**PRECIDING IMDB SCORE**

**INTRODUCTION:**

**1.BACKGROUND**

**2.DATADESCRIPTION**

**3.PROBLEM STATEMENT**

**. DATAEXPLORATION**

**. LOADDATA**

**.REMOVE DUPLICATES**

**.TIDY UPMOVIETITLE**

**.SPLIT GENERS**

**4.DATA CLEANING**

**.MISSING VALUES**

**.ADD COLUMNS**

**.REMOVE COLUMNS**

**4. DATA VIRUTALIZATION**

**.HISTOGRAM OF MOVIE RELEASED**

**.TOP 20 MOVIES BASED ON ITS PROFIT**

**.TOP 20 MOVIES BASED ON ITSRETURN ON INVESTEMENT**

**.TOP20DIRECTORIESWITH HIGHRESTAVERAGE IMDB SCORE**

**.5.DATA PRE PROCESSING**

**.REMOVE NAMES**

**.REMOVE LINEAR DEPENDENT VARIABLES**

**.REMOVE HIGHLY CORRLETED VARIABLES**

**.BIN RESPONSW VARIABLE**

**.6.IMPLEMENT ALGORITHM**

**.CLASSIFICATION TREE**

**.LOGISTIC REGRESSION**

**.K\_NEAREST NEIGHOURS**

**.SUPPORT VECTOR CLASSIFIER**

**1.BACKGROUND :the original use was theatrical; the word was applied to painting ("part of a picture representing what is furthest from the spectator") by 1752.**

**1.1. DATA DESCRIPTION:Document Format**

**The document should be a simple text document. While it could be constructed in a word processor and saved in that format (i.e. .odt or .docx format), ideally it should be saved as a simple text file (.txt) for maximum compatibility with future software. Alternately, PDF files will also be accepted if formatting of the file contents is critical and cannot be achieved via a simple text file. The file name should be DataDescription-<DatasetName> with the appropriate file extension based on its format (.txt, .odt, .docx, etc.) <DatasetName> should be the title of the dataset (or part of the title if it is long) and is used to distinguish the data description document from the others in the repository.**

**1.2 PROBLEM STATEMENT:A problem statement is a short, clear explanation of an issue or challenge that sums up what you want to change. It helps you, team members, and other stakeholders to focus on the problem, why it's important, and who it impacts. A good problem statement should create awareness and stimulate creative thinking.**

**2.DATA EXPLORATION:**

**2.1 LOAD DATA :Suppose the user have a .txt file name 'pub.txt' in a folder of its own, containing 1 record per line and separated by tabs and arranged in order as the columns listed in the table. You can use LOAD statement to populate the table. For missing values, the user can use NULL values and that should be represented by ‘\N’ in the text file.**

**Syntax:**

**LOAD DATA LOCAL INFILE '[path/][file\_name]' INTO TABLE [table\_name ];**

**Arguments:**

**Name Description**

**path The address of the file.**

**file\_name The name of the .txt file.**

**table\_name The table where the data will be loaded.**

**Example:**

**In the following code, the content of the text file pub.txt will be loaded in the publisher table.**

**The path of the file should be mentioned.**

**The text file contains the row like -**

**P002<tab>BPP Publication<tab>Mumbai<tab>India<tab>New Delhi<tab>10<tab>1985-10-01**

**Code:**

**LOAD DATA LOCAL INFILE 'pub.txt' INTO TABLE publisher;**

**Sample table: publisher**

**MySQL LOAD statement with line terminator**

**MySQL LOAD statements with ‘\r\n’ as a line terminator can also be used to populate a table.**

**Example:**

**The following code will load the records from the pub.txt file in publisher table. The ‘\r\n’ can be used as a line terminator.**

**The text file contains the row like -**

**P002<tab>BPP Publication<tab>Mumbai<tab>India<tab>New Delhi<tab>10<tab>1985-10-01\r\n**

**Sample table: publisher**

**Code:**

**LOAD DATA LOCAL INFILE 'path/pub.txt' INTO TABLE publisher**

**LINES TERMINATED BY \r\n ;**

**MySQL loading data into a table with insert statement**

**To insert new records into a table INSERT statement can be used. The values will be supplied by the user in the same order as columns are listed in the table. String and date values are needed to be specified within quoted string.**

**SyntaxSuppose the user have a .txt file name 'pub.txt' in a folder of its own, containing 1 record per line and separated by tabs and arranged in order as the columns listed in the table. You can use LOAD statement to populate the table. For missing values, the user can use NULL values and that should be represented by ‘\N’ in the text file.**

**Syntax:**

**LOAD DATA LOCAL INFILE '[path/][file\_name]' INTO TABLE [table\_name ];**

**Arguments:**

**Name Description**

**path The address of the file.**

**file\_name The name of the .txt file.**

**table\_name The table where the data will be loaded.**

**Example:**

**In the following code, the content of the text file pub.txt will be loaded in the publisher table.**

**The path of the file should be mentioned.**

**The text file contains the row like -**

**P002<tab>BPP Publication<tab>Mumbai<tab>India<tab>New Delhi<tab>10<tab>1985-10-01**

**Code:**

**LOAD DATA LOCAL INFILE 'pub.txt' INTO TABLE publisher;**

**Sample table: publisher**

**MySQL LOAD statement with line terminator**

**MySQL LOAD statements with ‘\r\n’ as a line terminator can also be used to populate a table.**

**Example:**

**The following code will load the records from the pub.txt file in publisher table. The ‘\r\n’ can be used as a line terminator.**

**The text file contains the row like -**

**P002<tab>BPP Publication<tab>Mumbai<tab>India<tab>New Delhi<tab>10<tab>1985-10-01\r\n**

**Sample table: publisher**

**Code:**

**LOAD DATA LOCAL INFILE 'path/pub.txt' INTO TABLE publisher**

**LINES TERMINATED BY \r\n ;**

**MySQL loading data into a table with insert statement**

**To insert new records into a table INSERT statement can be used. The values will be supplied by the user in the same order as columns are listed in the table. String and date values are needed to be specified within quoted string.**

**SyntaxSuppose the user have a .txt file name 'pub.txt' in a folder of its own, containing 1 record per line and separated by tabs and arranged in order as the columns listed in the table. You can use LOAD statement to populate the table. For missing values, the user can use NULL values and that should be represented by ‘\N’ in the text file.**

**Syntax:**

**LOAD DATA LOCAL INFILE '[path/][file\_name]' INTO TABLE [table\_name ];**

**Arguments:**

**Name Description**

**path The address of the file.**

**file\_name The name of the .txt file.**

**table\_name The table where the data will be loaded.**

**Example:**

**In the following code, the content of the text file pub.txt will be loaded in the publisher table.**

**The path of the file should be mentioned.**

**The text file contains the row like -**

**P002<tab>BPP Publication<tab>Mumbai<tab>India<tab>New Delhi<tab>10<tab>1985-10-01**

**Code:**

**LOAD DATA LOCAL INFILE 'pub.txt' INTO TABLE publisher;**

**Sample table: publisher**

**MySQL LOAD statement with line terminator**

**MySQL LOAD statements with ‘\r\n’ as a line terminator can also be used to populate a table.**

**Example:**

**The following code will load the records from the pub.txt file in publisher table. The ‘\r\n’ can be used as a line terminator.**

**The text file contains the row like -**

**P002<tab>BPP Publication<tab>Mumbai<tab>India<tab>New Delhi<tab>10<tab>1985-10-01\r\n**

**Sample table: publisher**

**Code:**

**LOAD DATA LOCAL INFILE 'path/pub.txt' INTO TABLE publisher**

**LINES TERMINATED BY \r\n ;**

**MySQL loading data into a table with insert statement**

**To insert new records into a table INSERT statement can be used. The values will be supplied by the user in the same order as columns are listed in the table. String and date values are needed to be specified within quoted string.**

**SyntaxSuppose the user have a .txt file name 'pub.txt' in a folder of its own, containing 1 record per line and separated by tabs and arranged in order as the columns listed in the table. You can use LOAD statement to populate the table. For missing values, the user can use NULL values and that should be represented by ‘\N’ in the text file.**

**Syntax:**

**LOAD DATA LOCAL INFILE '[path/][file\_name]' INTO TABLE [table\_name ];**

**Arguments:**

**Name Description**

**path The address of the file.**

**file\_name The name of the .txt file.**

**table\_name The table where the data will be loaded.**

**Example:**

**In the following code, the content of the text file pub.txt will be loaded in the publisher table.**

**The path of the file should be mentioned.**

**The text file contains the row like -**

**P002<tab>BPP Publication<tab>Mumbai<tab>India<tab>New Delhi<tab>10<tab>1985-10-01**

**Code:**

**LOAD DATA LOCAL INFILE 'pub.txt' INTO TABLE publisher;**

**Sample table: publisher**

**MySQL LOAD statement with line terminator**

**MySQL LOAD statements with ‘\r\n’ as a line terminator can also be used to populate a table.**

**Example:**

**The following code will load the records from the pub.txt file in publisher table. The ‘\r\n’ can be used as a line terminator.**

**The text file contains the row like -**

**P002<tab>BPP Publication<tab>Mumbai<tab>India<tab>New Delhi<tab>10<tab>1985-10-01\r\n**

**Sample table: publisher**

**Code:**

**LOAD DATA LOCAL INFILE 'path/pub.txt' INTO TABLE publisher**

**LINES TERMINATED BY \r\n ;**

**MySQL loading data into a table with insert statement**

**To insert new records into a table INSERT statement can be used. The values will be supplied by the user in the same order as columns are listed in the table. String and date values are needed to be specified within quoted string.**

**2.2 REMOVE DUPLICATES:**

**To remove dupliates from ArrayList, we can convert it into Set. Since Set doesn't contain duplicate elements, it will have only unique elements.**

**Let's see an example to remove duplicates from ArrayList:**

**public class RemoveDuplicateArrayList {**

**public static void main(String[] args) {**

**List<String> l = new ArrayList<String>();**

**l.add("Mango");**

**l.add("Banana");**

**l.add("Mango");**

**l.add("Apple");**

**System.out.println(l.toString());**

**Set<String> s = new LinkedHashSet<String>(l);**

**System.out.println(s);**

**}**

**2.3TIDY UP MOVIE TITLE:**

**n each episode, Kondo visits a different American family household in need of organizing and de-cluttering.[10] Each family has individual backgrounds and needs, which the show addresses with both hands-on guidance from Kondo and cutaways of Kondo giving additional KonMari explanations.[11]**

**Episodes**

**No. Title Original air date**

**1 "Tidying With Toddlers" January 1, 2019**

**2 "Empty Nesters" January 1, 2019**

**3 "The Downsizers" January 1, 2019**

**4 "Sparking Joy After a Loss" January 1, 2019**

**5 "From Students to Improvements" January 1, 2019**

**6 "Breaking Free from a Mountain of Stuff" January 1, 2019**

**7 "Making Room for a Baby" January 1, 2019**

**8 "When Two (Messes) Become One" January 1, 2019**

**Methodology of the show**

**Kondo's ideology towards de-cluttering is called the KonMari method. Her process, which she explains in the series, includes having participants go through their homes section by section—"1) Clothing, 2) Books, 3) Paper, 4) Komono (kitchen, bathroom, garage and miscellaneous) 5) Sentimental items.[12] Then, when the items are gathered together, Kondo asks the participants to go through the individual items one by one and only keep what "sparks joy".**

**While her method to cleaning is effective in organizing people's homes and environments, Kondo came to understand that her method was "far more psychological than it was practical."[13] This perspective is supported well by studies done on how people perform and feel in tidy versus messy environments.[14] Research suggests that "clean organized environments" have a variety of psychologically beneficial effects—such as clearer thought processes, increased confidence, and abilities.[14]**

**Further research shows that an untidy environment can negatively impact many aspects of human health—such as mood and stress levels, memory capabilities, and even the ability to process other people's facial expressions.[15] When people's personal space is "de-cluttered," they are better able to sleep, focus, and hold feelings of satisfaction with their lives.[15]**

**The evidence that Kondo's show positively affects the mental health of the people featured in it isn't just seen on the show itself, but in interviews given by participants later on.[16] One couple, seen in Episode 1, "Tidying with Toddlers," Kevin and Rachel Friend, said after Kondo's help that, "Chaos for us doesn’t happen anymore. We’re able to take on things. Nothing overwhelms us anymore. Now, it's just our lifestyle."**

**Reception**

**A protester at 2019 Women's March in Washington D.C. holding a sign that reads "This administration does not spark joy!", derived from Kondo's idea of only keeping things that "spark joy"**

**Reviews of the show have been generally positive.[17] On the review aggregator site Rotten Tomatoes, the show holds a score of 81%, noting that Marie Kondo "makes for a delightful instructor".[18] On Metacritic, the show has a score of 69 based on eight critics, indicating "generally favorable reviews".[19]**

**In The Atlantic, Sarah Archer says that the show is about "cultivating empathy for the things that surround us", noting that Kondo's empathy is the key to her success in helping people.[20] In Vice, Nicole Clark writes that by showcasing the inner workings of family homes, the show has unintentionally highlighted differing gender expectations towards tidying and organizing households.[21]**

**Jack Seale in The Guardian is less positive, noting that the show is simply "a show where a woman just tells people to tidy up." He adds that the before and after reveals lack excitement, revealing the same house except tidier.[22]**

**Others have been on the fence, like Kristin van Ogtrop from Time, who wrote about the "completely bananas" method Kondo employs, suggesting that while necessary items should still be kept around, it is still a worthwhile endeavor to mentally refocus on what brings one joy.[23]**

**Effect on donations to charity shops**

**In the immediate wake of the show's release, some charity shops saw an increase in the number of donations received. Donations to Goodwill stores in the Washington D.C. area were up by 66% for the first week of January, an effect attributed to the show encouraging people to tidy their houses.[8] Beacon's Closet in New York also saw similar increases in donations.[24] In western Sydney, charity St Vincent de Paul saw a 38% rise in donations in the first three weeks after the show's premiere.[25] Nationwide, Goodwill reported a 10%–20% increase in donations for the year, as of FebrARY**

**2.4 SPLITGENERS:**

**The split gene theory is a theory of the origin of introns, long non-coding sequences in eukaryotic genes between the exons.[1][2][3] The theory holds that the randomness of primordial DNA sequences would only permit small (< 600bp) open reading frames (ORFs), and that important intron structures and regulatory sequences are derived from stop codons. In this introns-first framework, the spliceosomal machinery and the nucleus evolved due to the necessity to join these ORFs (now "exons") into larger proteins, and that intronless bacterial genes are less ancestral than the split eukaryotic genes. The theory originated with Periannan Senapathy.**

**The theory provides solutions to key questions concerning the split gene architecture, including split eukaryotic genes, exons, introns, splice junctions, and branch points, based on the origin of split genes from random genetic sequences. It also provides possible solutions to the origin of the spliceosomal machinery, the nuclear boundary and the eukaryotic cell.**

**This theory led to the Shapiro–Senapathy algorithm, which provides the methodology for detecting the splice sites, exons and split genes in eukaryotic DNA, and which is the main method for detecting splice site mutations in genes that cause hundreds of diseases.**

**Split gene theory requires a separate origin of all eukaryotic species. It also requires that the simpler prokaryotes evolved from eukaryotes. This completely contradicts the scientific consensus about the formation of eukaryotic cells by endosymbiosis of bacteria. In 1994, Senapathy wrote a book about this aspect of his theory - The Independent Birth of Organisms. It proposed that all eukaryotic genomes were formed separately in a primordial pool. Dutch biologist Gert Korthoff criticized the theory by posing various problems that cannot be explained by a theory of independent origins. He pointed out that various eukaryotes need nurturing and called this the 'boot problem', in that even the initial eukaryote needed parental care. Korthoff notes that a large fraction of eukaryotes are parasites. Senapathy's theory would require a coincidence to explain their existence.[4][5] Senapathy's theory cannot explain the strong evidence for common descent (homology, universal genetic code, embryology, fossil record.)[6]**

**3.DATA CLEANING:**

**MISSIN VALUES:Missing data, or missing values, occur when you don’t have data stored for certain variables or participants. Data can go missing due to incomplete data entry, equipment malfunctions, lost files, and many other reasons.**

**In any dataset, there are usually some missing data. In quantitative research, missing values appear as blank cells in your spreadsheet.Missing completely at random**

**When data are missing completely at random (MCAR), the probability of any particular value being missing from your dataset is unrelated to anything else.**

**The missing values are randomly distributed, so they can come from anywhere in the whole distribution of your values. These MCAR data are also unrelated to other unobserved variables.**

**Example: MCAR data**

**You note that there are a few missing values in your holiday spending dataset. Some people started answering your survey but dropped out or skipped a question.**

**However, you note that you have data points from a wide distribution, ranging from low to high values.**

**Therefore, you conclude that the missing values aren’t related to any specific holiday spending amount range.**

**Data are often considered MCAR if they seem unrelated to specific values or other variables. In practice, it’s hard to meet this assumption because “true randomness” is rare.**

**When data are missing due to equipment malfunctions or lost samples, they are considered MCAR.**

**Missing at random**

**Data missing at random (MAR) are not actually missing at random; this term is a bit of a misnomer.**

**This type of missing data systematically differs from the data you’ve collected, but it can be fully accounted for by other observed variables.**

**The likelihood of a data point being missing is related to another observed variable but not to the specific value of that data point itself.**

**Example: MAR data**

**You repeat your data collection with a new group. You notice that there are more missing values for adults aged 18–25 than for other age groups.**

**But looking at the observed data for adults aged 18–25, you notice that the values are widely spread. It’s unlikely that the missing data are missing because of the specific values themselves.**

**Instead, some younger adults may be less inclined to reveal their holiday spending amounts for unrelated reasons (e.g., more protective of their privacy).**

**Missing not at random**

**Data missing not at random (MNAR) are missing for reasons related to the values themselves.**

**Example: MNAR data**

**In the new dataset, you also notice that there are fewer low values. Some participants with low incomes avoid reporting their holiday spending amounts because they are low.**

**This type of missing data is important to look for because you may lack data from key subgroups within your sample. Your sample may not end up being representative of your population.**

**Attrition bias**

**In longitudinal studies, attrition bias can be a form of MNAR data. Attrition bias means that some participants are more likely to drop out than others.**

**For example, in long-term medical studies, some participants may drop out because they become more and more unwell as the study continues. Their data are MNAR because their health outcomes are worse, so your final dataset may only include healthy individuals, and you miss out on important data.**

**3.2 ADD COLUMNS:n this article**

**Syntax**

**Return value**

**Remarks**

**Example**

**Adds calculated columns to the given table or table expression.**

**SyntaxADDCOLUMNS(<table>, <name>, <expression>[, <name>, <expression>]…)**

**Parameters**

**Term Definition**

**table Any DAX expression that returns a table of data.**

**name The name given to the column, enclosed in double quotes.**

**expression Any DAX expression that returns a scalar expression, evaluated for each row of table.**

**Return value**

**A table with all its original columns and the added ones.**

**Remarks**

**This function is not supported for use in DirectQuery mode when used in calculated columns or row-level security (RLS) rules.**

**Example**

**The following example returns an extended version of the Product Category table that includes total sales values from the reseller channel and the internet sales.**

**DAX**

**Copy**

**ADDCOLUMNS(ProductCategory,**

**, "Internet Sales", SUMX(RELATEDTABLE(InternetSales\_USD), InternetSales\_USD[SalesAmount\_USD])**

**, "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales\_USD), ResellerSales\_USD[SalesAmount\_USD]))**

**The following table shows a preview of the data as it would be received by any function expecting to receive a table:**

**ProductCategory[ProductCategoryName] ProductCategory[ProductCategoryAlternateKey] ProductCategory[ProductCategoryKey] [Internet Sales] [Reseller Sales]**

**Bikes 1 1 25107749.77 63084675.04**

**Components 2 2 11205837.96**

**Clothing 3 3 306157.5829 1669943.267**

**Accessories 4 4 640920.1338 534301.9888**

**3.3REMOVE COLUMNS:In this article**

**Syntax**

**About**

**Example 1**

**Syntax**

**Table.RemoveColumns(table as table, columns as any, optional missingField as nullable number) as table**

**About**

**Removes the specified columns from the table provided. If the specified column doesn't exist, an error is raised unless the optional parameter missingField specifies an alternative behavior (for example, MissingField.UseNull or MissingField.Ignore).**

**Example 1**

**Remove column [Phone] from the table.**

**Usage**

**Power Query M**

**Copy**

**Table.RemoveColumns(**

**Table.FromRecords({[CustomerID = 1, Name = "Bob", Phone = "123-4567"]}),**

**"Phone"**

**)**

**Output**

**Table.FromRecords({[CustomerID = 1, Name = "Bob"]})**

**Example 2**

**Try to remove a non-existent column from the table.**

**Usage**

**Power Query M**

**Copy**

**Table.RemoveColumns(**

**Table.FromRecords({[CustomerID = 1, Name = "Bob", Phone = "123-4567"]}),**

**"Address"**

**)**

**Output**

**[Expression.Error] The column 'Address' of the table wasn't found.**

**4. DATA VISULIZATION:**

**4.1.HISTOGRAM MOVIE RELEASIED:This content is subject to copyright. Terms and conditions apply.**

**Download**

**View publication**

**Histogram of movie release years**

**Histogram of movie release years**

**4.2 TOP 20 MOVIES BASED ON ITS PROFIT:Deep Throat (1972) ...**

**Facing the Giants (2006) ...**

**Paranormal Activity (2007) ...**

**Fireproof (2008) ...**

**The Texas Chainsaw Massacre (1974) ...**

**The Gallows (2015) ...**

**Eraserhead (1977) ...**

**An Inconvenient Truth (2006)**

**4.3 TOP 20 MOVIES BASED ON ITS RETURN ON INVESTEMENT:**

**Here I list films with large gross, relative to their budget -- effectively, how much the producers got per dollar invested. I find this to be a much more interesting figure than the absolute gain often listed, as this allows cheap films to rank highly -- and I do strongly admire filmmakers who are able to achieve much with very little. (Of course, this has become a strategy in recent years with all those found footage horror flicks.) And this has the advantage of not having to care for inflation.**

**4.4 Christopher Nolan. 3,239. 6 Films. Batman Begins 2. ...**

**Stanley Kubrick. 2,465. 8 Films. 2001: A Space Odyssey 2. ...**

**Alfred Hitchcock. 1,553. 9 Films. ...**

**Quentin Tarantino. 1,337. 5 Films. ...**

**Martin Scorsese. 1,261. 6 Films. ...**

**Steven Spielberg. 830. 6 Films. ...**

**Ingmar Bergman. 509. 4 Films. ...**

**Akira Kurosawa. 356. 5 Films.**

**More items...**

**5.DATA PRE\_ PROCESSING:**

**5.1 REMOVE NAMES:Secure your social media accounts or delete them entirely. ...**

**Scan for old posts, comments and reviews. ...**

**Delete your online shopping accounts. ...**

**Delete old email accounts. ...**

**Get help. ...**

**Remove outdated search results. ...**

**Hide bad stuff by flooding search engines**

**5.2 However if you want to remove the linearly dependent row vectors in the matrix then remove all the columns which do not contain pivot elements in the reduced row echelon form of the transpose of the given matrix.What is a linearly dependent variable?**

**correlation - What is the difference between linearly ...**

**Two variables are linearly dependent if one can be written as a linear function of the other. If two variable are linearly dependent the correlation between them is 1 or -1. Linearly correlated just means that two variables have a non-zero correlation but not necessarily having an exact linear relationship.**

**5.3 Table of Contents**

**Step 1 - Import the library**

**Step 2 - Setup the Data**

**Step 3 - Creating the Correlation matrix and Selecting the Upper triangular matrix**

**Step 5 - Droping the column with a high correlation**

**Step 6 - Analysing the output**

**Start your journey as a Data Scientist today with solved end-to-end Data Science Projects**

**Step 1 - Import the library**

**import pandas as pd import numpy as np from sklearn import datasets**

**We have imported numpy, pandas and datasets. We will use datasets to get the inbuilt iris dataset.**

**Step 2 - Setup the Data**

**We have used datasets to load the inbuilt iris dataset and created objects X and y to store the data and the target value, respectively. With the data in X, we have created a dataframe and printed the first five rows. iris = datasets.load\_iris() X = iris.data y = iris.target df = pd.DataFrame(X) print(df.head())**

**Step 3 - Creating the Correlation matrix and Selecting the Upper triangular matrix**

**So now we are creating a square matrix with dimensions equal to the number of features. We will have the elements as the absolute correlation value between the features. cor\_matrix = df.corr().abs() print(cor\_matrix)**

**Note that the Correlation matrix will be a mirror image of the diagonal, and all the diagonal elements will be 1. So, It does not matter that we select the upper or lower triangular part of the correlation matrix, but we should not include the diagonal elements. So we are selecting the upper traingular. upper\_tri = cor\_matrix.where(np.triu(np.ones(cor\_matrix.shape),k=1).astype(np.bool)) print(upper\_tri)**

**Step 5 - Droping the column with a high correlation**

**So we are selecting the columns with an absolute correlation greater than 0.95 and making a list of those columns named 'to\_drop'. to\_drop = [column for column in upper\_tri.columns if any(upper\_tri[column] > 0.95)] print(); print(to\_drop) Now we are dropping the columns which are in the list 'to\_drop' from the dataframe df1 = df.drop(df.columns[to\_drop], axis=1) print(); print(df1.head())**

**Step 6 - Analysing the output**

**Initially, there will be a dataframe with 4 columns in the output. Then there will be the correlation matrix where we can observe all diagonal elements as 1, and the upper and lower triangular is the mirror image. After that, the upper triangular matrix and the final dataframe with the highly correlated columns were removed.**

**0 1 2 3**

**0 5.1 3.5 1.4 0.2**

**1 4.9 3.0 1.4 0.2**

**2 4.7 3.2 1.3 0.2**

**3 4.6 3.1 1.5 0.2**

**4 5.0 3.6 1.4 0.2**

**0 1 2 3**

**0 1.000000 0.117570 0.871754 0.817941**

**1 0.117570 1.000000 0.428440 0.366126**

**2 0.871754 0.428440 1.000000 0.962865**

**3 0.817941 0.366126 0.962865 1.000000**

**0 1 2 3**

**0 NaN 0.11757 0.871754 0.817941**

**1 NaN NaN 0.428440 0.366126**

**2 NaN NaN NaN 0.962865**

**3 NaN NaN NaN NaN**

**[3]**

**6. IMPLEMEBNT ALGORITHM:**

**6.1 CLASSIFICATION TREE:A Classification tree labels, records, and assigns variables to discrete classes. A Classification tree can also provide a measure of confidence that the classification is correct. A Classification tree is built through a process known as binary recursive partitioning.**

**6.2 LOGISTIC REGRESSION:This type of statistical model (also known as logit model) is often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of**

**6.3 The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.:**

**6.4 SUPPORTVECTOR Support Vector Machine (SVM) is a powerful machine learning algorithm used for linear or nonlinear classification, regression, and even outlier detection tasks. SVMs can be used for a variety of tasks, such as text classification, image classification, spam detection, handwriting identification, gene expression analysis, face detection, and anomaly detection. SVMs are adaptable and efficient in a variety of applications because they can manage high-dimensional data and nonlinear relationships.**

**SVM algorithms are very effective as we try to find the maximum separating hyperplane between the different classes available in the target feature.**

**Support Vector Machine**

**Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well it’s best suited for classification. The main objective of the SVM algorithm is to find the optimal hyperplane in an N-dimensional space that can separate the data points in different classes in the feature space. The hyperplane tries that the margin between the closest points of different classes should be as maximum as possible. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.**

**Let’s consider two independent variables x1, x2, and one dependent variable which is either a blue circle or a red circle.**

**Linearly Separable Data points**

**From the figure above it’s very clear that there are multiple lines (our hyperplane here is a line because we are considering only two input features x1, x2) that segregate our data points or do a classification between red and blue circles. So how do we choose the best line or in general the best hyperplane that segregates our data points?**

**How does SVM work?**

**One reasonable choice as the best hyperplane is the one that represents the largest separation or margin between the two classes.**

**Multiple hyperplanes separating the data from two classes**

**Multiple hyperplanes separate the data from two classes**

**So we choose the hyperplane whose distance from it to the nearest data point on each side is maximized. If such a hyperplane exists it is known as the maximum-margin hyperplane/hard margin. So from the above figure, we choose L2. Let’s consider a scenario like shown below**

**Selecting hyperplane for data with outlier**

**Selecting hyperplane for data with outlier**

**Here we have one blue ball in the boundary of the red ball. So how does SVM classify the data? It’s simple! The blue ball in the boundary of red ones is an outlier of blue balls. The SVM algorithm has the characteristics to ignore the outlier and finds the best hyperplane that maximizes the margin. SVM is robust to outliers.**

**Hyperplane which is the most optimized one**

**Hyperplane which is the most optimized one**

**So in this type of data point what SVM does is, finds the maximum margin as done with previous data sets along with that it adds a penalty each time a point crosses the margin. So the margins in these types of cases are called soft margins. When there is a soft margin to the data set, the SVM tries to minimize (1/margin+∧(∑penalty)). Hinge loss is a commonly used penalty. If no violations no hinge loss.If violations hinge loss proportional to the distance of violation.**

**Till now, we were talking about linearly separable data(the group of blue balls and red balls are separable by a straight line/linear line). What to do if data are not linearly separable?**

**Original 1D dataset for classificationCLASSIFIER:**