# 20AD32P2 - DATA SCIENCE LABORATORY

|  |  |  |  |
| --- | --- | --- | --- |
| Course Category: | Professional Core | Credits: | 1.5 |
| Course Type: | Practical | Lecture-Tutorial-Practical: | 0-0-3 |
| Prerequisite: | A Course on - Foundations of Data Science | Sessional Evaluation:  Univ. Exam Evaluation:  Total Marks: | 40  60  100 |
| Objectives: | * To impart knowledge on data manipulation and exploratory data analysis concepts that is vital for data science. * To develop skills for applying python tools and techniques to analyze, visualize and interpret data. | | |

|  |  |  |
| --- | --- | --- |
| Course Outcomes | Upon successful completion of the course, the students will be able to: | |
| CO1 | Demonstrate efficient storage and data handling methods in NumPy to perform mathematical computations vital for data science |
| CO2 | Apply Data Preparation and Exploration methods using Pandas to gain insights about raw data and transform quality data to perform analysis. |
| CO3 | Create data visualization using charts, plots and histograms to identify trends, patterns and outliers in data importing Matplotlib and Seaborn. |
| CO4 | Develop methods to analyse and interpret time series data to extract meaningful statistics. |
| Course Content | LIST OF EXPERIMENTS   1. **Array Computations using NumPy**    1. Perform arithmetic operations using array.    2. Perform slicing and indexing on multi-dimensional arrays.    3. Perform computations on multi-dimensional array using universal functions (ufunc).    4. Compute arithmetic mean, standard deviation, variance, percentile, minimum and maximum, cumulative sum and product using statistical functions in NumPy.    5. Perform set theory operations such as union, intersection, symmetric difference and fetching unique values. 2. **Linear Algebra and Random Number generation using linalg and random module in NumPy**    1. Compute dot product, vector product and inner product of two arrays.    2. Perform matrix operations such as multiplication, determinant, sum of diagonal elements and inverse.    3. Compute eigenvalues, eigenvectors and singular value decomposition for a square matrix.    4. Generate random samples from uniform, normal, binomial, chi-square and Gaussian distributions using numpy.random functions.    5. Implement a single random walk with 1000 steps using random module and extract the statistics like minimum and maximum value along the walk‘s trajectory. 3. **Data Manipulation using pandas**    1. Create Data Frame from List, Dict, List of Dicts, Dicts of Series and perform operations such as column selection, addition, deletion and row selection, addition and deletion.    2. Create a Data Frame and perform descriptive statistics functions such as sum, mean, median, mode, standard deviation, skewness, kurtosis, cumulative sum, cumulative product and percent changes.    3. Implement the computation of correlation and covariance by considering the Data Frames of stock prices and volumes obtained from Yahoo Finance! Using pandas-data reader package. 4. **Working with different data formats using pandas**    1. Perform reading and writing data in text format using read\_csv and read\_table considering any online dataset in delimited format (CSV).    2. Perform reading, writing and parsing data in JSON (Javascript Object Notation) format using read\_json.    3. Perform reading and writing of Microsoft Excel Files (xslx) using read\_excel. 5. **Interacting with Web APIs and Databases**    1. Predict the last 30 GitHub issues for pandas using request and response object‘s json method. Move the extracted data to DataFrame and extract fields of interest. (Use url: 'https://api.github.com/repos/pandas-dev/pandas/issues')    2. Connect to any relational database using corresponding SQL drivers and perform operations such as table creation, populating the table, selecting data from table, moving data from table to DataFrame, updating records and deleting records in a table. 6. **Data Cleaning and Preparation**    1. Perform data cleaning by creating a DataFrame and identifying missing data using NA(Not Available) handling methods, filter out missing data using dropna function, fill the missing data using fillna function and remove duplicates using duplicated and drop\_duplicates functions.    2. Perform data transformation by modifying set of values using map and replace method and create transformed version of original dataset without modification using rename method.    3. Create a DataFrame with normally distributed data using random sampling and detect possible outliers.    4. Perform text manipulation with regular expression by applying relevant regular expression methods to split a string with a variable number of whitespace characters (tabs, spaces, and newlines) and get a list of all patterns matching. 7. **Data Wrangling**    1. Perform hierarchical indexing by creating a series with a list of lists (or arrays) as the index, select subsets of data at outer and inner levels using partial indexing.    2. Rearrange the tabular data with hierarchical indexing using unstack and stack method.    3. Create two different DataFrames and merge them using index as merge key and combine data with overlap using combine\_first method. 8. **Perform Data Visualization with Matplotlib and SeaBorn considering online dataset for processing.**    1. Create a Line Plot by setting the title, axis labels, ticks, ticklabels , annotations on subplots and save to a file.    2. Create Bar Plots using Series and DataFrame index.       1. Create bar plots with a DataFrame to group the values in each row together in a group in bars side by side for each value.       2. Create stacked bar plots from a DataFrame.    3. Create Histogram to display the value frequency and Density Plot to generate continuous probability distribution function for observed data.    4. Create Scatter Plot and examine the relationship between two one-dimensional data series.    5. Create Box plots to visualize data with many categorical variables. 9. **Time Series Analysis**    1. Create time series using datetime object in pandas indexed by timestamps.    2. Use pandas.date\_range to generate a DatetimeIndex with an indicated length.    3. Generate data ranges by setting time zone, localize time zone and convert to particular time zone using tz\_convert and combine two different time zones.    4. Perform period arithmetic such as adding and subtracting integers from periods and construct range of periods using period\_range function.    5. Convert Periods and PeriodIndex objects to another frequency with asfreq method.    6. Convert Series and DataFrame objects indexed by timestamps to periods with the to\_period method.    7. Perform resampling, downsampling and upsampling for the time series. 10. **Data Aggregation**     1. Create a tabular dataset as a DataFrame and split data into groups using groupby method including single key and multiple key values. Select group by considering single and multiple columns.     2. Compute summary statistics such as sum, mean and standard deviation for the grouped data using aggregate method.     3. Use groupby function to split data into groups based on one column, miultiple columns, compute summary statistics and perform exploratory data analysis. Consider any online dataset for processing. | |
| Text Books | **TEXT BOOKS:**   1. Wes McKinney, Python for Data Analysis, O‘Reilly,2nd Edition,2017. 2. Sinan Ozdemir, Principles of Data Science, Packt Publishers, 2nd Edition,2018. 3. Rachel Schutt, Cathy O‘Neil, Doing Data Science: Straight Talk from the Frontline, O‘Reilly, 2014. | |
| E-Resources | 1. <https://swayam.gov.in/nd1_noc19_cs60/preview> 2. <https://towardsdatascience.com/> 3. <https://www.w3schools.com/datascience/> 4. <https://github.com/jakevdp/PythonDataScienceHandbook> 5. [https://www.kaggle.com](https://www.kaggle.com/) | |
| SOFTWARE / Tools | 1. Python 3.8 2. Python Libraries – NumPy, Pandas, Matplotlib, Seaborn, Beautiful Soup, Vader 3. Anaconda Framework | |