3 |

## Program No. 1

**Title:**

- **Process the Movie dataset and visualize the correlations**

**using R Objective:**

- **To write a program to visualize the correlations in**

**a dataset Reference:**

- **Data Mining Concept and Technique By Han & Kamber**

**Theory:**

**Step 1: Importing The Data**

**In order to access the movies data set and put it to use, use the read.csv() function to import your data into a data frame and store it in the variable with the stunningly original name movies.**

**Step 2: Basic Inspection of Your Data**

**It's a good idea, once a data frame has been imported, to get an idea about your data. First, check out the structure of the data that is being examined. Use function str(): The console lists each variable by name, the class of each variable, and a few instances of each variable. This gives good idea of what is in the data frame, the understanding of which is crucial to our analytic endeavors.**

**Another great function to help perform a quick, high-level overview of our data frame is summary(). Note the similarities and differences between the output produced by running str().**

**Step 3:**

**In reviewing the variables available to you, it appears that some of the numeric variables can be manipulated to provide new insights into our data frame.**

**For instance, to get gross and budget variables. Calculate profit by using the formula profit = gross - budget.**

**Step 4:Correlation**

**Now that profit has been added as a new column in our data frame, it's time to take a closer look at the relationships between the variables of your data set. Let's check out how profit fluctuates relative to each movie's rating.For this, you can use R's built in**

**plot and abline functions, where plot will result in a scatter plot and abline will result in a regression line or "line of best fit" due to our inclusion of the linear model argument**

**Step 5:Calculating Correlation in R**

**To identify the types of relationships exist between the variables in movies, and how can you evaluate those relationships quantitatively?**

**The first way is to produce correlations and correlation matrices with cor(): Visually Exploring Correlation: The R Correlation Matrix Plot a correlation matrix using the variables available in your movies data frame. This simple plot will enable to quickly visualize which variables have a negative, positive, weak, or strong correlation to the other variables. To pull this wizardry off, use a nifty package called GGally and a function called ggcorr(). The form of this function call will be ggcorr(df), where df is the name of the data frame you're calling the function on. The output of this function will be a triangular-shaped, color-coded matrix labelled with our variable names. The correlation coefficient of each variable relative to the other variables can be found by reading across and / or down the matrix, depending on the variable's location in the matrix.**

| Program No. | Marks for Execution (7) | | | Marks for Viva voce (3) | | TOTAL (10) | Signature of the Faculty |
|---|---|---|---|---|---|---|---|
| | Rubrics | | | Rubrics | | | |
| | Understanding of problem (2) | Execution (3) | Results and Documentation (2) | Conceptual Understanding and Communication of Concepts (2) | Use of appropriate Design Techniques (1) | | |
| 1 | | | | | | | |

## Program No. 1

**Paste your DATA SHEET here**

In [3]:
```
data = pd.read_csv("./datasets/P1_movies_metadata.csv")
data.head()
```

Out[3]:

| | adult | belongs_to_collection | budget | genres | homepage | id | imdb_id |
|---|---|---|---|---|---|---|---|
| **0** | False | {'id': 10194, 'name': 'Toy Story Collection', ... | 30000000 | [{'id': 16, 'name': 'Animation'}, {'id': 35, '... | http://toystory.disney.com/toy-story | 862 | tt0114709 |
| **1** | False | NaN | 65000000 | [{'id': 12, 'name': 'Adventure'}, {'id': 14, '... | NaN | 8844 | tt0113497 |
| **2** | False | {'id': 119050, 'name': 'Grumpy Old Men Collect... | 0 | [{'id': 10749, 'name': 'Romance'}, {'id': 35, ... | NaN | 15602 | tt0113228 |
| **3** | False | NaN | 16000000 | [{'id': 35, 'name': 'Comedy'}, {'id': 18, 'nam... | NaN | 31357 | tt0114885 |
| **4** | False | {'id': 96871, 'name': 'Father of the Bride Col... | 0 | [{'id': 35, 'name': 'Comedy'}] | NaN | 11862 | tt0113041 |

5 rows × 24 columns

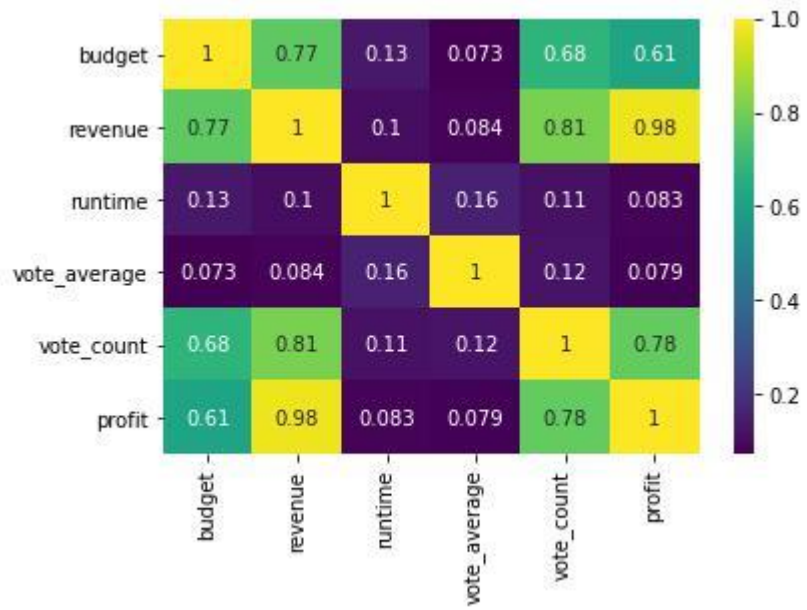
In [4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45466 entries, 0 to 45465
Data columns (total 24 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   adult                  45466  non-null object 
 1   belongs_to_collection  4494 non-null   object 
 2   budget                 45466  non-null object 
 3   genres                 45466  non-null object 
 4   homepage               7782 non-null   object 
 5   id                     45466  non-null object 
 6   imdb_id                45449  non-null object 
 7   original_language      45455  non-null object 
 8   original_title         45466  non-null object 
 9   overview               44512  non-null object 
 10  popularity             45461  non-null object 
 11  poster_path            45080  non-null object 
 12  production_companies   45463  non-null object 
 13  production_countries   45463  non-null object 
 14  release_date           45379  non-null object 
 15  revenue                45460  non-null float64
 16  runtime                45203  non-null float64
 17  spoken_languages       45460  non-null object 
 18  status                 45379  non-null object 
 19  tagline                20412  non-null object 
 20  title                  45460  non-null object 
 21  video                  45460  non-null object 
 22  vote_average           45460  non-null float64
 23  vote_count             45460  non-null float64
dtypes: float64(4), object(20)
memory usage: 8.3+ MB
```

In [7]:
```
import matplotlib.pyplot as plt
import seaborn as sn
corr = data.corr()
print(corr)
```

| | budget | revenue | runtime | vote_average | vote_count | profit |
|---|---|---|---|---|---|---|
| budget | 1.000000 | 0.768776 | 0.134733 | 0.073494 | 0.676642 | 0.614339 |
| revenue | 0.768776 | 1.000000 | 0.103917 | 0.083868 | 0.812022 | 0.976896 |
| runtime | 0.134733 | 0.103917 | 1.000000 | 0.158146 | 0.113539 | 0.083189 |
| vote_average | 0.073494 | 0.083868 | 0.158146 | 1.000000 | 0.123607 | 0.078916 |
| vote_count | 0.676642 | 0.812022 | 0.113539 | 0.123607 | 1.000000 | 0.775756 |
| profit | 0.614339 | 0.976896 | 0.083189 | 0.078916 | 0.775756 | 1.000000 |

In [8]:
```
sn.heatmap(corr, annot=True,cmap='viridis')
plt.show()
```




In [13]:
```
from collections import Counter
genres_json = list(data['genres'])

import json
genres_clean = []
for g in genres_json:
    g = g.replace("\'","\"")
    genres_clean.append(json.loads(g))

genres = []
for a in genres_clean:
    for b in a:
        genres.append(b['name'])

g_counts = dict(Counter(genres))
g_counts
```

Out[13]:
```
{'Animation': 1935,
 'Comedy': 13182,
 'Family': 2770,
 'Adventure': 3496,
 'Fantasy': 2313,
 'Romance': 6735,
```

```
'Drama': 20265,
'Action': 6596,
'Crime': 4307,
'Thriller': 7624,
'Horror': 4673,
'History': 1398,
'Science Fiction': 3049,
'Mystery': 2467,
'War': 1323,
'Foreign': 1622,
'Music': 1598,
'Documentary': 3932,
'Western': 1042,
'TV Movie': 767,
'Carousel Productions': 1,
'Vision View Entertainment': 1,
'Telescene Film Group Productions': 1,
'Aniplex': 1,
'GoHands': 1,
'BROSTA TV': 1,
'Mardock Scramble Production Committee': 1,
'Sentai Filmworks': 1,
'Odyssey Media': 1,
'Pulser Productions': 1,
'Rogue State': 1,
'The Cartel': 1}
```

## Program No. 2

**Title:**

**Data Preprocessing Techniques for Data**

**Mining Objective:**

**To write a program to preprocess the data for building a model**

**Reference:**

**Data Mining Concept and Technique By Han & Kamber**

**Theory:**

**Data Pre-processing Methods**

**Raw data is highly susceptible to noise, missing values, and inconsistency. The quality of data affects the data mining results. In order to help improve the quality of the data and, consequently, of the mining results raw data is pre-processed so as to improve the efficiency and ease of the mining process. Data preprocessing is one of the most critical steps in a data mining process which deals with the preparation and transformation of the initial dataset. Data preprocessing methods are divided into following categories: Data Cleaning**

**Data Integration**

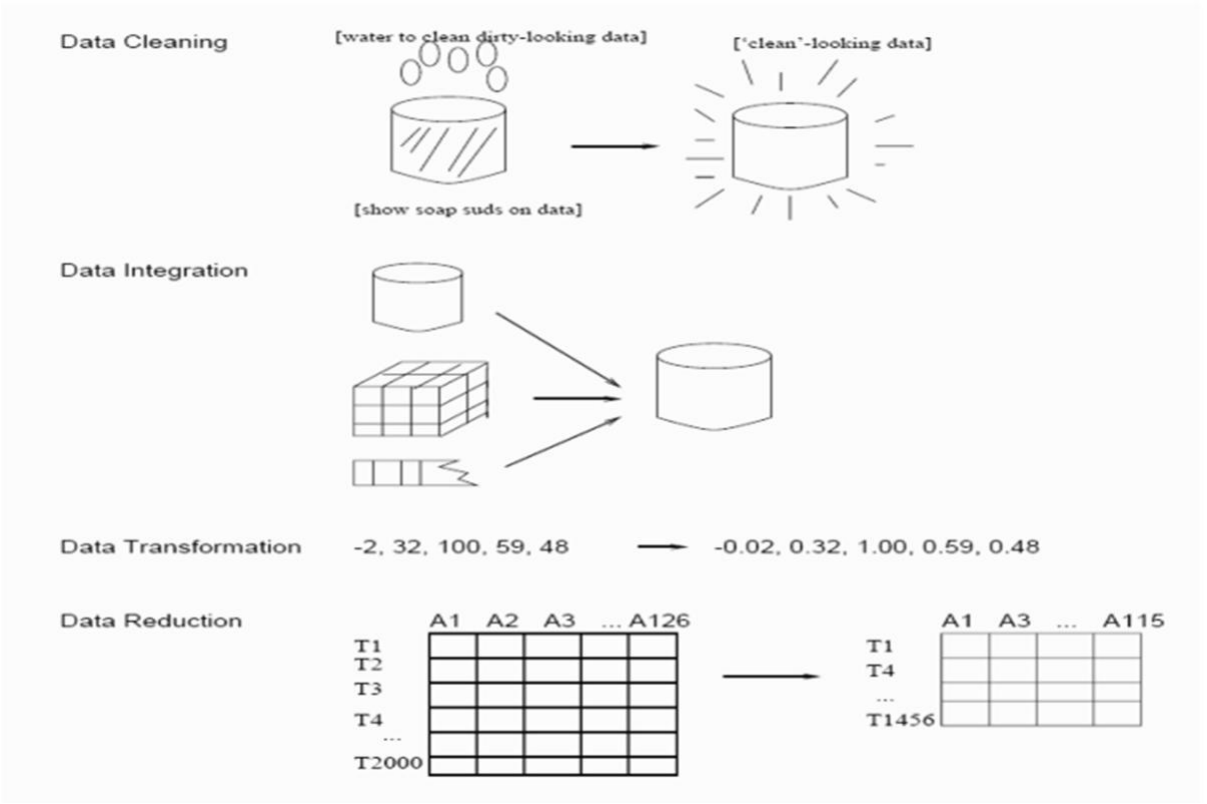
**Data Transformation**

**Data Reduction**

### Data Cleaning

**Data that is to be analyze by data mining techniques can be incomplete (lacking attribute values or certain attributes of interest, or containing only aggregate data), noisy (containing errors, or outlier values which deviate from the expected), and inconsistent (e.g., containing discrepancies in the department codes used to categorize items).Incomplete, noisy, and inconsistent data are commonplace properties of large, real-world databases anddata warehouses. Incomplete data can occur for a number of reasons. Attributes of interest may not always be availab le, such as customer information for sales transaction data. Other data may not be included simply because it was not considered important at the time of entry. Relevant data may not be recorded due to a misunderstanding, or because of**

**equipment malfunctions. Data can be noisy, having incorrect attribute values, owing to the following. The data collection instruments used may be faulty. There may have been human or computer errors occurring at data entry. Errors in data transmission can also occur. There may be technology lim itations, such as limited buffer size for coordinating synchronized data transfer and consumption. Dirty data can cause confusion for the mining procedure. Although most mining routines have some procedures for dealing with incomplete or noisy data, they are not always robust. Instead, they may concentrate on avoiding over fitt ing the data to the function being modelled. Therefore, a useful preprocessing step is to run data through some data cleaning routines.**




## Forms of Data Preprocessing

**Missing Values: If it is noted that there are many tuples that have no recorded value for several attributes, then the missing values can be filled in for the attribute by various methods described below:**

**1. Ignore the tuple: This is usually done when the class label is missing (assuming the mining task involves c lass if ication or description). This method is not very**

**effective, unless the tuple contains several attributes with missing values. It is especially poor when the percentage of missing values per attribute varies considerably.**

**2. Fill in the missing value manually: In general, this approach is time-consuming and**

**may not be feasible given a large data set with many missing values.**

**3. Values are replaced by, say, Unknown", then the mining program may mistakenly think that they form an interesting concept, since they all have a value in common that of Unknown". Hence, although this method is simple, it is not recommended.**

**4. Use the attribute mean to fi ll in the missing value**

**5. Use the attribute mean for all samples belonging to the same class as the given tuple.**

**6. Use the most probable value to fill in the missing value: This may be determined with inference-based tools using a Bayesian formalism or decision tree induction.**

**Methods 3 to 6 bias the data. The filled-in value may not be correct. Method 6, however, is a popular strategy. In comparison to the other methods, it uses the most information from the present data to predict missing values.**

**Noisy Data: Noise is a random error or variance in a measured variable. Given a numeric attribute such as, say, price, how can the data be "smoothed" to remove the noise? The following data smoothing techniques describes this.**

**1. Binning methods: Binning methods smooth a sorted data value by consulting the neighborhood", or values around it. The sorted values are distributed into a number of 'buckets', or bins. Because binning methods consult the neighborhood of values, they perform local smoothing values around it. The sorted values are distributed into a number of 'buckets', or bins. Because binning methods consult the neighborhood of values, they perform local smoothing.**

**2. Clustering: Outliers may be detected by clustering, where similar values are organized into groups or \clusters".**

**3. Combined computer and human inspection: Outliers may be identified through a**

combination of computer and human inspection. In one application, for example, an information-theoretic measure was used to help identify outlier patterns in a handwritten character database for classification. The measure's value reflected the surprise" content of the predicted character label with respect to the known label. Outlier patterns may be informative (e.g., identifying useful data exceptions, such as different versions of the characters \0" or \7"), or \garbage" (e.g., mislabeled characters). Patterns whose surprise content is above a threshold are output to a list. A human can then sort through the patterns in the list to identify the actual garbage ones.

This is much faster than having to manually search through the entire database. The garbage patterns can then be removed from the (training) database.

4. Regression: Data can be smoothed by fitting the data to a function, such as with regression. Linear regression involves finding the \best" line to fit two variables, so that one variable can be used to predict the other. Multiple linear regression is an extension of linear regression, where more than two variables are involved and the data are fit to a multidimensional surface. Using regression to find a mathematical equation to fit the data helps smooth out the noise.

Inconsistent data: There may be inconsistencies in the data recorded for some transactions. Some data inconsistencies may be corrected manually using external references. For example, errors made at data entry may be corrected by performing a paper trace. This may be coupled with routines designed to help correct the inconsistent use of codes. Knowledge engineering tools may also be used to detect the violation of known data constraints. For example, known functional dependencies between attributes can be used to find values contradicting the functional constraints.

## Data Integration

It is likely that your data analysis task will involve data integration, which combines data from multiple sources into a coherent data store, as in data warehousing. These sources may include multiple databases, data cubes, or flat files. There are a number of issues to consider during data integration. Schema integration can be tricky. How can like rea l world entities from multiple data sources be 'matched up'? This is referred to as the entity identification problem. For example, how can

the data analyst or the computer be sure that customer id in one database, and cust_number in another refer to the same entity? Databases and data warehouses typically have metadata - that is, data about the data. Such metadata can be used to help avoid errors in schema integration. Redundancy is another important issue. An attribute may be redundant if it can be \derived" from another table, such as annual revenue. Inconsistencies in attribute or dimension naming can also cause redundancies in the resulting data set.

### Data Trans formation

In data transformation, the data are transformed or consolidated into forms appropriate for mining. Data transformation can involve the following:

1. Normalization, where the attribute data are scaled so as to fall within a small specified
range, such as -1.0 to 1.0, or 0 to 1.0.
2. Smoothing works to remove the noise from data. Such techniques include binning, clustering, and regression.
3. Aggregation, where summary or aggregation operations are applied to the data. For example, the daily sales data may be aggregated so as to compute monthly and annual total amounts. This step is typically used in constructing a data cube for analysis of the data at multiple granularities.
4. Generalization of the data, where low level or 'primitive' (raw) data are replaced by higher level concepts through the use of concept hierarchies. For example, categorical attributes, like street, can be generalized to higher level concepts, like city or county. Similarly, values for numeric attributes, like age, may be mapped to higher level concepts, like young, middle-aged, and senior.

### Data Reduction

Complex data analysis and mining on huge amounts of data may take a very long time, making such analysis impractical or infeasible. Data reduction techniques have been helpful in analyzing reduced representation of the dataset without compromising the integrity of the original data and yet producing the quality knowledge. The concept of data reduction is commonly understood as either reducing the volume or reducing the dimensions (number of attributes). There are

a number of methods that have facilitated in analyzing a reduced volume or dimension of data and yet yield useful knowledge. Certain partition based methods work on partition of data tuples.That is, mining on the reduced data set should be more efficient yet produce the same (or almost the same) analytical results Strategies for data reduction include the following.

1. Data cube aggregation, where aggregation operations are applied to the data in the construction of a data cube.
2. Dimension reduction, where irrelevant, weakly relevant, or redundant attributes or dimensions may be detected and removed.
3. Data compression, where encoding mechanisms are used to reduce the data set size. The methods used for data compression are wavelet transform and Principal Component Analysis.
4. Numerosity reduction, where the data are replaced or estimated by alternative, smaller data representations such as parametric models (which need store only the model parameters instead of the actual data e.g. regression and log-linear models), or nonparametric methods such as clustering, sampling, and the use of histograms.
5. Discretization and concept hierarchy generation, where raw data values for attributes are replaced by ranges or higher conceptual levels. Concept hierarchies allow the mining of data at multiple levels of abstraction, and are a powerful tool for data mining.

| Program No. | Marks for Execution (7) | | | Marks for Viva voce (3) | | TOTAL (10) | Signature of the Faculty |
|---|---|---|---|---|---|---|---|
| | Rubrics | | | Rubrics | | | |
| | Understanding of problem (2) | Execution (3) | Results and Documentation (2) | Conceptual Understanding and Communication of Concepts (2) | Use of appropriate Design Techniques (1) | | |
| 2 | | | | | | | |

## Program No. 2

Paste your DATA SHEET here

In [159]:
```
# Data Reduction: choosing required columns:
colms = ['budget','genres','original_language','title','overview','release_date' df
= df[colms]
```

In [161]:
```
df.describe()
```

Out[161]:

| | budget | runtime | vote_average | profit |
|---|---|---|---|---|
| **count** | 4.803000e+03 | 4801.000000 | 4803.000000 | 4.803000e+03 |
| **mean** | 2.904504e+07 | 106.875859 | 6.092172 | 5.321560e+07 |
| **std** | 4.072239e+07 | 22.611935 | 1.194612 | 1.359677e+08 |
| **min** | 0.000000e+00 | 0.000000 | 0.000000 | -1.657101e+08 |
| **25%** | 7.900000e+05 | 94.000000 | 5.600000 | -7.995375e+05 |
| **50%** | 1.500000e+07 | 103.000000 | 6.200000 | 2.511317e+06 |
| **75%** | 4.000000e+07 | 118.000000 | 6.800000 | 5.531286e+07 |
| **max** | 3.800000e+08 | 338.000000 | 10.000000 | 2.550965e+09 |

# Data Cleaning: - handling missing data:

1. runtime has 2 missing values
2. overview has 3 missing values
3. release date has one missing value

In [162]:
```
df.isnull().sum()
```

Out[162]:
```
budget              0
genres              0
original_language   0
title               0
overview            3
release_date        1
runtime             2
status              0
vote_average        0
profit              0
dtype: int64
```

In [164]:
```
# filled missing value with mean
df["runtime"] = df["runtime"].fillna(df["runtime"].mean())
```

In [168]:
```
#overview and release_date have 3 and 1 missing values respectively, lets fill t
df["overview"] = df["overview"].fillna('NaN')
df["release_date"] = df["release_date"].fillna('NaN')
```

```
#check if any other missing values present
df.isnull().sum()
```

Out[168]:
```
budget              0
genres              0
original_language   0
title               0
overview            0
release_date        0
runtime             0
status              0
vote_average        0
profit              0
dtype: int64
```

In [169]:
```
#binning vote_average by adding a new column called Status
df['status'] = np.where(df['vote_average']>=6,'HIT','FLOP')
```

In [171]:
```
#normalizing budget (absolute maximum scaling)
df["revenue_scaled"] = df['budget']/df['budget'].abs().max()
```

In [173]:
```
df.drop(['budget'], axis=1, inplace=True)
df.head()
```

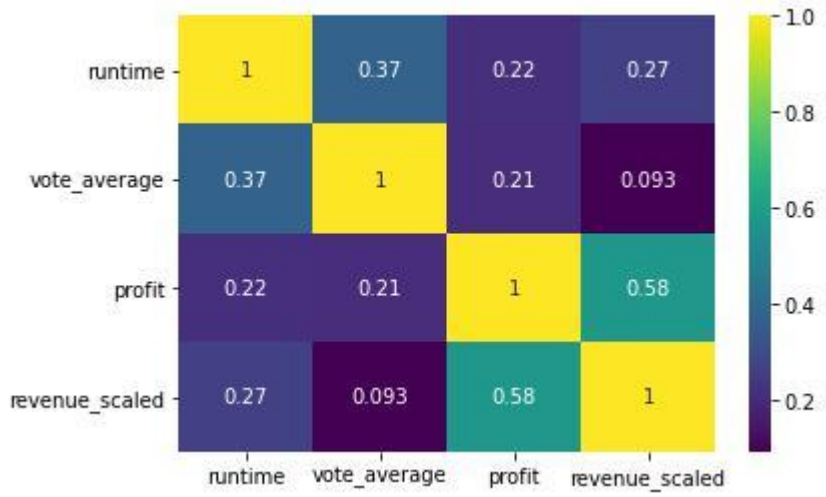
Out[173]:

| | genres | original_language | title | overview | release_date | runtime | status | vote_averag |
|---|---|---|---|---|---|---|---|---|
| **0** | [{"id": 28, "name": "Action"}, {"id": 12, "nam... | en | Avatar | In the 22nd century, a paraplegic Marine is di... | 2009-12-10 | 162.0 | HIT | 7 |
| **1** | [{"id": 12, "name": "Adventure"}, {"id": 14, "... | en | Pirates of the Caribbean: At World's End | Captain Barbossa, long believed to be dead, ha... | 2007-05-19 | 169.0 | HIT | 6 |
| **2** | [{"id": 28, "name": "Action"}, {"id": 12, "nam... | en | Spectre | A cryptic message from Bond's past sends him o... | 2015-10-26 | 148.0 | HIT | 6 |
| **3** | [{"id": 28, "name": "Action"}, {"id": 80, "nam... | en | The Dark Knight Rises | Following the death of District Attorney Harve... | 2012-07-16 | 165.0 | HIT | 7 |
| **4** | [{"id": 28, "name": "Action"}, {"id": 12, "nam... | en | John Carter | John Carter is a war-weary, former military ca... | 2012-03-07 | 132.0 | HIT | 6 |

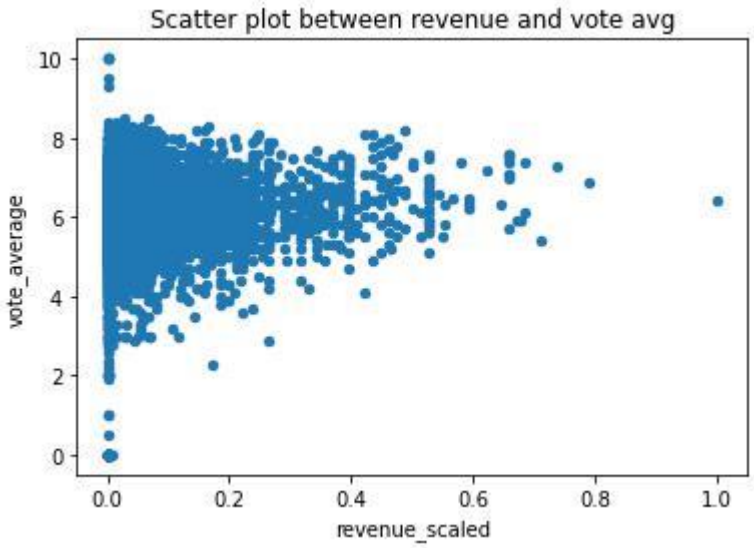
In [174]:
```
import matplotlib.pyplot as plt
import seaborn as sn
corr = df.corr()
print(corr)
```

```
                runtime  vote_average    profit  revenue_scaled
runtime        1.000000      0.373989  0.219919        0.269834
vote_average   0.373989      1.000000  0.208241        0.093146
profit         0.219919      0.208241  1.000000        0.575852
revenue_scaled 0.269834      0.093146  0.575852        1.000000
```

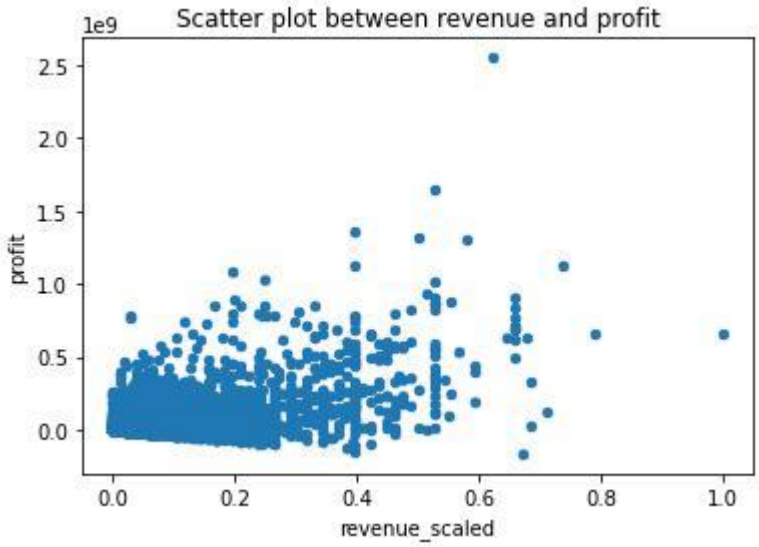
In [175]:
```
sn.heatmap(corr, annot=True,cmap='viridis')
plt.show()
```




In [181]:
```
#scatter plot 1
import matplotlib.pyplot as plot

df.plot.scatter(x='revenue_scaled', y='vote_average', title= "Scatter plot betwe

plot.show(block=True);
```



In [182]:
```
#scatter plot 2
import matplotlib.pyplot as plot


df.plot.scatter(x='revenue_scaled', y='profit', title= "Scatter plot between rev

plot.show(block=True);
```



## Program No. 3

**Title:**

**Linear Regression**

**Objective:**

**To write a program to classify the tuples using linear regression.**

**Reference:**

**Data Mining Introductory & Advanced Topic by Margaret H. Dunham Data Mining Concept and Technique By Han & Kamber Pre-requisite:**

**Knowledge of regression techniques**

**Theory:**

**Regression problem deals with estimation of output values based on input values. In the method we estimate the formula of straight line, which partitions data into 2 classes, by defining the regression coefficient C, the relation between output parameter Y and input parameter X1, X2, X3 ….. Xn can be estimated Input : Training data set**

| Name | Gender | Height |
|---|---|---|
| Christina | F | 1.6m |
| Jim | M | 1.9m |
| Maggie | F | 1.9m |
| Martha | F | 1.88m |
| Stephony | F | 1.7m |
| Bob | M | 1.85m |
| Dave | M | 1.7m |
| Steven | M | 2.1m |
| Amey | F | 1.8m |

**Output**

**The tuple is being classified using linear regression technique. Having value > 0.5 is classified as medium else < 0.5 then tuple is classified as short.**

| Program No. | Marks for Execution (7) | | | Marks for Viva voce (3) | | TOTAL (10) | Signature of the Faculty |
|---|---|---|---|---|---|---|---|
| | Rubrics | | | Rubrics | | | |
| | Understanding of problem (2) | Execution (3) | Results and Documentation (2) | Conceptual Understanding and Communication of Concepts (2) | Use of appropriate Design Techniques (1) | | |
| 3 | | | | | | | |

## Program No. 3

**Paste your DATA SHEET here**

In [37]:
```
colms = ['Precip','MinTemp','MeanTemp','Snowfall','MaxTemp']
df = df[colms]
df.head()
```

Out[37]:

| | Precip | MinTemp | MeanTemp | Snowfall | MaxTemp |
|---|---|---|---|---|---|
| **0** | 1.016 | 22.222222 | 23.888889 | 0 | 25.555556 |
| **1** | 0 | 21.666667 | 25.555556 | 0 | 28.888889 |
| **2** | 2.54 | 22.222222 | 24.444444 | 0 | 26.111111 |
| **3** | 2.54 | 22.222222 | 24.444444 | 0 | 26.666667 |
| **4** | 0 | 21.666667 | 24.444444 | 0 | 26.666667 |

In [38]:
```
df.dtypes
```

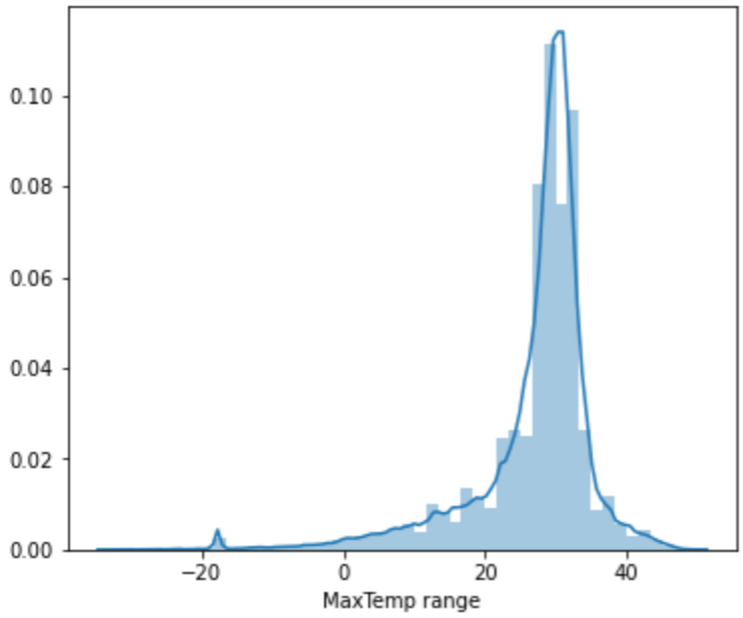
Out[38]:
```
Precip        object
MinTemp      float64
MeanTemp     float64
Snowfall      object
MaxTemp      float64
dtype: object
```

In [59]:
```
df.shape,df.isnull().sum()
```

Out[59]:
```
((119040, 5), 5)
```

In [41]:
```
plt.figure(figsize=(6,5))
pl = sns.distplot(df['MaxTemp'])
pl.set(xlabel = "MaxTemp range")
```

Out[41]:
```
[Text(0.5, 0, 'MaxTemp range')]
```



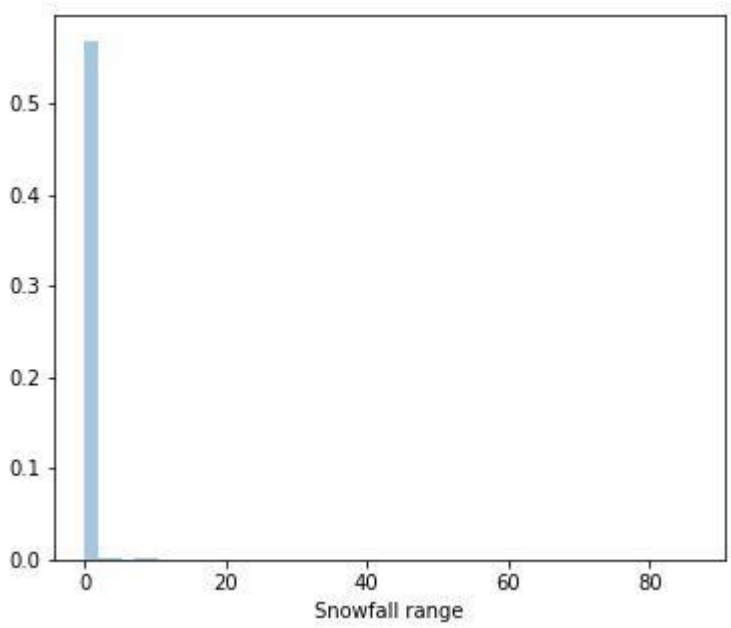

In [42]:
```
plt.figure(figsize=(6,5))
pl = sns.distplot(df['Precip'])
pl.set(xlabel = "Precipitation range")
```

Out[42]:
```
[Text(0.5, 0, 'Precipitation range')]
```

In [43]:
```
plt.figure(figsize=(6,5))
pl = sns.distplot(df['Snowfall'])
pl.set(xlabel = "Snowfall range")
```

```
C:\Users\SVM\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarn
ing: Default bandwidth for data is 0; skipping density estimation.
  warnings.warn(msg, UserWarning)
```

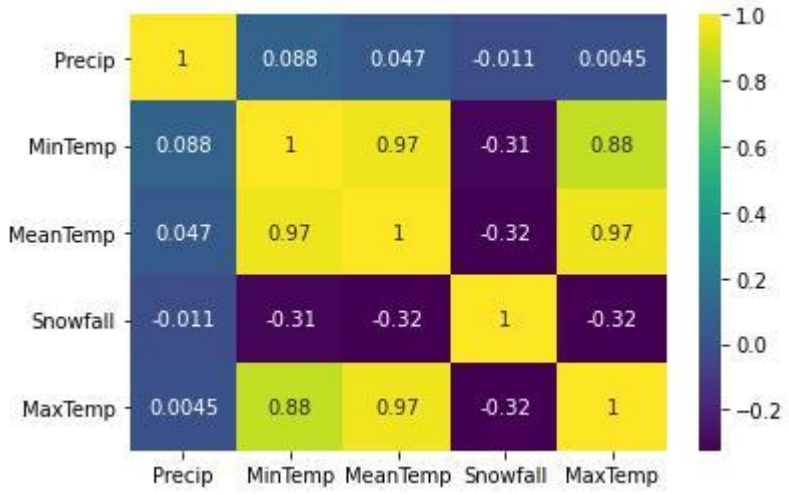
Out[43]:
```
[Text(0.5, 0, 'Snowfall range')]
```




In [45]:
```
import matplotlib.pyplot as plt
import seaborn as sn
corr = df.corr()
print(corr)
```

```
              Precip   MinTemp  MeanTemp  Snowfall   MaxTemp
Precip      1.000000  0.088455  0.047061 -0.011043  0.004457
MinTemp     0.088455  1.000000  0.965425 -0.307854  0.878384
MeanTemp    0.047061  0.965425  1.000000 -0.323671  0.969048
Snowfall   -0.011043 -0.307854 -0.323671  1.000000 -0.322013
```

In [46]:
```
sn.heatmap(corr, annot=True,cmap='viridis')
plt.show()
```

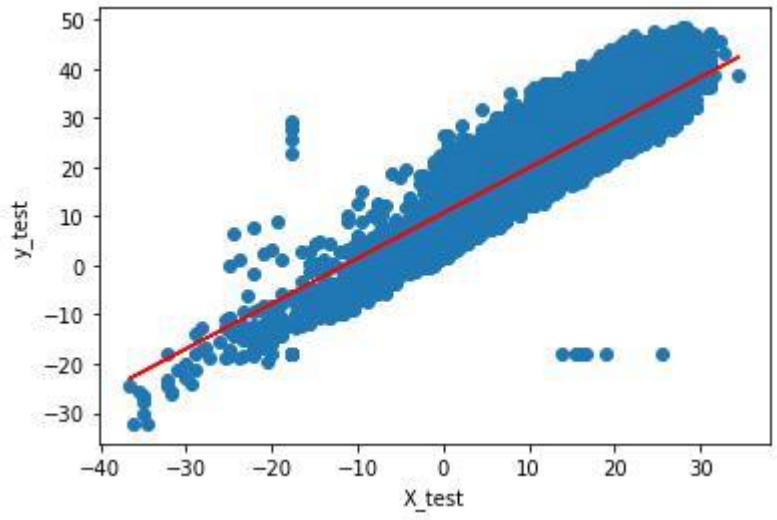


# Simple linear regression

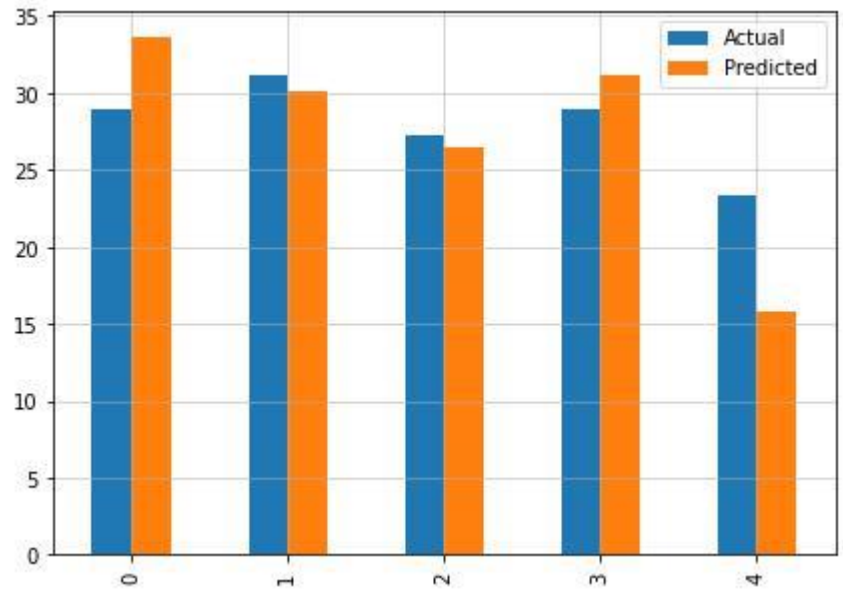
In [51]:
```
print(regressor.coef_)
print(regressor.intercept_)
```

```
[[0.92033997]]
[10.66185201]
```

In [52]:
```
plt.scatter(X_test,y_test)
plt.xlabel('X_test')
plt.ylabel('y_test')
plt.plot(X_test,y_pred,color='red')
plt.show()
```



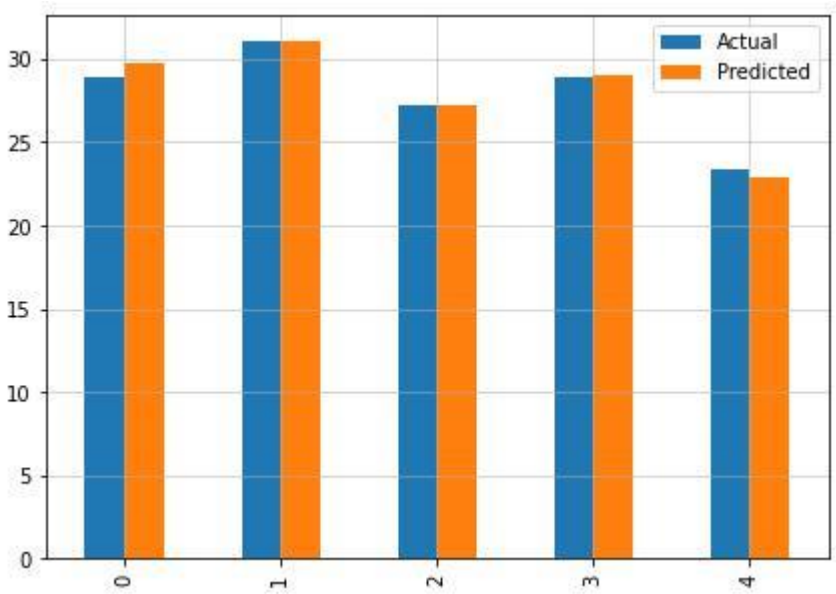

In [54]:
```
df_new=pd.DataFrame({'Actual':y_test.flatten(),'Predicted':y_pred.flatten()})
df1=df_new.head(5)
df1.plot(kind='bar',figsize=(7,5))

plt.grid(which='major',linewidth='0.5')
plt.grid(which='minor',linewidth='0.5')
plt.show()
```



# Multiple Linear Regression

In [55]:
```
df.columns
```

Out[55]:
```
Index(['Precip', 'MinTemp', 'MeanTemp', 'Snowfall', 'MaxTemp'], dtype='object')
```



## Program No. 4

**Title:**

**Implement classification using K nearest neighbor classification**

**Objective:**

**To learn how to classify data by K nearest neighbor algorithm for classification**

**Reference:**

**Data Mining Introductory & Advanced Topic by Margaret H.**

**Dunham Data Mining Concept and Technique By Han & Kamber**

**Theory:**

**In k-nearest-neighbor classification, the training dataset is used to classify each member of a "target" dataset.**

**The structure of the data is that there is a classification (categorical) variable of interest ("buyer," or "non-buyer," for example), and a number of additional predictor variables (age, income, location so on.)**

**Algorithm:**

**For each row (case) in the target dataset (the set to be classified), locate the k closest members (the k nearest neighbors) of the training dataset.**

**A Euclidean Distance measure is used to calculate how close each member of the training set is to the target row that is being examined.**

**Examine the k nearest neighbors - which classification (category) do most of them belong to? Assign this category to the row being examined.**

**Repeat this procedure for the remaining rows (cases) in the target set.**

**This algorithm lets the user select a maximum value for k, builds models parallelly on all values of k upto the maximum specified value and scoring is done on the best of these models.**

**The computing time goes up as k goes up, but the advantage is that higher values of k provide smoothing that reduces vulnerability to noise in the training data.**

**In practical applications, typically, k is in units or tens rather than in hundreds or thousands.**

| Name | Gender | Height(m) |
|---|---|---|
| Kristina | F | 1.6 |
| Jim | M | 2 |
| Maggie | F | 1.9 |
| Bob | M | 1.85 |
| Dave | F | 1.7 |
| Kimm | M | 1.9 |
| Todd | M | 1.9 |
| Amy | F | 1.85 |
| Kathy | F | 1.6 |

**2m<=Tall, 1.7m< H<2m Medium, H<=1.7m Short**

**OutPut:**

**New Tuple <Pat,F,1.6> , suppose K=5 is given than K nearest neighbors to input tuple**

**{(Kristina,F,1.6) , (Kathy,F,1.6),(Dave,F,1.7)}**

| Program No. | Marks for Execution (7) | | | Marks for Viva voce (3) | | TOTAL (10) | Signature of the Faculty |
|---|---|---|---|---|---|---|---|
| | Rubrics | | | Rubrics | | | |
| | Understanding of problem (2) | Execution (3) | Results and Documentation (2) | Conceptual Understanding and Communication of Concepts (2) | Use of appropriate Design Techniques (1) | | |
| 4 | | | | | | | |

## Program No. 4

**Paste your DATA SHEET here**

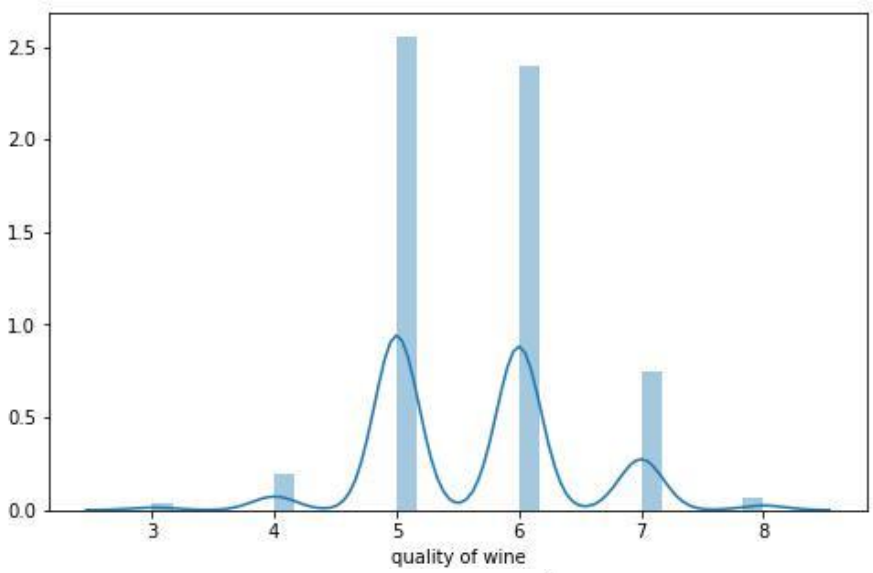
In [2]:
```
file = './datasets/P4_winequality.csv'
df = pd.read_csv(file)

df.head()
```

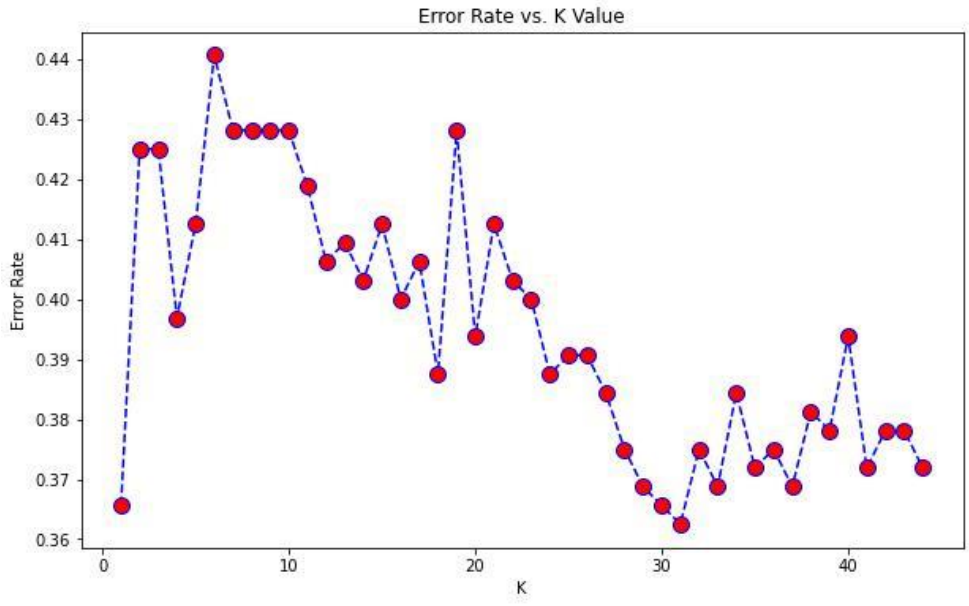
Out[2]:

| | fixed acidity | volatile acidity | citric acid | residual sugar | chlorides | free sulfur dioxide | total sulfur dioxide | density | pH | sulphates | alcohol |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 7.4 | 0.70 | 0.00 | 1.9 | 0.076 | 11.0 | 34.0 | 0.9978 | 3.51 | 0.56 | 9.4 |
| **1** | 7.8 | 0.88 | 0.00 | 2.6 | 0.098 | 25.0 | 67.0 | 0.9968 | 3.20 | 0.68 | 9.8 |
| **2** | 7.8 | 0.76 | 0.04 | 2.3 | 0.092 | 15.0 | 54.0 | 0.9970 | 3.26 | 0.65 | 9.8 |
| **3** | 11.2 | 0.28 | 0.56 | 1.9 | 0.075 | 17.0 | 60.0 | 0.9980 | 3.16 | 0.58 | 9.8 |
| **4** | 7.4 | 0.70 | 0.00 | 1.9 | 0.076 | 11.0 | 34.0 | 0.9978 | 3.51 | 0.56 | 9.4 |

In [11]:
```
plt.figure(figsize=(8,5))
pl = sns.distplot(df['quality'])
pl.set(xlabel = "quality of wine")
```

Out[11]:
```
[Text(0.5, 0, 'quality of wine')]
```



## Program No. 5

**Title:**

**Implement decision tree based algorithm for classification**

**Objective:**

**To learn decision tree based algorithm for classification**

**Reference:**

**Data Mining Introductory & Advanced Topic by Margaret H.**

**Dunham Data Mining Concept and Technique By Han & Kamber**

**Theory:**

**Decision tree learning, used in data mining and machine learning, uses a decision tree as a predictive model which maps observations about an item to conclusions about the item's target value. More descriptive names for such tree models are classification trees or regression trees. In these tree structures, leaves represent classifications and branches represent conjunctions of features that lead to those classifications.**

**In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data but not decisions; rather the resulting classification tree can be an input for decision making.**

**Basic steps in building tree**

**Applying the tree to database**

**Internal node-test on attribute Branch-outcome of test**

**Leaf node-class Topmost-root node**

**Algorithm**

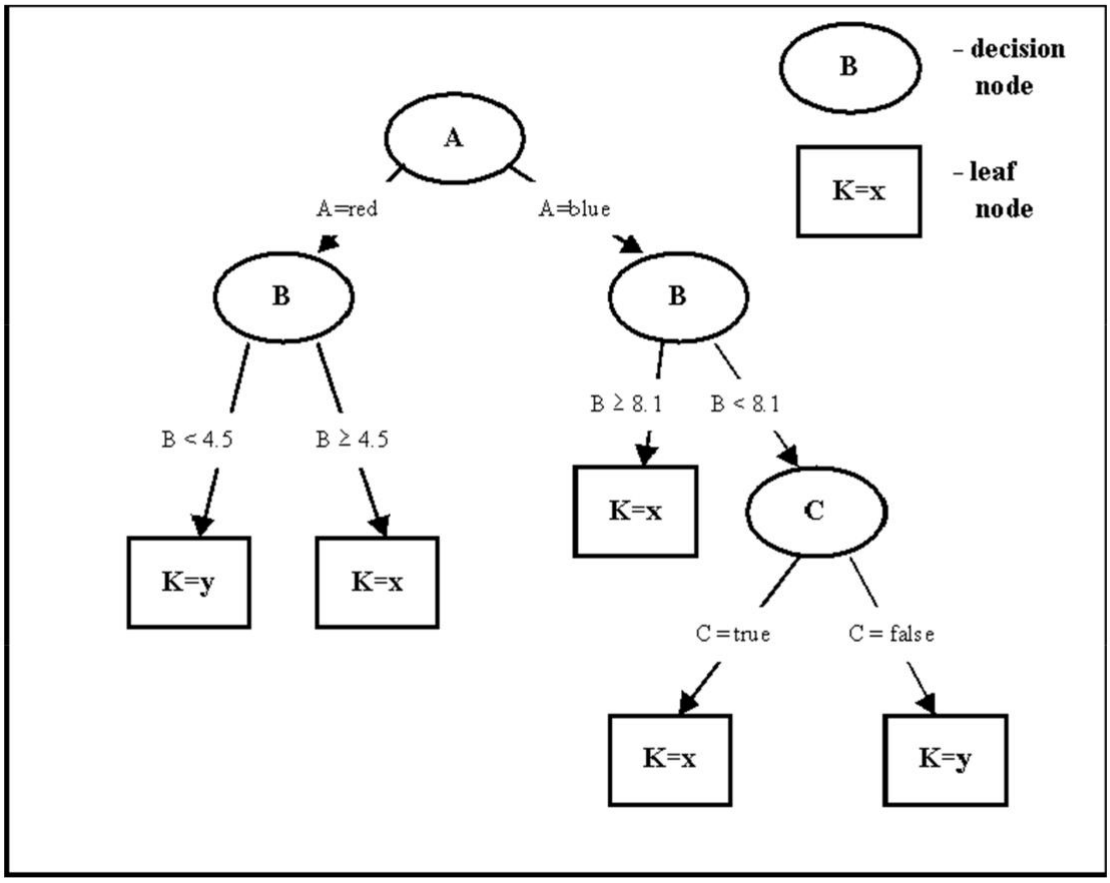
**1.compute the entropy for data-set**

**2.for every attribute/feature:**

- **calculate entropy for all categorical values**
- **take average information entropy for the current**
- **attribute calculate gain for the current attribute**

**3. pick the highest gain attribute.**

**4. Repeat until the desired tree is constructed .**



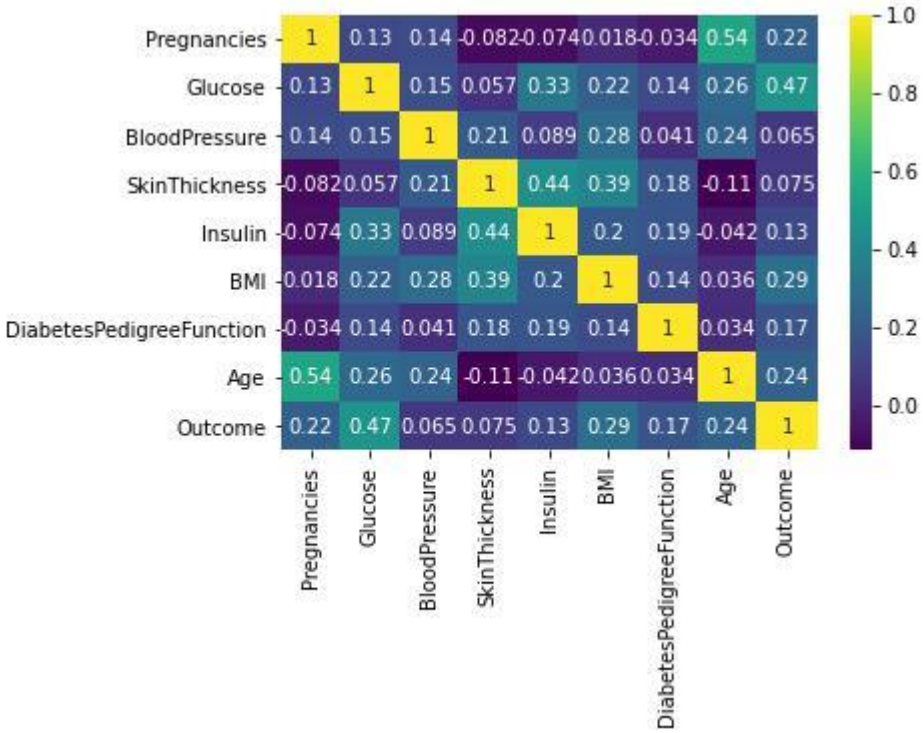

**Final Decision Tree**

| Program No. | Marks for Execution (7) | | | Marks for Viva voce (3) | | TOTAL (10) | Signature of the Faculty |
|---|---|---|---|---|---|---|---|
| | Rubrics | | | Rubrics | | | |
| | Understanding of problem (2) | Execution (3) | Results and Documentation (2) | Conceptual Understanding and Communication of Concepts (2) | Use of appropriate Design Techniques (1) | | |
| 5 | | | | | | | |

## Program No. 5

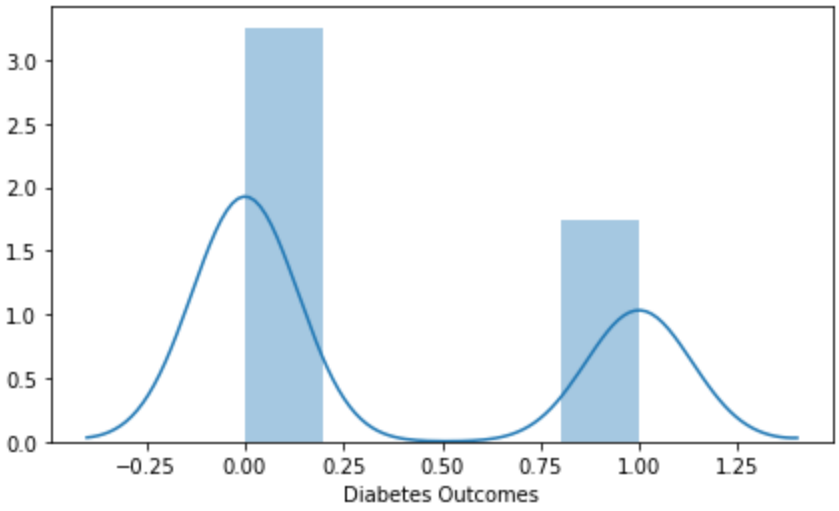
**Paste your DATA SHEET here**

In [6]:
```
sn.heatmap(corr, annot=True,cmap='viridis')
plt.show()
```

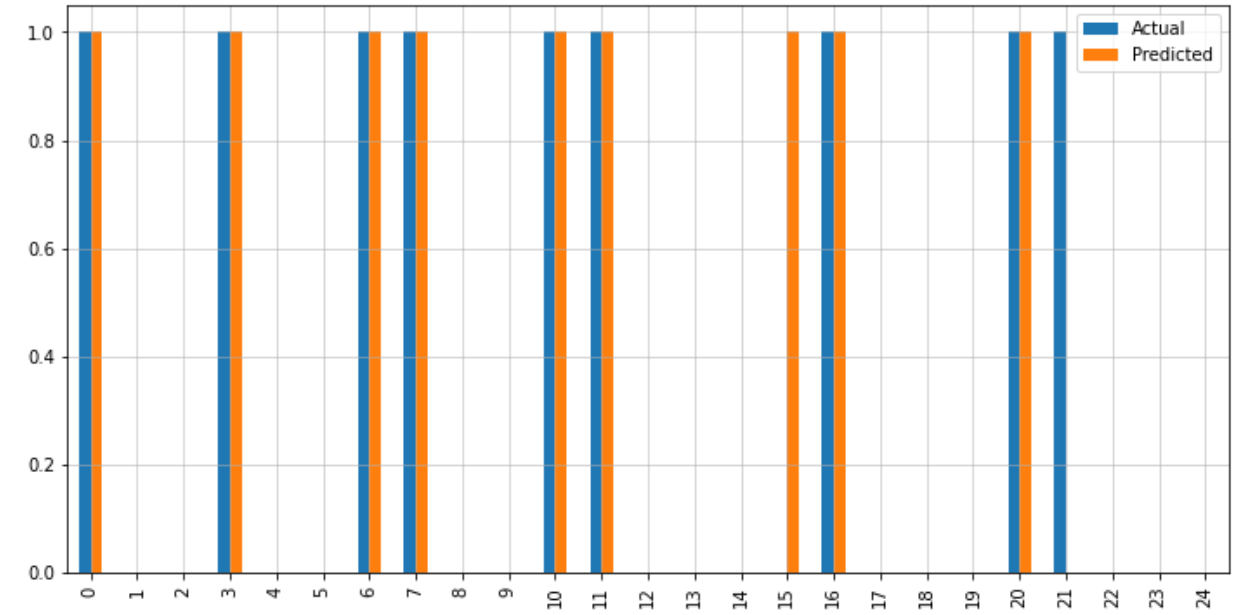


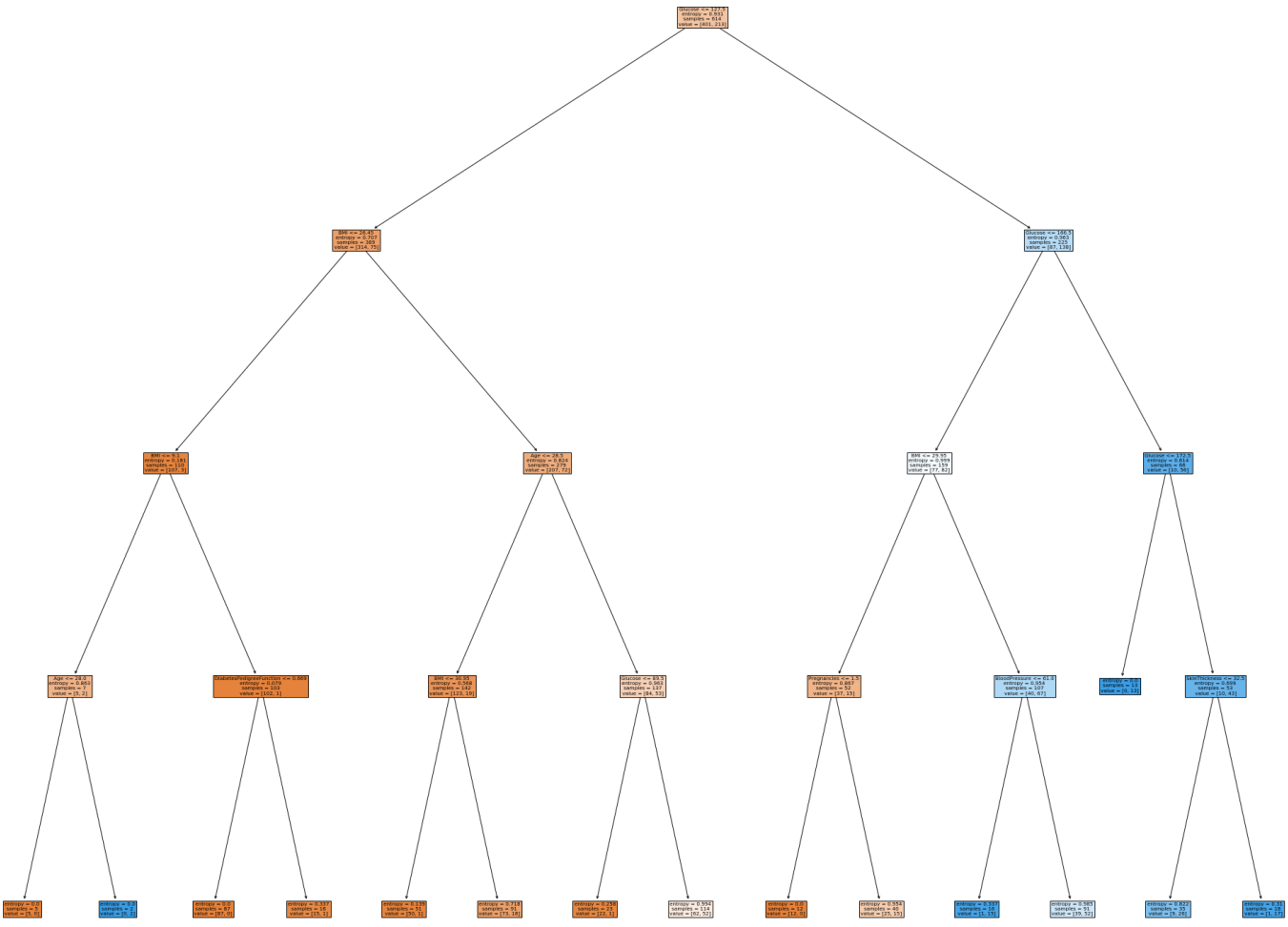

In [15]:
```
plt.figure(figsize=(7,4))
pl = sn.distplot(df.Outcome)
pl.set(xlabel = "Diabetes Outcomes")
```

Out[15]:
```
[Text(0.5, 0, 'Diabetes Outcomes')]
```




In [21]:
```
df_new=pd.DataFrame({'Actual':y_test.flatten(),'Predicted':y_pred.flatten()})
df1=df_new.head(25)
df1.plot(kind='bar',figsize=(12,6))

plt.grid(linewidth='0.5')
plt.grid(linewidth='0.5')
plt.show()
```


Actual
Predicted
1.0
0.8
0.6
0.4
0.2
0.0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24



## Program No. 6

**Title:**

**Navie Bayesian Classification**

**Objective:**

**To implement classification using Bayes theorm.**

**Reference:**

**Data Mining Introductory & Advanced Topic by Margaret H.**

**Dunham Data Mining Concept and Technique By Han & Kamber**

**Theory:**

**The simple baysian classification assumes that the effect of an attribute value of a given class membership is independent of other attribute. The Bayes theorem is as follows –**

**Let X be an unknown sample. Let it be hypothesis such that X belongs to particular class C. We need to determine P(H/X).**

**The probability that hypothesis it holds is given that all values of X are observed. P(H/X)=(P(X/H).P(H))/P(X)**

**In this program, initially take the number of tuples in training data set in variable L. The string array's name, gender, height, output to store the details and output respectfully. Therefore, the tuple details are taken from user using 'for' loops. Bayesian classification has an expected classification. Now using the counter variables for various attributes i.e. (male/female) for gender and (short/medium/tall) for height. The tuples are scanned and the respective counter is incremented accordingly using if- else-if structure. Therefore variables pshort, pmed, plong are used to convert the counter variables to corresponding values.**

**Algorithm –**

**START**

**Store the training data set**

**Specify ranges for classifying the data**

**Calculate the probability of being tall, medium, short**

**Also, calculate the probabilities of tall, short, medium according to gender and**

**classification ranges**

**Calculate the likelihood of short, medium and tall**

**Calculate P(t) by summing up of probable**

**likelihood Calculate actual probabilities**

**Input :**

**Training data set**

| Name | Gender | Height | Output |
|---|---|---|---|
| Christina | F | 1.6m | Short |
| Jim | M | 1.9m | Tall |
| Maggie | F | 1.9m | Medium |
| Martha | F | 1.88m | Medium |
| Stephony | F | 1.7m | Medium |
| Bob | M | 1.85m | Short |
| Dave | M | 1.7m | Short |
| Steven | M | 2.1m | Tall |
| Amey | F | 1.8m | Medium |

**Output**

**The tuple belongs to the class having highest probability. Thus new tuple is classified.**

| Program No. | Marks for Execution (7) | | | Marks for Viva voce (3) | | TOTAL (10) | Signature of the Faculty |
|---|---|---|---|---|---|---|---|
| | Rubrics | | | Rubrics | | | |
| | Understanding of problem (2) | Execution (3) | Results and Documentation (2) | Conceptual Understanding and Communication of Concepts (2) | Use of appropriate Design Techniques (1) | | |
| 6 | | | | | | | |

## Program No. 6

**Paste your DATA SHEET here**

In [3]:

```
df.head()
```

Out[3]:

| | v1 | v2 | Unnamed: 2 | Unnamed: 3 | Unnamed: 4 |
|---|---|---|---|---|---|
| 0 | ham | Go until jurong point, crazy.. Available only ... | NaN | NaN | NaN |
| 1 | ham | Ok lar... Joking wif u oni... | NaN | NaN | NaN |
| 2 | spam | Free entry in 2 a wkly comp to win FA Cup fina... | NaN | NaN | NaN |
| 3 | ham | U dun say so early hor... U c already then say... | NaN | NaN | NaN |
| 4 | ham | Nah I don't think he goes to usf, he lives aro... | NaN | NaN | NaN |

In [9]:

```
df['v2'] = df['v2'].apply(text_process)
df['length'] = df['v2'].apply(len)
unique_words = []
for msg in df['v2']:
    for word in msg:
        unique_words.append(word)
unique_words = set(unique_words)
df.head()
```

Out[9]:

| | v1 | v2 | length |
|---|---|---|---|
| 0 | ham | [Go, jurong, point, crazy, Available, bugis, n... | 9 |
| 1 | ham | [Ok, lar, Joking, wif, u, oni, 11304, 11304, 1... | 9 |
| 2 | spam | [Free, entry, 2, wkly, comp, win, FA, Cup, final] | 9 |
| 3 | ham | [U, dun, say, early, hor, U, c, already, say] | 9 |
| 4 | ham | [Nah, dont, think, goes, usf, lives, around, t... | 9 |

```
print("accuracy: ",accuracy_score(label_test,y_pred))
print(classification_report(label_test,y_pred))
print(confusion_matrix(label_test,y_pred))
```

```
accuracy:  0.8493273542600897
              precision    recall  f1-score   support

         ham       0.85      1.00      0.92       947
        spam       0.00      0.00      0.00       168

    accuracy                           0.85      1115
   macro avg       0.42      0.50      0.46      1115
weighted avg       0.72      0.85      0.78      1115

[[947   0]
 [168   0]]
```

## Program No. 7

**Title:**

**Implementation of Support Vector Machine**

**Objective:**

**To understand the dynamics of SVM**

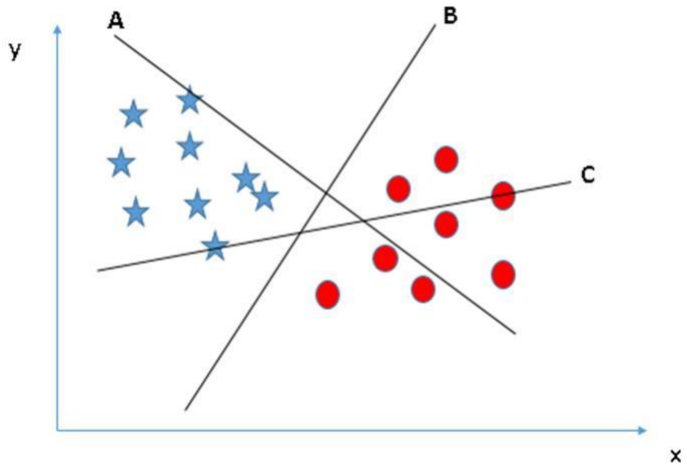
**Reference:**

**Data Mining Concept and Technique By Han & Kamber**

**Theory:**

**"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well .**

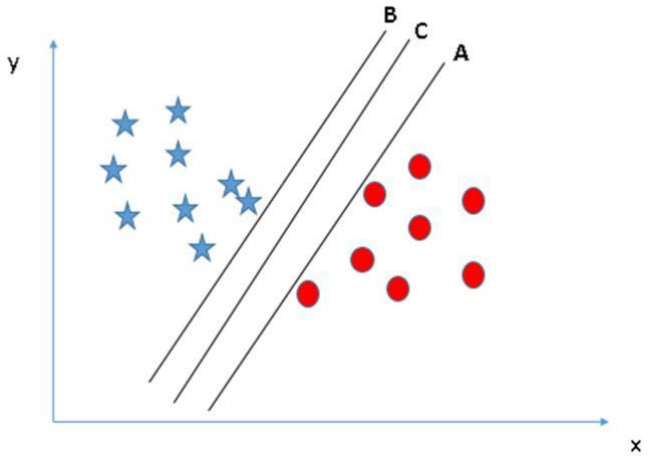
**Let's understand: Identify the right hyper-plane (Scenario-1): Consider three hyper-planes (A, B and C). Now, identify the right hyper-plane to classify star and circle.**



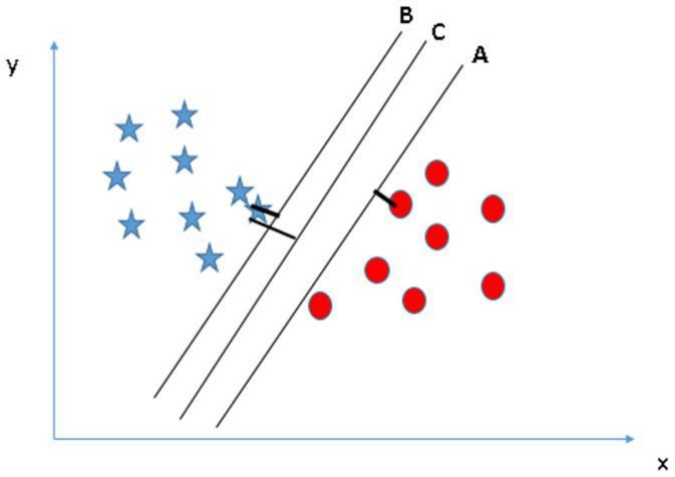

**You need to remember a thumb rule to identify the right hyper-plane: "Select the hyper-plane which segregates the two classes better". In this scenario, hyper-plane "B" has excellently performed this job.**

**Identify the right hyper-plane (Scenario-2): Here, we have three hyper-planes (A, B and C) and all are segregating the classes well. Now, How can we identify the right**

**hyper-plane?**

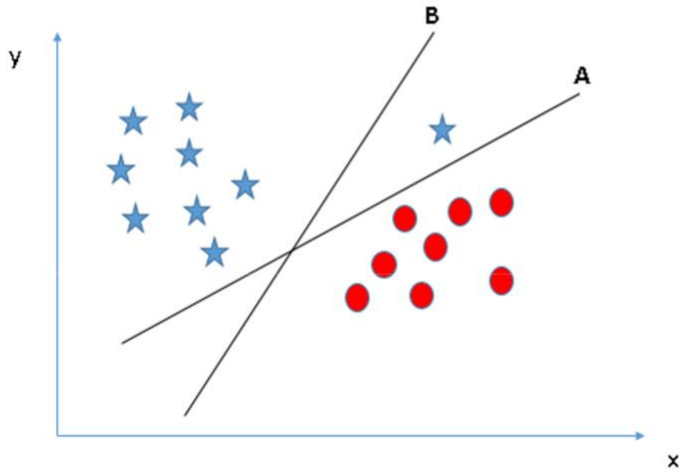



**Here, maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as Margin. Let's look at the below snapshot:**



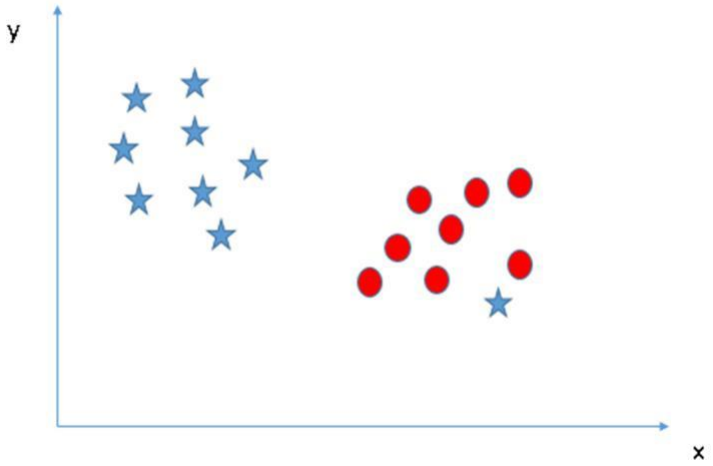

**Above, you can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C. Another lightning reason for selecting the hyper-plane with higher margin is robustness. If we select a hyper-plane having low margin then there is high chance of miss-classification.**

**Identify the right hyper-plane (Scenario-3):Hint: Use the rules as discussed in previous section to identify the right hyper-plane**



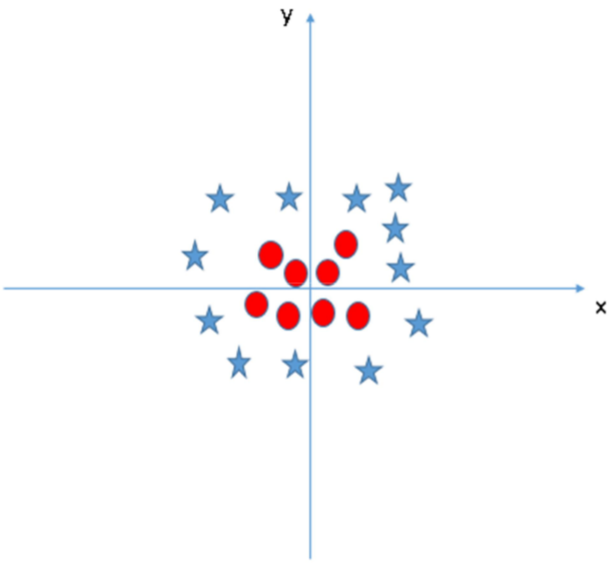

**Some of you may have selected the hyper-plane B as it has higher margin compared to A. But, here is the catch, SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin. Here, hyper-plane B has a classification error and A has classified all correctly. Therefore, the right hyper-plane is A.**

**Can we classify two classes (Scenario-4)?: Below, I am unable to segregate the two classes using a straight line, as one of star lies in the territory of other(circle) class as an outlier.**

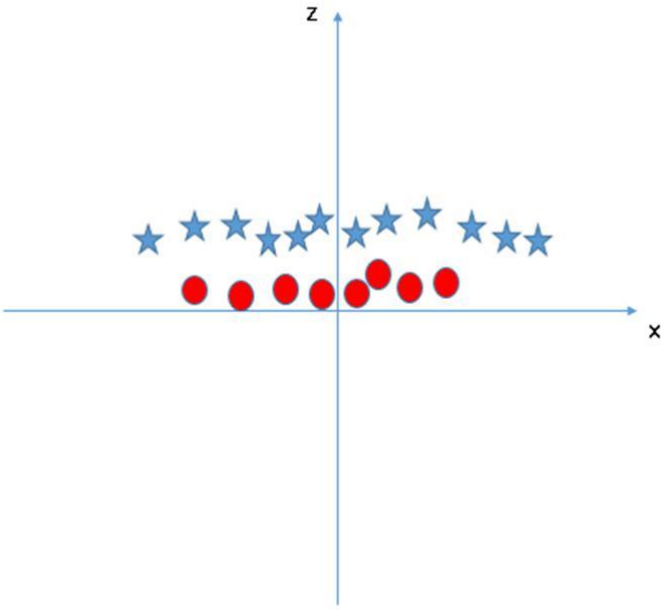



**As I have already mentioned, one star at other end is like an outlier for star class. SVM has a feature to ignore outliers and find the hyper-plane that has maximum margin. Hence, we can say, SVM is robust to outliers.**

**Find the hyper-plane to segregate to classes (Scenario-5): In the scenario below, we can't have linear hyper-plane between the two classes, so how does SVM classify these two classes? Till now, we have only looked at the linear hyper-plane.**




**SVM can solve this problem. Easily! It solves this problem by introducing additional feature. Here, we will add a new feature z=x^2+y^2. Now, let's plot the data points on axis x and z:**



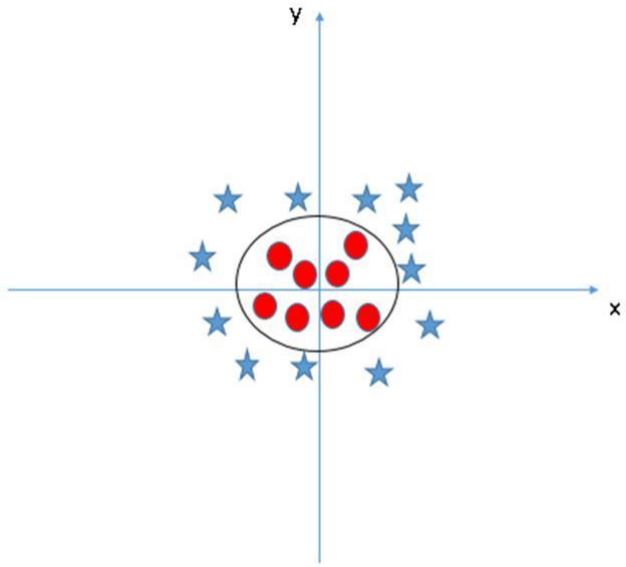

**In above plot, points to consider are:**

**All values for z would be positive always because z is the squared sum of both x and y**

**In the original plot, red circles appear close to the origin of x and y axes, leading to lower value of z and star relatively away from the origin result to higher value of z. In SVM, it is easy to have a linear hyper-plane between these two classes. But, another burning question which arises is, should we need to add this feature manually to have a hyper-plane. No, SVM has a technique called the kernel trick. These are**

functions which takes low dimensional input space and transform it to a higher dimensional space i.e. it converts not separable problem to separable problem, these functions are called kernels. It is mostly useful in non-linear separation problem. Simply put, it does some extremely complex data transformations, then find out the process to separate the data based on the labels or outputs you've defined.

**When we look at the hyper-plane in original input space it looks like a circle:**




| Program No. | Marks for Execution (7) | | | Marks for Viva voce (3) | | TOTAL (10) | Signature of the Faculty |
|---|---|---|---|---|---|---|---|
| | Rubrics | | | Rubrics | | | |
| | Understanding of problem (2) | Execution (3) | Results and Documentation (2) | Conceptual Understanding and Communication of Concepts (2) | Use of appropriate Design Techniques (1) | | |
| 7 | | | | | | | |

## Program No. 7

**Paste your DATA SHEET here**

```
df = pd.read_csv(path,encoding = 'latin-1')
```

In [3]:

```
df.head()
```

Out[3]:

| | v1 | v2 | Unnamed: 2 | Unnamed: 3 | Unnamed: 4 |
|---|---|---|---|---|---|
| 0 | ham | Go until jurong point, crazy.. Available only ... | NaN | NaN | NaN |
| 1 | ham | Ok lar... Joking wif u oni... | NaN | NaN | NaN |
| 2 | spam | Free entry in 2 a wkly comp to win FA Cup fina... | NaN | NaN | NaN |
| 3 | ham | U dun say so early hor... U c already then say... | NaN | NaN | NaN |
| 4 | ham | Nah I don't think he goes to usf, he lives aro... | NaN | NaN | NaN |

In [18]:

```
print("accuracy: ",accuracy_score(label_test,y_pred))
print(classification_report(label_test,y_pred))
print(confusion_matrix(label_test,y_pred))
```

```
accuracy:  0.8690582959641255
              precision    recall  f1-score   support

         ham       0.87      1.00      0.93       969
        spam       0.00      0.00      0.00       146

    accuracy                           0.87      1115
   macro avg       0.43      0.50      0.46      1115
weighted avg       0.76      0.87      0.81      1115

[[969   0]
 [146   0]]
```

## Program No. 8

**Title:**

**Implement Apriori algorithm for association rule**

**Objective:**

**To learn association rule for Apriori algorithm**

**Reference:**

**Data Mining Introductory & Advanced Topic by Margaret H.**

**Dunham Data Mining Concept and Technique By Han & Kamber**

**Theory:**

**Association rule mining is defined as:**

**Let I= { …} be a set of 'n' binary attributes called items. Let D= { ….} be set of transaction called database. Each transaction in D has a unique transaction ID and contains a subset of the items in I. a rule is defined as implication of the form X→Y where X, Y⊆ I and X∩Y=Φ. The set of items X and Y are called antecedent and consequent of the rule respectively.**

**Useful Terms**

**To select interesting rules from the set of all possible rules, constraints on various measures of significance and interest can be used. The best known constraints are minimum thresholds on support and confidence.**

**Support**

**The support supp(X) of an item set X can be defined as proportion of transactions in the data set which contain the item set.**

**Supp(X) = no. of transactions which contain the item set 'X' / total no. of transactions Confidence**

**The confidence of a rule is defined as:**

**Conf (X→Y) = supp (XUY)/supp(X)**

**Definition of Apriori Algorithm**

**The Apriori Algorithm is an influential algorithm for mining frequent item sets for Boolean association rules. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation, and groups of candidates are tested against the data. Apriori is designed to operate on**

**database containing transactions (for example, collections of items bought by customers, or details of a website frequentation).**

**Key Concept**

**Frequent item sets: All the sets which contain the item with the minimum support (denoted as for item set.**

**Apriori Property: Any subset of frequent item set must be frequent.**

**Join operation: To find, a set of candidate k-item sets is generated by joining with itself.**

**Apriori Algorithm Steps**

**Below are the apriori algorithm steps:**

**Scan the transaction data base to get the support 'S' each 1-itemset, compare 'S' with min_sup, and get a support of 1-itemsets,**

**Use join to generate a set of candidate k-item set. Use apriori property to prune the unfrequented k-item sets from this set.**

**Scan the transaction database to get the support 'S' of each candidate k-item set in the given set, compare 'S' with min_sup, and get a set of frequent k-item set**

**If the candidate set is NULL, for each frequent item set 1, generate all nonempty subsets of 1.**

**For every nonempty subsets of 1, output the rule "s=>(1-s)" if confidence C of the rule "s=>(1-s)" min_conf**

**If the candidate set is not NULL, go to step 2.**

**Example for Apriori Algorithms Market-Basket**

**Analysis is one of the examples for Apriori.**

**Provides insight into which products tend to be purchased together and which are most amenable to promotion.**

**Actionable rules**

**Trivial rules**

**People who buy chalk-piece also buy duster**

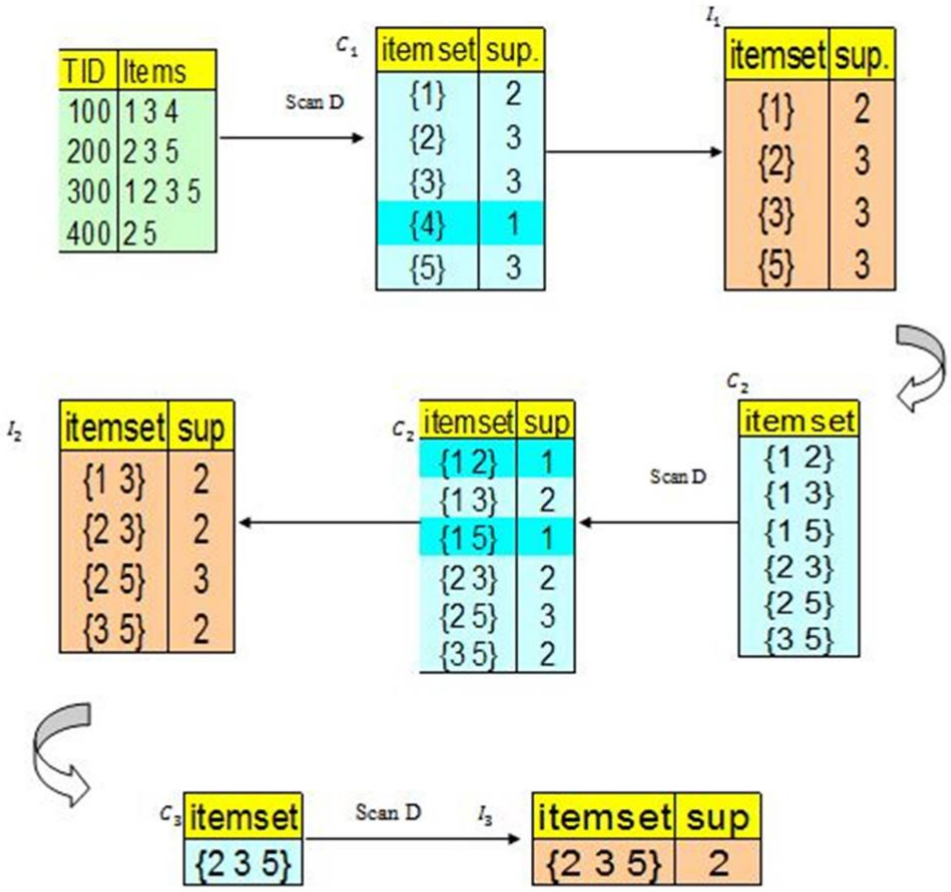
**Inexplicable**

**People who buy mobile also buy bag**

**Database D**

**Minsup = 0.5**

**Example of Apriori algorithm**




| Program No. | Marks for Execution (7) | | | Marks for Viva voce (3) | | TOTAL (10) | Signature of the Faculty |
|---|---|---|---|---|---|---|---|
| | Rubrics | | | Rubrics | | | |
| | Understanding of problem (2) | Execution (3) | Results and Documentation (2) | Conceptual Understanding and Communication of Concepts (2) | Use of appropriate Design Techniques (1) | | |
| 8 | | | | | | | |

**Program No. 8**

**Paste your DATA SHEET here**

```
df = pd.read_csv(path, header=None)
```

In [3]:

```
df.head()
```

Out[3]:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | shrimp | almonds | avocado | vegetables mix | green grapes | whole weat flour | yams | cottage cheese | energy drink | tomato juice | low fat yogurt | green tea | honey |
| 1 | burgers | meatballs | eggs | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| 2 | chutney | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| 3 | turkey | avocado | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| 4 | mineral | milk | energy | whole | green | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |

In [16]:

```
num_associations = 0
for item in association_results:
    for i in range(len(item.ordered_statistics)):
        print("Rule: " + str(list(item.ordered_statistics[i].items_base)) + " -
        print("Confidence: " + str(item[2][i][2]))
        print("Lift: " + str(item[2][i][3]))
        print("====================================")
        num_associations += 1
print(f"total number of association rules = {num_associations}")
```

```
Rule: ['light cream'] -> ['chicken']
Confidence: 0.29059829059829057
Lift: 4.84395061728395
====================================
Rule: ['mushroom cream sauce'] -> ['escalope']
Confidence: 0.3006993006993007
Lift: 3.790832696715049
====================================
Rule: ['pasta'] -> ['escalope']
Confidence: 0.3728813559322034
Lift: 4.700811850163794
====================================
Rule: ['herb & pepper'] -> ['ground beef']
Confidence: 0.3234501347708895
Lift: 3.2919938411349285
====================================
Rule: ['tomato sauce'] -> ['ground beef']
Confidence: 0.3773584905660377
Lift: 3.840659481324083
====================================
Rule: ['whole wheat pasta'] -> ['olive oil']
Confidence: 0.2714932126696833
Lift: 4.122410097642296
====================================
Rule: ['pasta'] -> ['shrimp']
```

```
Confidence: 0.3220338983050847
Lift: 4.506672147735896
====================================
Rule: ['frozen vegetables', 'chocolate'] -> ['shrimp']
Confidence: 0.23255813953488375
Lift: 3.2545123221103784
====================================
Rule: ['shrimp', 'chocolate'] -> ['frozen vegetables']
Confidence: 0.29629629629629634
Lift: 3.1084175084175087
====================================
Rule: ['ground beef', 'cooking oil'] -> ['spaghetti']
Confidence: 0.5714285714285714
Lift: 3.2819951870487856
====================================
Rule: ['spaghetti', 'cooking oil'] -> ['ground beef']
Confidence: 0.3025210084033613
Lift: 3.0789824749438446
====================================
Rule: ['spaghetti', 'frozen vegetables'] -> ['ground beef']
Confidence: 0.31100478468899523
Lift: 3.165328208890303
====================================
Rule: ['frozen vegetables', 'milk'] -> ['olive oil']
Confidence: 0.20338983050847456
Lift: 3.088314005352364
====================================
Rule: ['frozen vegetables', 'olive oil'] -> ['milk']
Confidence: 0.4235294117647058
Lift: 3.2684095860566447
====================================
Rule: ['mineral water', 'shrimp'] -> ['frozen vegetables']
Confidence: 0.30508474576271183
Lift: 3.200616332819722
====================================
Rule: ['spaghetti', 'frozen vegetables'] -> ['olive oil']
Confidence: 0.20574162679425836
Lift: 3.1240241752707125
====================================
Rule: ['spaghetti', 'frozen vegetables'] -> ['shrimp']
Confidence: 0.21531100478468898
Lift: 3.0131489680782684
====================================
Rule: ['spaghetti', 'frozen vegetables'] -> ['tomatoes']
Confidence: 0.23923444976076558
Lift: 3.4980460188216425
====================================
Rule: ['spaghetti', 'tomatoes'] -> ['frozen vegetables']
Confidence: 0.3184713375796179
Lift: 3.341053850607991
====================================
Rule: ['grated cheese', 'spaghetti'] -> ['ground beef']
Confidence: 0.3225806451612903
Lift: 3.283144395325426
====================================
Rule: ['herb & pepper', 'mineral water'] -> ['ground beef']
Confidence: 0.39062500000000006
Lift: 3.975682666214383
====================================
Rule: ['spaghetti', 'herb & pepper'] -> ['ground beef']
Confidence: 0.3934426229508197
Lift: 4.004359721511667
====================================
Rule: ['ground beef', 'milk'] -> ['olive oil']
```

```
Confidence: 0.22424242424242427
Lift: 3.40494417862839
=====================================
Rule: ['ground beef', 'shrimp'] -> ['spaghetti']
Confidence: 0.5232558139534884
Lift: 3.005315360233627
=====================================
Rule: ['spaghetti', 'milk'] -> ['olive oil']
Confidence: 0.20300751879699247
Lift: 3.0825089038385434
=====================================
Rule: ['mineral water', 'soup'] -> ['olive oil']
Confidence: 0.22543352601156072
Lift: 3.4230301186492245

=====================================
Rule: ['spaghetti', 'pancakes'] -> ['olive oil']
Confidence: 0.20105820105820105
Lift: 3.0529100529100526
=====================================
Rule: ['spaghetti', 'mineral water', 'milk'] -> ['frozen vegetables']
Confidence: 0.28813559322033894
Lift: 3.0228043143297376
=====================================
total number of association rules = 28
```

In [9]:

```
records
```

**Out[9]:**

```
[['shrimp',
  'almonds',
  'avocado',
  'vegetables mix',
  'green grapes',
  'whole weat flour',
  'yams',
  'cottage cheese',
  'energy drink',
  'tomato juice',
  'low fat yogurt',
  'green tea',
  'honey',
  'salad',
  'mineral water',
  'salmon',
  'antioxydant juice',
  'frozen smoothie',
  'spinach',
  'olive oil'],
 ['burgers', 'meatballs', 'eggs'],
 ['chutney'],
```

## Program No. 9

**Title:**

**Implement K means algorithm for clustering**

**Objective:**

**To learn K means algorithm for clustering**

**Reference:**

**Data Mining Introductory & Advanced Topic by Margaret H.**

**Dunham Data Mining Concept and Technique By Han & Kamber**

**Theory:**

**In statistics and machine learning, k-means clustering is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean.-Mean Clustering algorithm works?**

**Step by step explanation of K-means clustering algorithm:**

**Step 1. Begin with a decision on the value of k = number of clusters**

**Step 2. Put any initial partition that classifies the data into k clusters. You may assign the training samples randomly, or systematically as the following: Take the first k training sample as single-element clusters**

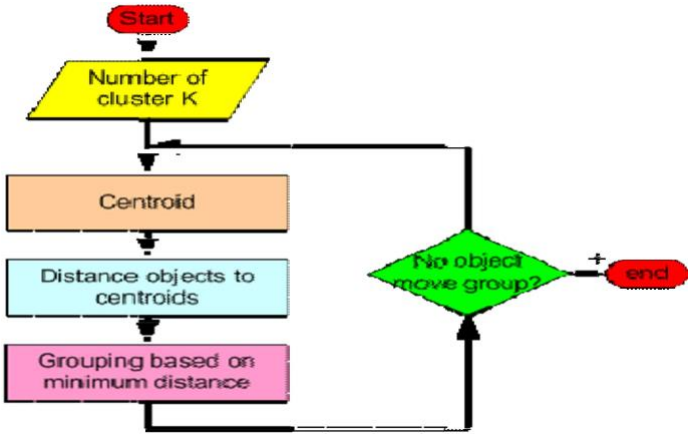
**Assign each of the remaining (N-k) training sample to the cluster with the nearest centroid. After each assignment, recomputed the centroid of the gaining cluster.**

**Step 3 . Take each sample in sequence and compute its distance from the centroid of each of the clusters. If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing the sample.**

**Step 4 . Repeat step 3 until convergence is achieved, that is until a pass through the training sample causes no new assignments.**

**If the number of data is less than the number of cluster then we assign each data as the centroid of the cluster. Each centroid will have a cluster number. If the number of data is bigger than the number of cluster, for each data, we calculate the distance to all centroid and get the minimum distance. This data is said belong to the cluster that has minimum distance from this data.**

## Flow chart for K-means Clustering




| Program No. | Marks for Execution (7) | | | Marks for Viva voce (3) | | TOTAL (10) | Signature of the Faculty |
|---|---|---|---|---|---|---|---|
| | Rubrics | | | Rubrics | | | |
| | Understanding of problem (2) | Execution (3) | Results and Documentation (2) | Conceptual Understanding and Communication of Concepts (2) | Use of appropriate Design Techniques (1) | | |
| 9 | | | | | | | |

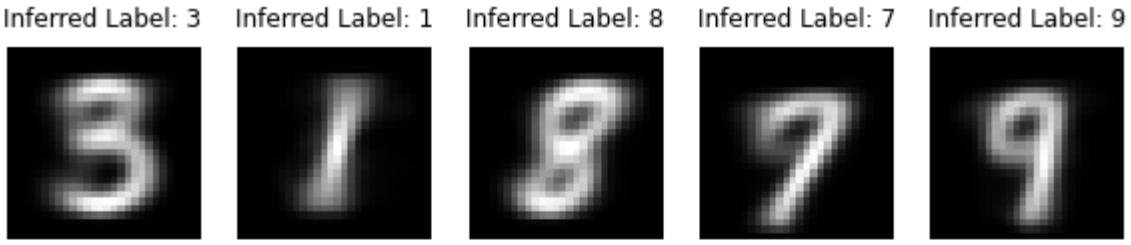
## Program No. 9

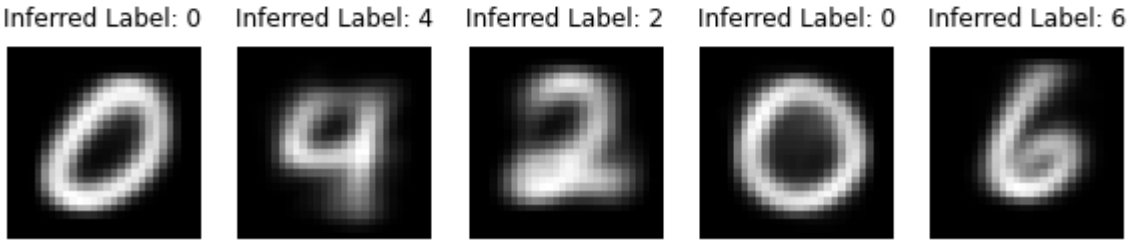
**Paste your DATA SHEET here**

In [27]:
```
X_test,y_test=loadlocal_mnist(images_path='t10k-images.idx3-ubyte',labels_path='
```

In [28]:
```
 kmeans = MiniBatchKMeans(n_clusters = 256)
kmeans.fit(X)
cluster_labels = infer_cluster_labels(kmeans, y)

# predict labels for testing data
test_clusters = kmeans.predict(X_test)
predicted_labels = infer_data_labels(kmeans.predict(X_test), cluster_labels)

# calculate and print accuracy
print('Accuracy: {}\n'.format(metrics.accuracy_score(y_test, predicted_labels)))
```

```
Accuracy: 0.5463
```



# Case Study:

**Documents to be attached:**

**Description of the case study**

**Data visualization techniques used and justification**

**Pre-processing techniques used and justification**

**Algorithm to build the model**

**Evaluation of the model**

**Conclusion of the Experiment wrt dataset.**

| Program No. | Marks for Execution (7) | | | Marks for Viva voce (3) | | TOTAL (10) | Signature of the Faculty |
|---|---|---|---|---|---|---|---|
| | Rubrics | | | Rubrics | | | |
| | Understanding of problem (2) | Execution (3) | Results and Documentation (2) | Conceptual Understanding and Communication of Concepts (2) | Use of appropriate Design Techniques (1) | | |
| **Case Study** | | | | | | | |

# Banks’ Marketing Campaign

## 1. Description

In banks, huge data records information about their customers. This data can be used to create and keep clear relationships and connections with the customers in order to target them individually for definite products or banking offers. Usually, the selected customers are contacted directly through: personal contact, telephone cellular, mail, and email or any other contacts to advertise the new product/service or give an offer, this kind of marketing is called direct marketing. In fact, direct marketing is in the main a strategy of many of the banks and insurance companies for interacting with their customers.

We are given the data of direct marketing campaigns (phone calls) of a Portuguese banking institution. The classification goal is to predict if the client will subscribe to a term deposit (target variable y).

### About the dataset

1. age (numeric)
2. job : type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed' ,'services','student','technician','unemployed','unknown')
3. marital : marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)
4. education (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university. degree','unknown')
5. default: has credit in default? (categorical: 'no','yes','unknown')
6. housing: has a housing loan? (categorical: 'no','yes','unknown')
7. loan: has personal loan? (categorical: 'no','yes','unknown') #### related with the last contact of the current campaign:
8. contact: contact communication type (categorical: 'cellular','telephone')
9. month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
10. day_of_week: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')
11. duration: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should

be discarded if the intention is to have a realistic predictive model. #### other attributes:

12. campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
13. pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
14. previous: number of contacts performed before this campaign and for this client (numeric)
15. poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success') #### social and economic context attributes
16. emp.var.rate: employment variation rate - quarterly indicator (numeric)
17. cons.price.idx: consumer price index - monthly indicator (numeric)
18. cons.conf.idx: consumer confidence index - monthly indicator (numeric)
19. euribor3m: euribor 3 month rate - daily indicator (numeric) - Euro Interbank Offered Rate. The Euribor rates are based on the average interest rates at which a large panel of European banks borrow funds from one another
20. nr.employed: number of employees - quarterly indicator (numeric)

Output variable (desired target):

y - has the client subscribed a term deposit? (binary: 'yes','no')

# 2. Data Visualization

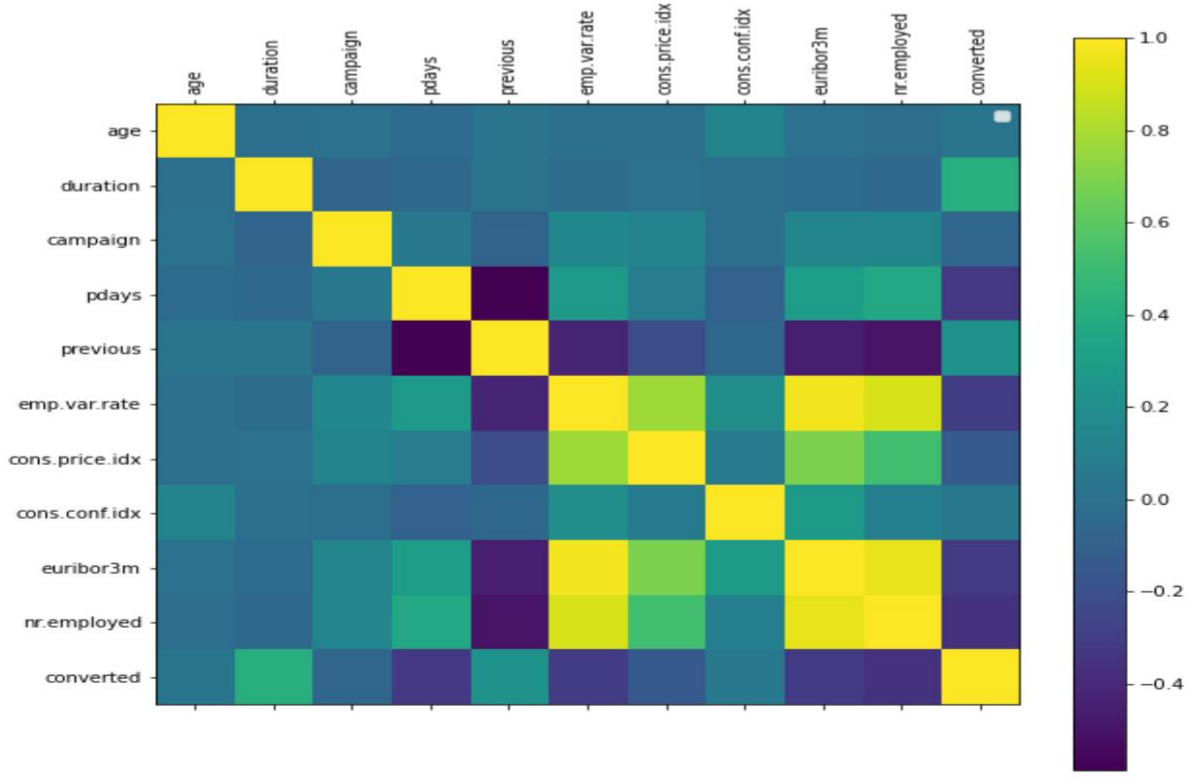
**No. of unique values in each column**

```
Missing values :    0

Unique values :
 age                78
job                 12
marital              4
education            8
default              3
housing              3
loan                 3
contact              2
month               10
day_of_week          5
duration          1544
campaign            42
pdays               27
previous             8
poutcome             3
emp.var.rate        10
cons.price.idx      26
cons.conf.idx       26
euribor3m          316
nr.employed         11
y                    2
dtype: int64
```

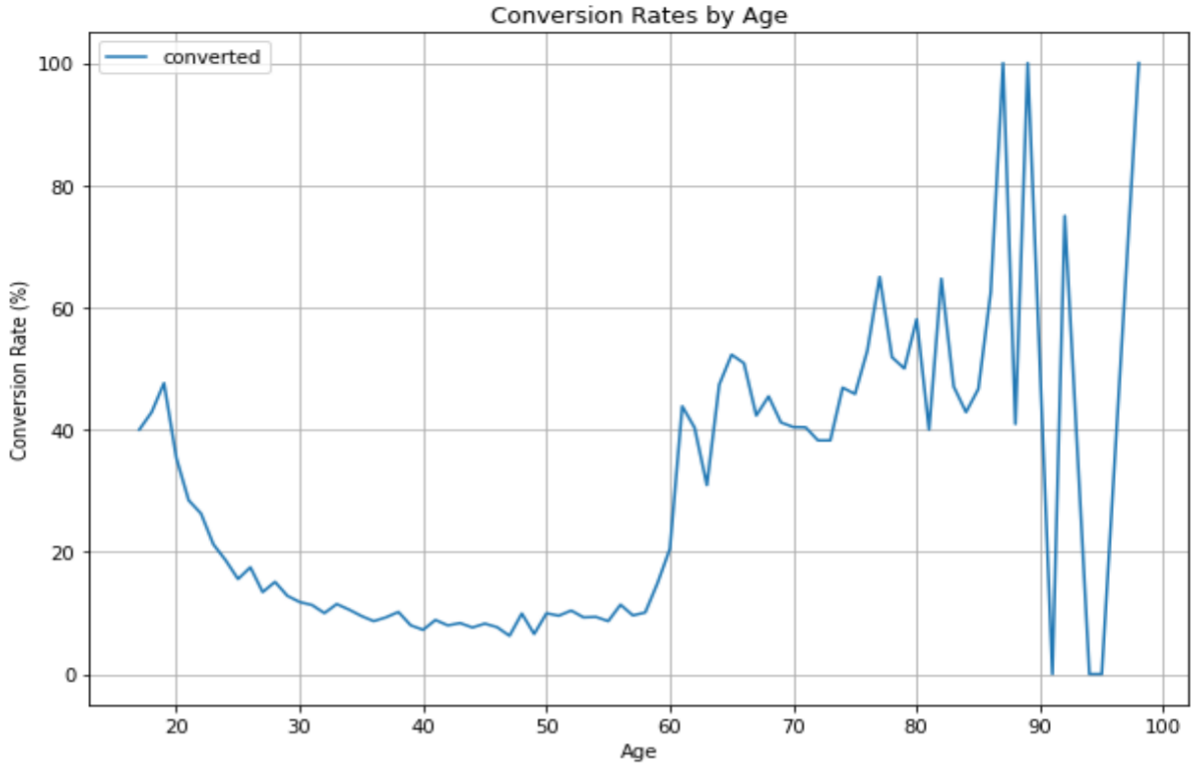
**Datatypes of each feature**

```
In [18]: bank.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41176 entries, 0 to 41175
Data columns (total 22 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             41176 non-null  int64
 1   job             41176 non-null  object
 2   marital         41176 non-null  object
 3   education       41176 non-null  object
 4   default         41176 non-null  object
 5   housing         41176 non-null  object
 6   loan            41176 non-null  object
 7   contact         41176 non-null  object
 8   month           41176 non-null  object
 9   day_of_week     41176 non-null  object
 10  duration        41176 non-null  int64
 11  campaign        41176 non-null  int64
 12  pdays           41176 non-null  int64
 13  previous        41176 non-null  int64
 14  poutcome        41176 non-null  object
 15  emp.var.rate    41176 non-null  float64
 16  cons.price.idx  41176 non-null  float64
 17  cons.conf.idx   41176 non-null  float64
 18  euribor3m       41176 non-null  float64
 19  nr.employed     41176 non-null  float64
 20  y               41176 non-null  object
 21  converted       41176 non-null  int64
dtypes: float64(5), int64(6), object(11)
memory usage: 6.9+ MB
```

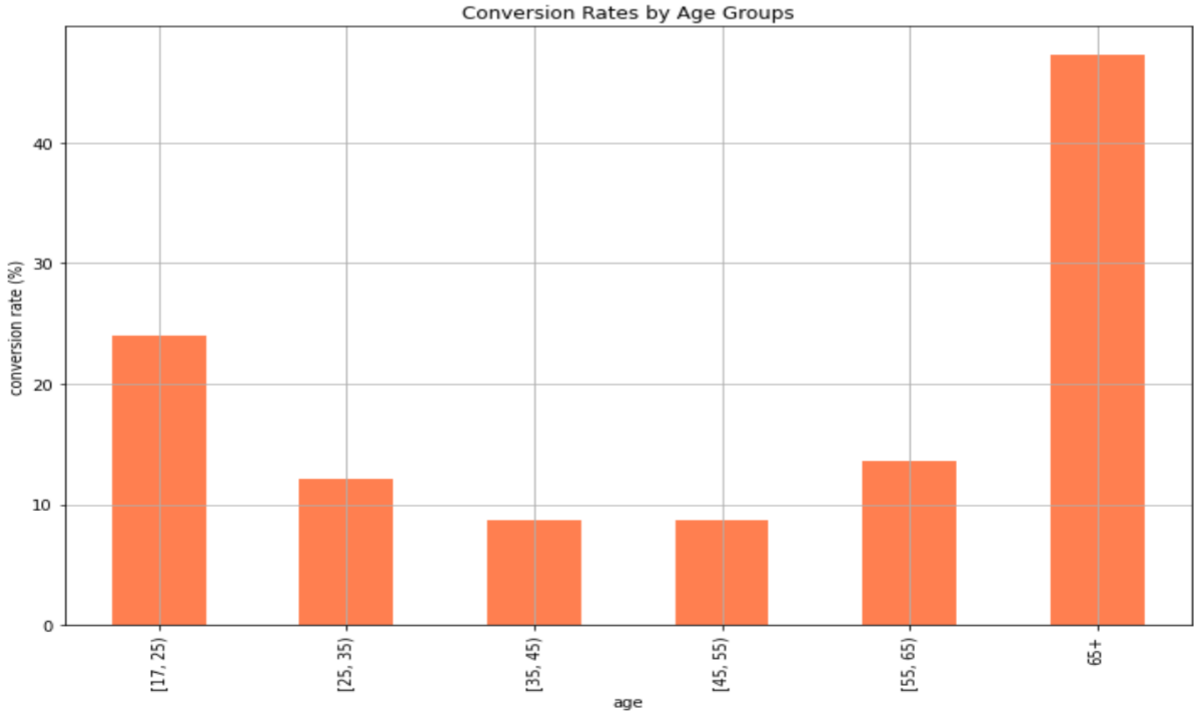
**Heatmap of correlation matrix**



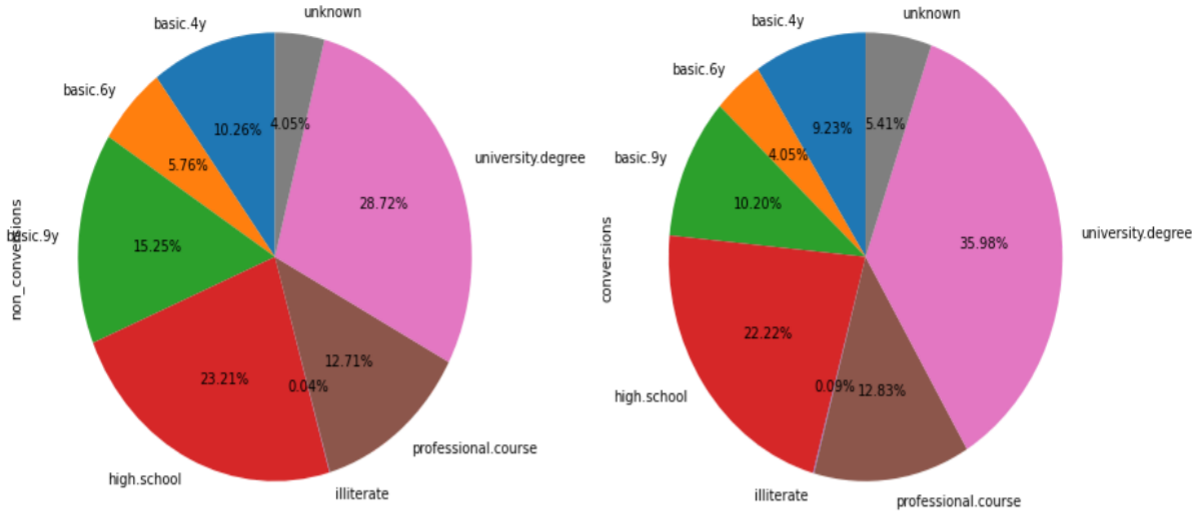
**Conversion rates by age**

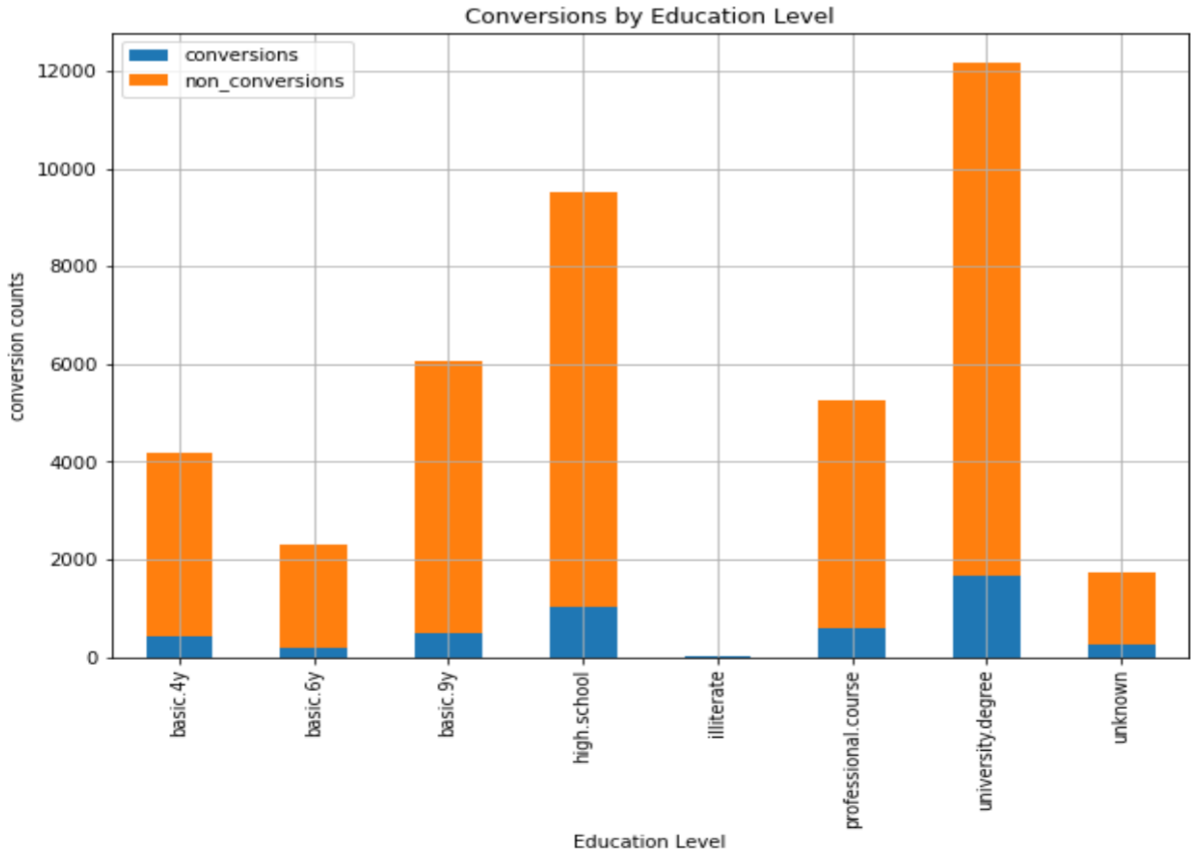



**Conversion rates based on Age Groups**

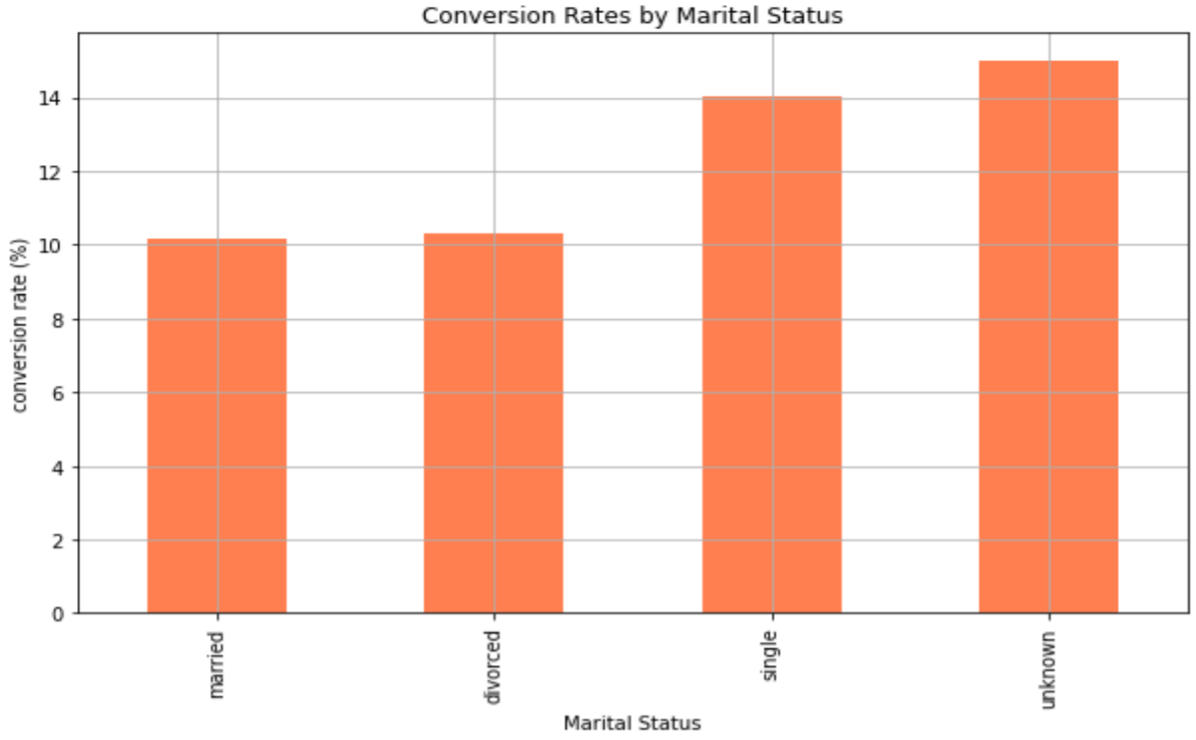


**conversion rates based on Education level**

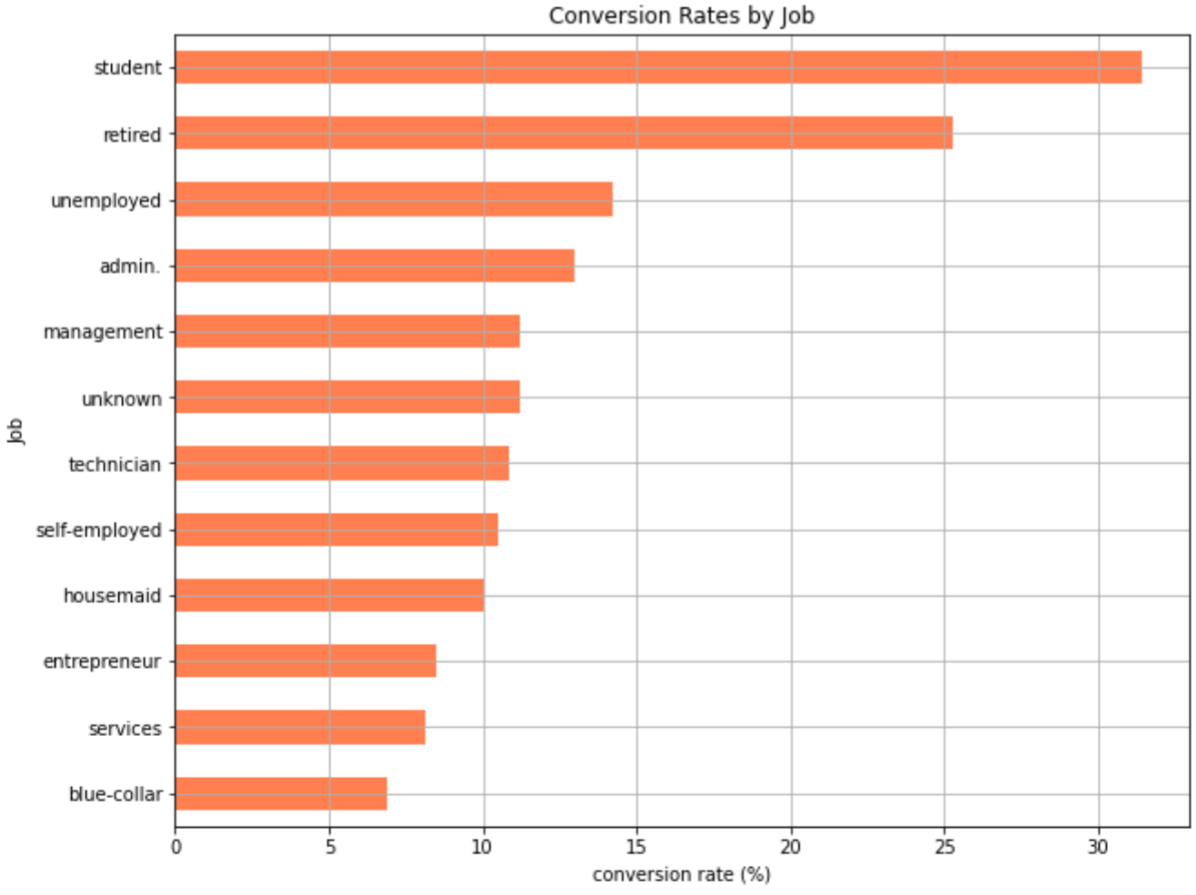


**conversion rate based on marital status**




**Conversion rates based on job**



# 3. Data pre-processing techniques used and justification:

## Data integration:

- The dataset extracted is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be (or not) subscribed.
- There were four datasets present, ‘bank.csv’, ‘bank-additional.csv’,’bank-full.csv’ and ‘bank-additional-full.csv’.
- Out of these, since bank.csv and bank-full.csv had the same columns, and the same with the other two csv files.
- Hence, ‘bank.csv’ and ‘bank-full.csv’ files are each loaded into pandas dataframes and concatenated together.
- Also, ‘bank-additional.csv’ and ‘bank-additional-full.csv’ are each loaded into pandas dataframes and concatenated together. The two concatenated datasets are as shown:

```
df_bank_final.head()
```

| | age | job | marital | education | default | balance | housing | loan | contact | day | month | duration | campaign | pdays | previous | poutcome | y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 30 | unemployed | married | primary | no | 1787 | no | no | cellular | 19 | oct | 79 | 1 | -1 | 0 | unknown | no |
| 1 | 33 | services | married | secondary | no | 4789 | yes | yes | cellular | 11 | may | 220 | 1 | 339 | 4 | failure | no |
| 2 | 35 | management | single | tertiary | no | 1350 | yes | no | cellular | 16 | apr | 185 | 1 | 330 | 1 | failure | no |
| 3 | 30 | management | married | tertiary | no | 1476 | yes | yes | unknown | 3 | jun | 199 | 4 | -1 | 0 | unknown | no |
| 4 | 59 | blue-collar | married | secondary | no | 0 | yes | no | unknown | 5 | may | 226 | 1 | -1 | 0 | unknown | no |

```
df_bank_additional_final.head()
```

| | age | job | marital | education | default | housing | loan | contact | month | day_of_week | ... | campaign | pdays | previous | poutcome | emp.var.r |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 30 | blue-collar | married | basic.9y | no | yes | no | cellular | may | fri | ... | 2 | 999 | 0 | nonexistent | -1.8 |
| 1 | 39 | services | single | high.school | no | no | no | telephone | may | fri | ... | 4 | 999 | 0 | nonexistent | 1.1 |
| 2 | 25 | services | married | high.school | no | yes | no | telephone | jun | wed | ... | 1 | 999 | 0 | nonexistent | 1.4 |
| 3 | 38 | services | married | basic.9y | no | unknown | unknown | telephone | jun | fri | ... | 3 | 999 | 0 | nonexistent | 1.4 |
| 4 | 47 | admin. | married | university.degree | no | yes | no | cellular | nov | mon | ... | 1 | 999 | 0 | nonexistent | -0.1 |

5 rows × 21 columns

- Now, the resulting two datasets are merged together and the duplicates are dropped to obtain a final dataset. It has 41176 rows, and 23 columns.

## Data Cleaning:

Removing ‘Nan’ values: Since the two columns, ‘balance’ and ‘day’ were entirely NaNs after merge, they are dropped and the final dataset now has 21 columns. It is as shown:

```
df_final.head()
```

| | age | job | marital | education | default | housing | loan | contact | month | day_of_week | ... | campaign | pdays | previous | poutcome | emp.var.r |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 30 | blue-collar | married | basic.9y | no | yes | no | cellular | may | fri | ... | 2 | 999 | 0 | nonexistent | -1.8 |
| 1 | 39 | services | single | high.school | no | no | no | telephone | may | fri | ... | 4 | 999 | 0 | nonexistent | 1.1 |
| 2 | 25 | services | married | high.school | no | yes | no | telephone | jun | wed | ... | 1 | 999 | 0 | nonexistent | 1.4 |
| 3 | 38 | services | married | basic.9y | no | unknown | unknown | telephone | jun | fri | ... | 3 | 999 | 0 | nonexistent | 1.4 |
| 4 | 47 | admin. | married | university.degree | no | yes | no | cellular | nov | mon | ... | 1 | 999 | 0 | nonexistent | -0.1 |

5 rows × 21 columns

### Data Transformation (Feature Extraction):

1. The target column ‘y’ is Label encoded into 1s and 0s.
2. The columns which are categorical like job, marital, etc are one hot encoded so that each of these columns are converted to binary values and hence can be used as features in the model.
3. The resulting dataframe that is used to train the model has 54 columns and is as shown below:

```
df_final.columns,df_final.shape
```

```
(Index(['age', 'duration', 'campaign', 'pdays', 'previous', 'emp.var.rate',
       'cons.price.idx', 'cons.conf.idx', 'euribor3m', 'nr.employed', 'y',
       'job_blue-collar', 'job_entrepreneur', 'job_housemaid',
       'job_management', 'job_retired', 'job_self-employed', 'job_services',
       'job_student', 'job_technician', 'job_unemployed', 'job_unknown',
       'marital_married', 'marital_single', 'marital_unknown',
       'education_basic.6y', 'education_basic.9y', 'education_high.school',
       'education_illiterate', 'education_professional.course',
       'education_university.degree', 'education_unknown', 'default_unknown',
       'default_yes', 'housing_unknown', 'housing_yes', 'loan_unknown',
       'loan_yes', 'contact_telephone', 'month_aug', 'month_dec', 'month_jul',
       'month_jun', 'month_mar', 'month_may', 'month_nov', 'month_oct',
       'month_sep', 'day_of_week_mon', 'day_of_week_thu', 'day_of_week_tue',
       'day_of_week_wed', 'poutcome_nonexistent', 'poutcome_success'],
      dtype='object'),
 (41176, 54))
```

```
df_final.head()
```

| ate | cons.price.idx | cons.conf.idx | euribor3m | nr.employed | ... | month_may | month_nov | month_oct | month_sep | day_of_week_mon | day_of_week_thu | day_ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 92.893 | -46.2 | 1.313 | 5099.1 | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 93.994 | -36.4 | 4.855 | 5191.0 | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 94.465 | -41.8 | 4.962 | 5228.1 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 94.465 | -41.8 | 4.959 | 5228.1 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 93.200 | -42.0 | 4.191 | 5195.8 | ... | 0 | 1 | 0 | 0 | 1 | 0 | 0 |

4. Since most of the features had moderate correlation, all the features except ‘y’ are chosen as X and the target variable as ‘y’.

# 4. Machine Learning Algorithms used for the model:

1. The X and y variables are split into training and test datasets.
2. The algorithms used are Logistic Regression and Support Vector Machine.
3. Justification:
    a. **Logistic Regression:**It is a simple model, easy to train and interpret. Since the feature space is large and sparse, logistic regression is chosen as it is relatively fast, and makes no assumptions on the features. Logistic regression is less inclined to over-fitting. It works best for binary classification.
    b. **Support Vector Machine**: SVM works relatively well when there is a clear margin of separation between classes. SVM is more effective in high dimensional spaces. SVM is relatively memory efficient.
4. Both algorithms are imported from ‘sklearn’ library and are fit with X and y training datasets with default parameters.
5. For Logistic Regression, the parameter ‘max_iter’ is set to 1000 to ensure convergence.

# 5. Testing the accuracy of the models:

- The accuracy measures of both the models are as shown:
  LR - 0.9176458900581352 SVM-0.8954301273088527

- Both the models yield almost the same accuracy scores and are also not over-fitting.
- Confusion matrices for both the models are generated to know the true positives, false positives, true negatives and false negatives. It is as shown:

### LR confusion matrix and classification report :

```
[[11677   341]
 [  914   657]]
```

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.93 | 0.97 | 0.95 | 12018 |
| 1 | 0.66 | 0.42 | 0.51 | 1571 |
| accuracy | | | 0.91 | 13589 |
| macro avg | 0.79 | 0.69 | 0.73 | 13589 |
| weighted avg | 0.90 | 0.91 | 0.90 | 13589 |

### SVM Confusion matrix and classification report :

```
[[11820   198]
 [ 1223   348]]
```

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.91 | 0.98 | 0.94 | 12018 |
| 1 | 0.64 | 0.22 | 0.33 | 1571 |
| accuracy | | | 0.90 | 13589 |
| macro avg | 0.77 | 0.60 | 0.64 | 13589 |
| weighted avg | 0.88 | 0.90 | 0.87 | 13589 |

- As observed, SVM produces less false positives, but more false negatives than Log Regression.

### Improving performance scores:

**Ensemble method:** A voting classifier that is importer from sklean's ensemble class is used to form an ensemble of both the LR and SVM classifiers.

Accuracy of voting classifier: 0.902789020531312

**Confusion matrix and classification report:**

```
[[11978   102]
 [ 1219   290]]

              precision    recall  f1-score   support

           0       0.91      0.99      0.95     12080
           1       0.74      0.19      0.31      1509

    accuracy                           0.90     13589
   macro avg       0.82      0.59      0.63     13589
weighted avg       0.89      0.90      0.88     13589
```

We can see that even though the accuracy remains about the same, the precision, recall and f1 scores are significantly better than the results of the previous two models individually.

## Viva - Voce Questions

1. Name Data mining techniques?
2. Name areas of applications of data mining?
3. Name different classification methods?
4. Name different methods in clustering?
5. What are the types of tasks that are carried out during data mining?
6. What is Data cleaning?
7. Explain Data reduction and transformation?
8. What is Discrete and Continuous data in Data mining world?\
9. What is Naïve Bayes Algorithm?
10. What is HDFS? Why is it important
11. Describe Hadoop ecosystem
12. How is pig different than hive in Hadoop?
13. Map parallel processing and Hadoop.
14. What are the limitations of R tool?
15. What is the importance of siket and pandas in python?
16. What is the limitation of Navie Bayes theorem ?
17. When to use SVM?
18. Give the application of Apriori algorithm.
19. What is the scope of regression in data mining?
20. What is a Decision Tree Algorithm?