20DS505
A.Y:2022-23

Bapatla Engineering College::Bapatla(Autonomous)
Department of Cyber Security and Data Science
III/IV B.Tech (Regular) DEGREE EXAMINATION
Scheme of Evaluation

February,2023 **Data Science**
Fifth Semester **Data Handling & Visualization**
Time: Three Hours **Maximum:** 70 Marks

Answer Question No. 1 Compulsorily. (14X1 = 14 Marks)
*Answer **ANY ONE** question from each Unit.* (4X14=56 Marks)

1. a) What effect does scaling have in a visual representation of data?
   Focus on a segment of visualization
   b) Explain what should be done with suspected or missing data?
   Either filled with legal values or dropped
   c) How can color of an object be defined?
   col attribute
   d) When will you use a histogram and when will you use a box plot?
   Histogram for frequency distribution of a series of values.
   Box plot gives five number summary of a series of values
   e) Explain what an outlier is?
   The data point which does not belong to the series of data.
   f) For what type of data is scatter plot usually used for?
   Continous data
   g) What is a dirty data record in context of data wrangling?
   A row of data Faulty/missing values
   h) How can we create a copy of the series in Pandas?
   `s2 = s1.copy()`
   i) How to get frequency counts of unique items of a series?
   `value_counts()` method
   j) How can we convert data Frame into an excel file?
   `df.to_excel('Courses.xlsx')`
   k) Which library would you prefer for plotting in Python language: Seaborn or Matplotlib?
   Seaborn
   l) What is the use of backward fill?
   Replace null values with the previous value in the column
   m) What is the Seaborn function for colouring plots?
   `sns.color_palette()`
   hue parameter
   n) Illustrate the 2 types of datasets?
   Wide dataset
   Long dataset

Unit -I

2. a) Explain the characteristics of structured, semi-structured and unstructured data with examples

Data that conforms to a pre-defined schema or structure we say it is structured data. Most of the structured data is stored in RDBMS. Some examples are Oracle, MySql, PostgreSQL and Microsoft SQL RDBMS.
Relation/Table – Domain constraints, Key constraints, Entity integrity and Referential integrity constraints.
Spread sheets, Personal RDBMS like Microsoft Access are other sources of structured data.
CRUD operations, Indexing, security and scalable at enterprise level.

Semi-structured data is also referred to as self-describing structure. It does not confirm to the data models that one typically associates with relational data bases or any other form of data tables. It uses tags to segregate semantic elements. Tags are also used to enforce hierarchies of records and fields within data.

Unstructured data does not conform to any data model. Its structure is quite unpredictable Human generated – social media comments, emails, word processing, PowerPoint presentations etc. Machine generated – satellite images, scientific data, surveillance images and videos etc.
Processing of unstructured data includes Association rule mining, Regression analysis, Collaborative filtering, Text Analytics or Text Mining, Natural Language Processing (NLP), Noisy Text Analytics, Manual Tagging with Metadata, PoS.

b) How do Gestalt principles influence the visual perception?

Gestalt principles are methods by which we organize the world so that it's familiar, makes sense, and is easy to process. They help in how you put together a graph or a dashboard. Gestalt's principles of design can transform and elevate a design from disorganized, messy, and jumbled to organized, seamless, and tidy design.
The following are the Gestalt's principles of design.
Figure/Ground: Perception from background or foreground.

Similarity: Humans relate similar things in shape and colour.
Continuation: Elements in a line or curve are perceived to be more related.
Closure: Humans build complete shapes from incomplete visual elements.
Proximity: Elements close to each other are related
Symmetry and Order/pragnanz: when you're presented with a set of ambiguous or complex objects, human brain will make them appear as simple as possible. Ex. Line emojis

(OR)

3. a) How does an info graphic differ from a data visualization?

Infographic: Tells us a story. It requires more efforts to make it more impactful and aesthetically pleasing. It is simple in content. It mandates great deal of work in design and make it visually appealing. It can be made interactive. Includes graphics and art.
Data Visualization: They are brief, discreet and be a part of Infographic. It does not entail great design effort. The complexity depends on the nature of chart used for visualization. Requires less time. It can be made highly interactive from data persepective. No room for art work.

b) Explain the benefits of Data Visualization.

Understanding the story, Exploring buiseness insights, Ease of data analysis, Making sense of complicated data, Quick action, Faster decision making, Demystifying patterns, Seamless data presentation, Finding errors, Interpreting correlation in relationships, Analysing and predicting trends, Easy spotting of outliers.

Unit -II

4. a) Code the different ways in which data can be read from different file types, Dictionary objects and Series objects into a DataFrame.

```
pd.read_json('posts.json')
pd.read_table('pokemon_data.txt',delimiter='\t')
pd.read_excel('pokemon_data.xlsx')
pd.read_csv('pokemon_data.csv')

dat = {
    'titles':['DHandV','DM','CN','SE','ATFL','ITK'],
    'codes':[501,502,503,504,505,506],
    'credits':[4,4,3,4,4,0]
}
df = pd.DataFrame(dat,index=dat['titles'])

l = list('abcdefg')
s1 = pd.Series(l)
s1.name= 'col1'
s2 = pd.Series(list('hijklmn'))
s2.name = 'col2'
df = pd.concat([s1,s2],axis=1)
```

b) Create a sample XML file and read the data from the file into a DataFrame.

```
import xml.etree.ElementTree as ETree
xmldata = 'xmltopandas.xml'
prstree = ETree.parse(xmldata)
root = prstree.getroot()
for book in root.iter('Book'):
    sno = book.attrib.get('slNo')
    aname = book.find('Author').text
    title = book.find('Title').text
    pub = book.find('Publisher').text
    yop = book.find('Year').text
    book_desc = [sno,aname,title,pub,yop]
    all_books.append(book_desc)
df = pd.DataFrame(all_books,columns=['SNo','Author','Title','Publisher','Year'])
```

(OR)

5. a) Illustrate the difference between count histogram, relative frequency histogram, cumulative frequency histogram and density histogram?

The count/frequency histogram shows the count of how many data values fall into a certain class wherein classes with greater frequencies have higher bars and classes with lesser frequencies have lower bars
A relative frequency histogram uses the same information as a frequency histogram but compares each class interval to the total number of items.
A Cumulative frequency histogram uses the same information as a frequency histogram but adds the sum of frequencies of previous bins to the frequency of the current bin.
Density plots can be thought of as plots of smoothed histograms and are drawn by computing kernel density.

b) What is a JSON? How can we read the JSON Data into a Pandas DataFrame?
Java Script Object Notation

```
[
    {"name":"Ram", "email":"Ram@gmail.com"},
    {"name":"Bob", "email":"bob32@gmail.com"}
```

```
]
json_df = pd.read_json('posts.json')
```

Unit -III

6. a) Create a sample DataFrame. Explain four different methods of slicing data from the DataFrame with example code, output and description.

```
df[['Name','HP']]
df.loc[[0,5],['Name']]
df.iloc[0:3,[3,4]]
df.loc[csv_df['HP'] > 100,['Name']]
```

b) Create a DataFrame with missing values. Write code to analyse, drop null values and fill the null values with different functions and combination of parameters. List the data after each operation

```
df.isnull()
df.dropna(how='any',axis=1)
df.dropna(how='all',axis=0)
df.dropna(thresh=3,axis=1)
df.fillna(10)
df.fillna(method='bfill')
df.fillna(method='ffill')
```

(OR)

7. a) A Superstore Sales data DataFrame consists of OrderID, Region, SalesAmt columns. Code a snippet without using groupby() function to compute the Region-wise total sales. Code the same functionality with groupby() function. Iterate over the groups and print the rows in each group.

```
for reg in df['Region'].unique():
    print(reg,df[df['Region']==reg]['Sales'].sum())

reg_sales = df.groupby('Region')[['Sales']].sum()
print(reg_sales)

reg_sales = df.groupby('Region')
for reg,data in reg_sales:
      print(reg)
      print(data)
```

b) Explain the syntax and semantics of Pandas pivot_table() function.
The Stock price DataFrame comprises of day-wise (for a month) indexed rows of StockCode, Open, Close, Min, Max and Volume columns. Write code to create Pivot tables of a) Day-wise Close values for each Stock b)Stock-wise values of total volmes of the month c) Stock-wise mean values of Open, Close, Min and Max.

```
stocks.pivot(index='symbol', columns='date', values='close')
stocks.pivot_table(index='symbol', values='volume')
stocks.pivot_table(index='symbol',          values=['open','close','min','max'],
aggfunc=np.mean)
```

Unit -IV

8. a) Create data for monthly sales of an organization for an year. Write Code to draw a waterfall chart with title, labels, formatting and colours.

```
Sales=pd.DataFrame({'price':[63.8,55.7,32.5,18.4,29.4,40.3,43.2,44.7,40.9,40.2,4
2.7,50]},index=['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov
','Dec'])
deltas=[Sales['price'][i] if i==0 else Sales['price'][i]-Sales['price'][i-1] for
i in range(len(Sales))]
Sales['delta']=deltas
import waterfall_chart
waterfall_chart.plot(Sales.index,Sales['delta'])
```

b) A Newly Under Graduate joined Students data consisting of Regd. No, Branch, 12th CGPA and I semester CGPA. Select a plot and code the snippet using Seaborn package to draw the plot for comparing 12th CGPA and the I Semester CGPA of each branch

```
import seaborn as sns
sns.lmplot(
    data=ugmarks, x="gpa_12", y="gpa_sem_1",
    hue="branch")
```

(OR)

9. a) Code snippets to draw Line Plot and Subplots using matplotlib with data selected from Iris dataset. Draw the resulting charts with brief descriptions.

a)

```
import matplotlib.pyplot as plt
```

```
iris=pd.read_csv('iris.csv')
plt.figure(figsize=(10,15))
x=range(0,len(iris.sepal_length))
plt.plot(x,iris.sepal_length)
plt.show()
```

b)

```
# Create a figure
fig = plt.figure(figsize=(10,15),linewidth=10,edgecolor='b')
# Add a subplot
ax1 = fig.add_subplot(121)
# Another equivalent but more general method
#ax = fig.add_subplot(1, 2, 1)
ax2 = fig.add_subplot(122)
ax1.plot(x,iris.sepal_length)
ax2.plot(x,iris.sepal_width)
```

b) State the packages available in python for visualizing the data. Explore any 3 of them in detail with a neat diagram for each.

a)Matplotlib
b)Seaborn
c)Plotly

Matplotlib:

Matplotlib is a comprehensive library for creating visualizations in Python. One can create publication quality plots, customize visual style and layout, draw subplots, add labels, titles, legends, export to many file formats, embed in JupyterLab and Graphical User Interfaces.

Seaborn:

Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets data scientists focus on what the different elements of plots mean, rather than on the details of how to draw them.

Plotly:

The plotly Python library is an interactive, open-source plotting library. It supports over 40 unique chart types covering a wide range of statistical, financial, geographic, scientific, and 3-dimensional use-cases.