DATA PREPROCESSING

1. The pandas library is used to load the dataset from the CSV file into a DataFrame.

2. Any necessary data cleaning steps can be performed. In this example, we remove unnecessary columns (e.g., 'PassengerId', 'Name', 'Ticket', 'Cabin') using the drop() function. Additionally, missing values in 'Age' and 'Embarked' columns are handled by filling them with the mean and mode values, respectively.

3. The dataset is split into features (X) and the target variable (y), where the 'Survived' column represents the target variable.

4. Label encoding is performed for categorical variables. Here, the LabelEncoder from scikit-learn is used to transform categorical values ('Sex' and 'Embarked') into numeric representations.

5. Feature scaling is applied to normalize the feature values. Here, the StandardScaler from scikit-learn is used to standardize the features.

6. The dataset is split into training and testing sets using the train_test_split() function from scikit-learn. In this example, 80% of the data is allocated to the training set and 20% to the testing set.

7. Finally, the shapes of the training and testing sets are printed to verify the sizes of the sets.

MANIPULATION OF TWITTER DATASET

1. Import the required modules, dataset and load the dataset.
2. Then drop a certain column from the dataset.
3. Drop a row from the dataset.
4. Then rename a column from the dataset.
5. Then group a data under a certain features.
6. Slicing the dataset to get required information.
7. Describe the dataset and print it.

EVALUATING ML ALGORITHMS

1. Import the required modules, dataset and load the dataset.
2. Import the necessary libraries and model
3. Import the model from sklearn.
4. The dataset is split into training and testing sets using the train_test_split() function from sklearn.
5. Use the predict() method from the model to examine our test dataset.
6. In this program, 75% of the data is allocated to the training set and 25% to the testing set.
7. Finally, the performance metrics are printed.

EX:4(a) Predicting Water Temperature based on Salinity using Regression

1. Import the Libraries, dataset and load the dataset.
2. Read the Dataset
3. Remove the unwanted data from the dataset
4. Splitting the data into training and testing data
5. Dropping any rows with Nan values.
6.Data scatter of predicted values Exploring our results

Ex-4(b) Implementation of Correlation Analysis of Iris and Boston Housing Datasets

1. **Import the necessary libraries**
2. **Load the Iris dataset using function from sklearn.datasets.**
3. **Convert the Iris data to a pandas DataFrame.**
4. **Calculate the correlation matrix for the Iris dataset.**
5. **Print the correlation matrix for the Iris dataset.**
6. **Load the Boston Housing dataset using function from sklearn.datasets.**
7. **Convert the Boston Housing data to a pandas DataFrame.**
8. **Calculate the correlation matrix for the Boston Housing dataset.**
9. **Print the correlation matrix for the Boston Housing dataset.**
10. **Calculate the pairwise correlation between the selected features in the Boston Housing dataset using the correlation function on the respective columns.**
11. **Print the calculated correlation value.**

SVM

1.Import the libraries.
2.Use read_csv() function of pandas library, which is used to read a csv file and performs various operations on it.
3.use iloc[ ] method of Pandas library, used to extract the required rows and columns from the dataset.
4.To create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future.
5. The SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a hyperplane.
6. The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features then hyperplane will be a straight line.
7. If there are 3 features, then hyperplane will be a 2-dimension plane.
8. To create the SVM classifier, import SVC class from Sklearn.svm library.
9.Predict the output for test set. For this, will create a new vector y_pred.
10. From the data set, the SVM classifier has divided the users into two regions (Purchased or Not purchased).
11. Users who purchased the SUV are in the red region with the red scatter points. And users who did not purchase the SUV are in the green region with green scatter points.
12.The hyperplane has divided the two classes into Purchased and not purchased variable.

EX:6 – NoSQL using MongoDB

1.Install MongoDB server from the link Download MongoDB Community Server | MongoDB
2. Install .msi file
3. Install mongodb shell : MongoDB Shell Download | MongoDB
4. Save it in C:
5. Set the path
a. Goto control panel - > system properties -> environmental variable
7. Show database using showdbs
8. Use database using use command
9. Create a school database
10. insert documents into school database
11. Display the documents in the database
12. Insert many documents in the collection
13. Sort the document in alphabetical order. 1-alphabetical order and -1 for reverse order
14. update one or many documents
15. Delete one or many documents

7(a) : MapReduce application for word counting on Hadoop cluster.

Step 1: Start Cloudera and open Eclipse. Create a new Java Project. Enter the project name and Check the
default output folder and Click Next
Step 2: Add external jars to compile the code. Click on Libraries Tab and "Add External Jars". Right Select
all the jars present in folder /usr/lib/hadoop/client , /usr/lib/Hadoop, /usr/lib/hadoop/lib.
Step 3: Select File system and click on usr/lib/Hadoop/lib. Select all Jar files and click on Ok and Finish.
Step 4: Project is created with name Word Count. Right Click on the project new Class.
Step 5 : Create a jar file of the program. Right click on project select "Export" and then click on "Jar File" under
Java folder. Give the location path where you want to store your .jar file.
Step 6: Write the java program using Map, Reduce and Driver methods and save it.
Step7:Create Input file. Open the terminal and create a input file which is a huge text file.
$vim input.txt
Step 8 : Make a new file directory on HDFS (Hadoop Distributed File System)
Step 9: Copy this file on the NameNode i.e., on HDFS $ hdfs dfs –copyFromLocal input.txt
Step 10: Run the program using the hadoop command. Open new terminal type cd Desktop press enter and
type $ hadoop jar WordCount.jar WordCount /WordCountFile /output2. Map Reduce program starts
to run. We can see the percentage of mapping and reducing theprogram is doing on the command line.

HDFS

1. Install Java 8 if not already installed.
2. Download Hadoop 3.3.1.
3. Extract the downloaded Hadoop archive.
4. Set environment variables for Java and Hadoop paths.
5. Configure HDFS by updating core-site.xml and hdfs-site.xml.
6. Format HDFS using hdfs namenode -format.
7. Verify HDFS is running using jps command.

YARN

1. We import the `subprocess` module to execute Yarn commands from within Python.
2. We define the name of the JavaScript package we want to install, in this case, "axios."
3. We construct the Yarn command using an f-string, which includes the `yarn add` command followed by the package name.
4. We use `subprocess.run()` to run the Yarn command. The `shell=True` argument allows us to run the command in a shell environment, and `check=True` ensures that an exception is raised if the Yarn command fails.
5. We handle any errors that may occur during the Yarn command execution and print a success message if the package installation is successful.
You can customize this script by changing the `package_name` variable and the Yarn command to install the specific JavaScript package you need. Make sure you have Yarn and Node.js installed in your environment before running this script.

PIG

Step 1: Install OpenJDK 8
Step 2: Download and Extract Apache Pig
Step 3: Set Environment Variables
Step 4: Define the Pig Script
Step 5: Save the Pig Script to a File
Step 6: Specify Input Data Path
Step 7: Construct the Pig Command
Step 8: Run the Pig Script
Step 9: Check Exit Status

HIVE

1. Update Package Repositories
□ Run the command to refresh the list of available packages.
2. Install Hive and Dependencies
□ Use the package manager to install Hive, Hive Metastore, and HiveServer2.
□ The -y flag is used to automatically confirm installation prompts.
3. Start Hive Metastore and HiveServer2
□ Start the Hive Metastore service using the hive --service metastore command.
□ Start the HiveServer2 service using the hive --service hiveserver2 command.
□ Use nohup to run these services in the background and suppress output.
4. Check if Hive is Installed and Running
□ Define a Python function, check_hive_status(), to determine if Hive is

correctly installed and operational.
▯ Execute a Hive query (SHOW DATABASES;) using subprocess.run().
▯ Check for successful execution or a FileNotFoundError.
5. Call the Function
▯ Invoke the check_hive_status() function to perform the Hive status check

TABLEAU

Tableau is a powerful data visualization tool that allows you to connect to various data sources, create interactive visualizations, and generate insightful reports and dashboards. To use Tableau for visualization with given datasets, follow these general steps:

1.Install Tableau:
Download and install Tableau Desktop from the official Tableau website. You might need a license or a trial version for this.

2.Load Data:
Open Tableau Desktop and connect to your dataset. Tableau supports a wide range of data sources including Excel, CSV, databases, and more. To load your data:

Click on "Connect to Data" on the start page.
Choose the appropriate data source and select your dataset file.
Follow the prompts to connect and load your data.

3.Create Visualizations:
After loading the data, you can create various types of visualizations:
Drag and drop fields from your dataset to the Columns and Rows shelves to create charts like bar charts, line charts, scatter plots, etc.
Use the "Show Me" feature to get recommendations on suitable visualization types based on your selected fields.

4.Change Chart Type :
Tableau might automatically create a bar chart or other visualization based on the initial fields you dragged. To change it to a required chart:
Click on the "Show Me" icon in the top-right corner (looks like a chart symbol).
In the "Show Me" panel, find the chart type option and click on it.

5.Adjust Labels and Colors:
By default, Tableau will assign colors to each category and add labels. You can adjust these settings:
Click on the "Label" button in the Marks card to show or hide labels.
Click on the "Color" button in the Marks card to change the color scheme.

Format any of the Chart:
You can format of the chart using the "Format" menu in the top bar. Here, you can adjust font sizes, colors, borders, and other properties.

6.Add Labels and Legends:
If labels aren't automatically shown, you can add them by clicking on "Label" in the Marks card and selecting the appropriate options.
To add a legend, click on "Color" in the Marks card and select "Edit Colors." In the Edit Colors dialog, check the "Show at the Bottom" option.

7.Save and Share:
Once you're satisfied with your chart, you can save it and share it with others. You can also publish it to Tableau Server, Tableau Online, or Tableau Public.