Table of Contents

Activity	Page No
1.Python Programming	
1.1. Create a simple calculator in Python.	**1 – 1**
1.2 An electric power distribution company charges domestic customers as follows: Consumption unit Rate of charge:	
1.2.1. 0-200 Rs. 0.50 per unit	
1.2.2. 201-400 Rs. 0.65 per unit in excess of 200	
1.2.3. 401-600 Rs 0.80 per unit excess of 400	
1.2.4. 601 and above Rs 1.00per unit excess of 600	
1.2.5. If the bill exceeds Rs. 400, then a surcharge of 15% will be charged, and the minimum bill should be Rs. 100/-	
Create a Python program based on the scenario mentioned above.	**2 – 3**
1.3. Create a Python program based on the numbers using for loops.	**4 – 4**
1.4. Write a program to find the number and sum of all integers greater than 100 and less than 200 that are divisible by 7.	**5 – 5**
1.5. Write a recursive function to calculate the sum of numbers from 0 to 10	**6 – 6**
1.6. Write a Python program to reverse the digits of a given number and add them to the original. If the sum is not a palindrome, repeat this procedure.	**7 – 7**
1.7. Write a menu-driven program that performs the following operations on strings	
1.7.1. Check if the String is a Substring of Another String	
1.7.2. Count Occurrences of Character	
1.7.3. Replace a substring with another substring	**8-10**
1.7.4. Convert to Capital Letters	

1.8. Write a function to find the factorial of a number but also store the factorials calculated in a dictionary. **11 – 11**

1.9. Perform various set operations

1.9.1. Set Union

1.9.2. Set Intersection

1.9.3. Set Difference **12 – 12**

1.10. Create a dictionary to store the name, roll_no, and total_mark of N students. Now print the details of the student with the highest total_mark. **13 – 13**

1.11. Write a Python program to copy the contents of a file into another file, line by line. **14 – 14**

1.12. Use the OS module to perform

1.12.1. Create a directory

1.12.2. Directory Listing

1.12.3. Search for ".py" files

1.12.4. Remove a particular file **15 – 16**

1.13 Create a simple banking application by using inheritance **17 – 19**

2. Database Operations

2.1. Implementation of MySQL connection using Python **20 – 23**

2.2. Implementation of SqLite3 connection using Python **24 – 26**

2.3. Write a program to implement CRUD operations using Python **27 – 29**

3. Common Gateway Interface

3.1. Create a Login form using CGI and display the input on a different page. **30 – 32**

3.2. Create a registration form for MCA admission and display the inserted data on the web page. **33 – 35**

3.3. Create a MySQL database and perform INSERT, UPDATE, DESTROY, and SELECT (display) operations using the CGI interface. **36 – 40**

4. Machine Learning Libraries in Python.

4.1. NumPy

4.1.1. Create a numpy array filled with all ones by defining its shape. **41 – 41**

4.1.2. How do you remove rows from a Numpy array that contains non-numeric values? **42 - 42**

4.1.3. Remove single-dimensional entries from the shape of an array **43 – 43**

4.1.4. How do you check whether specified values are present in the NumPy array? **44 – 44**

4.1.5. Write a program to get all 2D diagonals of a 3D NumPy array? **45 – 46**

4.1.6. Write a NumPy program to sort a given array of shape 2 along the first axis, last axis, and flattened array. **47 – 47**

4.1.7. Write a NumPy program to create a structured array from a given student name, height, class, and data type. Now sort by class, then height if the classes are equal. **48 – 48**

4.1.8. Write a NumPy program to sort a given complex array using the real part first, then the imaginary part. **49 – 49**

4.1.9. Write a NumPy program to sort a given array by the n th column. **50 -50**

4.1.10. Calculate the sum of the diagonal elements of a NumPy array **51 -51**

4.1.11. Write a program for Matrix Multiplication in NumPy **52 -52**

4.1.12. Multiply matrices of complex numbers using NumPy in Python **53 -53**

4.1.13. Calculate the inner, outer, and cross products of matrices and vectors using NumPy. **54 – 55**

4.1.14. Compute the covariance matrix of two given NumPy arrays. **56 – 56**

4.1.15. Convert covariance matrix to correlation matrix using Python **57 – 57**

4.1.16. Write a NumPy program to compute the histogram of nums against the bins. **58 – 58**

4.1.17. Write a NumPy program to compute the cross-correlation of two given arrays **59 – 59**

4.1.18. Write a NumPy program to compute the mean, standard deviation, and variance of a given array along the second axis. **60 – 60**

4.1.19. Write a NumPy program to compute the 80th percentile for all elements in a given array along the second axis. **61 - 61**

4.2. Pandas

4.2.1. Write a Pandas program to add, subtract, multiply, and divide two Pandas Series. **62 – 63**

4.2.2. Write a Pandas program to convert a dictionary to a Pandas series. **64 -64**

4.2.3. Write a Pandas program to convert the first column of a data frame into a Series **65 – 65**

4.2.4. Write a Pandas program to convert a Series of lists into one Series. **66 – 66**

4.2.5. Write a Pandas program to create a subset of a given series based on value and condition **67 – 67**

4.2.6. Write a Pandas program to get the items that are not common in two given series. **68 – 68**

4.2.7. Write a Pandas program to calculate the frequency counts of each unique value of a given series **69 – 69**

4.2.8. Write a Pandas program to filter words from a given series that contain at least two vowelsu. **70 – 70**

4.2.9. Write a Pandas program to find the index of the first occurrence of the smallest and largest values of a given series. **71 – 71**

4.2.10. Write a Pandas program to get the first 3 rows of a given data frame. **72 - 72**

4.2.11. Write a Pandas program to select the 'name' and 'score' columns from a data frame. **73 - 73**

4.2.12. Write a Pandas program to count the number of rows and columns in a data frame. **74 – 74**

4.2.13. Write a Pandas program to add one row to an existing data frame **75 -75**

4.2.14. Write a Pandas program to write a data frame to a CSV file using a tab separator. **76 – 76**

4.2.15. Write a Pandas program to replace all the NaN values with Zeros in a column of a data frame. Write a Pandas program to drop a list of rows from a specified data frame. **77 – 78**

4.2.16. Write a Pandas program to shuffle a given data frame row. **79 – 79**

4.2.17. Write a Pandas program to find the row where the value of a given column is maximum. **80 – 80**

4.2.18. Write a Pandas program to check whether a given column is present in a data frame or not. **81 – 81**

4.2.19. Write a Pandas program to append data to an empty data frame. **82 – 82**

4.2.20. Write a Pandas program to convert continuous values of a column in a given data frame to categorical. Input: { 'Name': ['Alberto Franco','Gino Mcneill','Ryan Parkes', 'Eesha Hinton', 'Syed Wharton'], 'Age': [18, 22, 40, 50, 80, 5] } **83 – 83**

4.2.21. Write a Pandas program to create data frames that contain random values, missing values, datetime values, and mixed values. **84 – 86**

4.2.22. Write a Pandas program to join the two given data frames along rows and assign all the data. student_data1: student_id name marks 0 s1 Danniella Fenton 200 1 s2 Ryder Storey 210 2 s3 Bryce Jensen 190 3 s4 Ed Bernal 222 4 s5 Kwame Morin 199 student_data2: student_id name marks 0 s4 Scarlette Fisher 201 1 s5 Carla Williamson 200 2 s6 Dante Morse 198 3 s7 Kaiser William 219 4 s8 Madeeha Preston 201 **87 – 88**

4.2.23. Write a Pandas program to join the two given data frames along columns and assign all the data. (Use the same dataset as above.) **89 – 90**

4.2.24. Write a Pandas program to join the two given data frames along rows and merge with another data frame along the common column id. exam_data: student_id exam_id 0 S1 23 1 S2 45 2 S3 12 3 S4 67 4 S5 21 5 S7 55 6 S8 33 7 S9 14 8 S10 56 9 S11 83 10 S12 88 11 S13 12 (Add this data to the above dataset.) **91 – 92**

4.2.25. Write a Pandas program to join the two data frames with matching records from both sides, where available. (Same dataset as above.) **93 – 94**

4.3. Pandas Grouping

4.3.1. Write a Pandas program to split the following data frame into groups based on school code. Also, check the type of GroupBy object. **95 – 96**

4.3.2. Write a Pandas program to split the following data frame by school code and get the mean, min, and max values of age for each school. (Use the above dataset.) **97 - 98**

4.3.3. Write a Pandas program to split the following data frame into groups based on all columns and calculate groupby value counts on the data frame. Test Data: Id type book 0 1 10 Math 1 2 15 English 2 1 11 Physics 3 1 20 Math 4 2 21 English 5 1 12 Physics 6 2 14 English schoo l class name date_Of_Birth age height weight address S1 s001 V Alberto Franco 15/05/2002 12 173 35 street1 S2 s002 V Gino Mcneill 17/05/2002 12 192 32 street2 S3 s003 VI Ryan Parkes 16/02/1999 13 186 33 street3 S4 s001 VI Eesha Hinton 25/09/1998 13 167 30 street1 S5 s002 V Gino Mcneill 11/05/2002 14 151 31 street2 S6 s004 VI David Parkes 15/09/1997 12 159 32 street4 **99 – 99**

4.3.4. Write a Pandas program to split the following data frame into groups by school code and get the mean, min, and max values of age with customized column names for each school **100 – 101**

4.4. Matplotlib

4.4.1. Visualize the following using the given dataset (alphabet_stock_data.csv),

4.4.1.1. Create a line plot of the historical stock prices of Alphabet Inc. between two specific dates. **102 – 103**

4.4.1.2. Create a bar plot of the trading volume of Alphabet Inc. stock between two specific dates. **103 – 105**

4.4.1.3. Create a stacked histogram plot with more bins of opening, closing, high, and low stock prices of Alphabet Inc. between two specific dates. **105 - 107**

4.4.1.4. Create a scatter plot of the trading volume/stock prices of Alphabet Inc. stock between two specific dates. **107 - 108**

4.4.2. Write a Python program to draw a line with a suitable label on the x-axis, y-axis, and title. **109 - 110**

4.4.3. Write a Python program to draw line charts of the financial data of Alphabet Inc. between October 3, 2016, and October 7, 2016. Date,Open,High,Low,Close 10-03-16,774.25,776.065002,769.5,772.559998 10-04-16,776.030029,778.710022,772.890015,776.429993 10-05-16,779.309998,782.070007,775.650024,776.469971 10-06-16,779,780.47998,775.539978,776.859985 10-07-16,779.659973,779.659973,770.75,775.080017 **111 – 113**

4.4.4. Write a Python program to draw a line with a suitable label on the x-axis, and y-axis and a title. Create the code snippet that gives the output shown in the following screenshot: **114 – 115**

4.4.5. Write a Python program to display the grid and draw line charts of the closing value of Alphabet Inc. between October 3, 2016, and October 7, 2016. Customized the grid lines with linestyle -, width 0.5, and color blue. Date,Close 03-10-16,772.559998 04-10-16,776.429993 05-10-16,776.469971 06-10-16,776.859985 07-10-16,775.080017 **116 – 117**

4.4.6. Write a Python program to create multiple plots as in the screenshot (use any method). **118 – 119**

4.4.7. Write a Python program to create a bar plot from a data frame. Sample Data Frame: s a b c d e f 2 4,8,5,7,6 4 2,3,4,2,6 6 4,7,4,7,8 8 2,6,4,8,6 10 2,4,3,3,2 Create the code snippet which gives the output shown in the following screenshot: **120 – 121**

4.4.8. Write a Python program to create a stacked bar plot with error bars. Note: Use the bottom to stack the women's bars on top of the men's bars. Sample Data: Means (men) = (22, 30, 35, 35, 26) Means (women) = (25, 32, 30, 35, 29) Men's Standard deviation = (4, 3, 4, 1, 5) Women's Standard deviation = (3, 5, 2, 3, 3) Create the code snippet that gives the output shown in the following screenshot: **122 - 124**

4.4.9. Write a Python program to create stack bar plot and add labels to each section. Sample data: people = ('G1','G2','G3','G4','G5','G6','G7','G8') segments = 4 # multi-dimensional data data = [[3.40022085, 7.70632498, 6.4097905, 10.51648577, 7.5330039, 7.1123587, 12.77792868, 3.44773477], [11.24811149, 5.03778215, 6.65808464, 12.32220677, 7.45964195, 6.79685302, 7.24578743, 3.69371847], [3.94253354, 4.74763549, 11.73529246, 4.6465543, 12.9952182, 4.63832778, 11.16849999, 8.56883433], [4.24409799, 12.71746612, 11.3772169, 9.00514257, 10.47084185, 10.97567589, 3.98287652, 8.80552122]] Create the code snippet that gives the output shown in the following screenshot: **125 – 127**

4.4.10. Write a Python program to add textures (black and white) to bars and wedges. Note: Use the bottom to stack the women's bars on top of the men's bars. Create the code snippet that gives the output shown in the following screenshot **128 – 129**

5. Data Analytics using Python

5.1. Handle the given dataset (Data.csv) with adequate preprocessing steps mentioned and visualize the dataset with appropriate graphs.

5.1.1. Handle Missing Data Values

5.1.2. Encode the categorical data

5.1.3. Scale your features **130 – 134**

5.2. Using the given dataset (dirtydata.csv),

5.2.1. Handle the data with empty cells (Use dropna() and fillna())

5.2.2. Replace the empty cells using mean, median, and mode.

5.2.3. Handle the data in the wrong format.

5.2.4. Handle the wrong data from the dataset.

5.2.5. Discover and remove duplicates. **135 - 138**

5.3. Create a cricketer dataset using a dictionary of lists, and create a new attribute 'Experience Category' using 'Age' as the binning factor. **139 – 140**

5.4. car_age = [5, 7, 8, 7, 2, 17, 2, 9, 4, 11, 12, 9, 6]
car_speed = [99,86,87,88,111,86,103,87,94,78,77,85,86]
Using the given dataset,

5.4.1. Draw the line of linear regression

5.4.2. Evaluate how well the data fit in linear regression.

5.4.3. Predict the speed of a 10-year-old car. **141 – 143**

5.5. Using the dataset (cars.csv),

5.5.1. Predict the CO2 emissions of a car with a weight of 2300 kg and volume of 1300 cm3.

5.5.2. Print the coefficient values of the regression object. **144 – 145**

5.6. Using the insurance dataset (insurance.csv) with adequate preprocessing steps,

5.6.1. Visualize the correlation among variables using a heatmap.

5.6.2. Create a linear regression model.

5.6.3. Evaluate the model. (Find MSE and R_square.)

5.6.4. Predict the charges for a person with an age of 30, a BMI of 32.00, and who is a smoker. **146 – 152**

5.7. Evaluate the dataset (User_Data.csv) and predict whether a user will purchase the company's product or not. (Use logistic regression.) **153 – 155**

5.8. Use the Iris dataset to visualize a decision tree with a depth=4 and save the plot as a PNG file. Also, print the confusion matrix and generate the classification report. **156 - 158**

5.9. Use the KNN algorithm to train the model and predict the future using the Iris dataset. Also, measure the accuracy of the model. **159 – 161**

5.10. Analyze the given dataset (gym_data.csv) using RandomForestRegressor and visualize the 'Effect of n_estimators. **162 – 164**

5.11. Visualize a 3-dimensional cluster using the given dataset 'Mall_Customers.csv', where no_of_clusters = 5. **165 – 167**

5.12. Using the dataset provided (Online Retail.xlsx),

5.12.1. Split the data according to the region of the transaction.

5.12.2. Build the models using the apriori algorithm.

5.12.3. Develop the association rules.

5.12.4. Find the most frequent items in any one of the regions. **168 – 169**

6. Project **170 – 178**