Est. 1841 | YORK ST JOHN UNIVERSITY

MSc Computer Science

LDS7001M Statistical Programming

Week 2:

Big Data Development Stages for Insightful Analysis

Module Director: Dr. Rebecca Balasundaram

Lecturer : Dr. Suman Ghosh

Objectives

1. Efficient Data Collection & Management
2. Data Quality & Preprocessing
3. Scalable Data Processing & Analysis
4. Effective Data Visualization & Reporting
5. Robust Deployment & Monitoring

Stretch and challenge:

Real-Time Big Data Processing

Automated Machine Learning Pipelines

Scalable and Cost-Optimized Cloud Infrastructure

Challenges

Data Privacy and Security

Handling Large and Diverse Datasets

Computational and Storage Limitations

Link to prior learning

https://moodle.yorksj.ac.uk/course/view.php?id=37068&section=6

- What is Big Data?
- What are the types?
- 5 Vs of Big data?
- Big Data Can Influence Decision-Making for Business, Use cases,

Stages of Big Data Development for Insightful Analysis

1. Data Acquisition (Collection & Ingestion)	2. Data Preprocessing (Cleaning & Transformation)
Goal: Gather structured, semi-structured, and unstructured data from multiple sources. Sources: Sensors (IoT devices, wearables) Web scraping & APIs Databases & Data Warehouses Social media, logs, and transactional systems Technologies: Apache Kafka, Flume (Streaming) Hadoop HDFS, Amazon S3 (Storage) SQL/NoSQL Databases	Goal: Clean and prepare data for analysis by handling missing values, duplicates, and inconsistencies. Key Tasks: Handling missing data (mean imputation, interpolation) Removing duplicates & inconsistencies Converting data types (date formats, categorical encoding) Normalization & Scaling (for machine learning) Technologies: Python (Pandas, NumPy, Scikit-learn)

Stages of Big Data Development for Insightful Analysis

3. Data Storage & Management	4. Data Processing (Computation & Analytics)
Goal: Organize data efficiently for retrieval and processing. Storage Options: Data Lakes (HDFS, Amazon S3, Azure Data Lake) Data Warehouses (Google BigQuery, Snowflake, Redshift) Databases (SQL - PostgreSQL, MySQL; NoSQL - MongoDB, Cassandra) Best Practices: Schema design for structured data Partitioning for large datasets Indexing for fast querying	Goal: Process large-scale data using batch or real-time methods. Processing Approaches: Batch Processing (MapReduce, Spark) – For historical analysis Real-time Streaming (Apache Kafka, Flink, Spark Streaming) – For live insights Key Frameworks: Apache Spark, Apache Hadoop Google Cloud Dataflow, AWS Lambda

Stages of Big Data Development for Insightful Analysis

5. Data Analysis & Machine Learning	6. Data Visualization & Reporting
Goal: Extract insights through statistical analysis, machine learning (ML), and deep learning (DL). Techniques: Descriptive Analytics (Summarization, Visualization) Predictive Analytics (Regression, Classification) Prescriptive Analytics (Optimization, Decision Support) Deep Learning (Neural Networks for Image/Text Analysis) Tools: Python (Pandas, Scikit-learn, TensorFlow, PyTorch) R, MATLAB, SAS	Goal: Present insights through dashboards, reports, and visual analytics. Visualization Techniques: Charts & Graphs (Bar, Line, Pie, Scatter Plots) Geospatial Visualization (Heatmaps, Choropleths) Interactive Dashboards (Power BI, Tableau, Plotly) Tools: Power BI, Tableau, Google Data Studio Matplotlib, Seaborn, Plotly

Stages of Big Data Development for Insightful Analysis

5. Data Analysis & Machine Learning Goal: Extract insights through statistical analysis, machine learning (ML), and deep learning (DL). Techniques: Descriptive Analytics (Summarization, Visualization) Predictive Analytics (Regression, Classification) Prescriptive Analytics (Optimization, Decision Support) Deep Learning (Neural Networks for Image/Text Analysis) Tools: Python (Pandas, Scikit-learn, TensorFlow, PyTorch) R, MATLAB, SAS	6. Data Visualization & Reporting Goal: Present insights through dashboards, reports, and visual analytics. Visualization Techniques: Charts & Graphs (Bar, Line, Pie, Scatter Plots) Geospatial Visualization (Heatmaps, Choropleths) Interactive Dashboards (Power BI, Tableau, Plotly) Tools: Power BI, Tableau, Google Data Studio Matplotlib, Seaborn, Plotly

Stages of Big Data Development for Insightful Analysis

7.Decision Making & Business Strategy
Goal: Use insights for decision-making, forecasting, and strategic planning.

Applications:
Optimizing marketing campaigns
Fraud detection & anomaly detection
Demand forecasting & trend analysis
Customer segmentation & personalization

Implementation:
AI-powered automation (Chatbots, Recommendation Systems)
Business Intelligence (BI) for executive decision-making

Big Data Development Stages

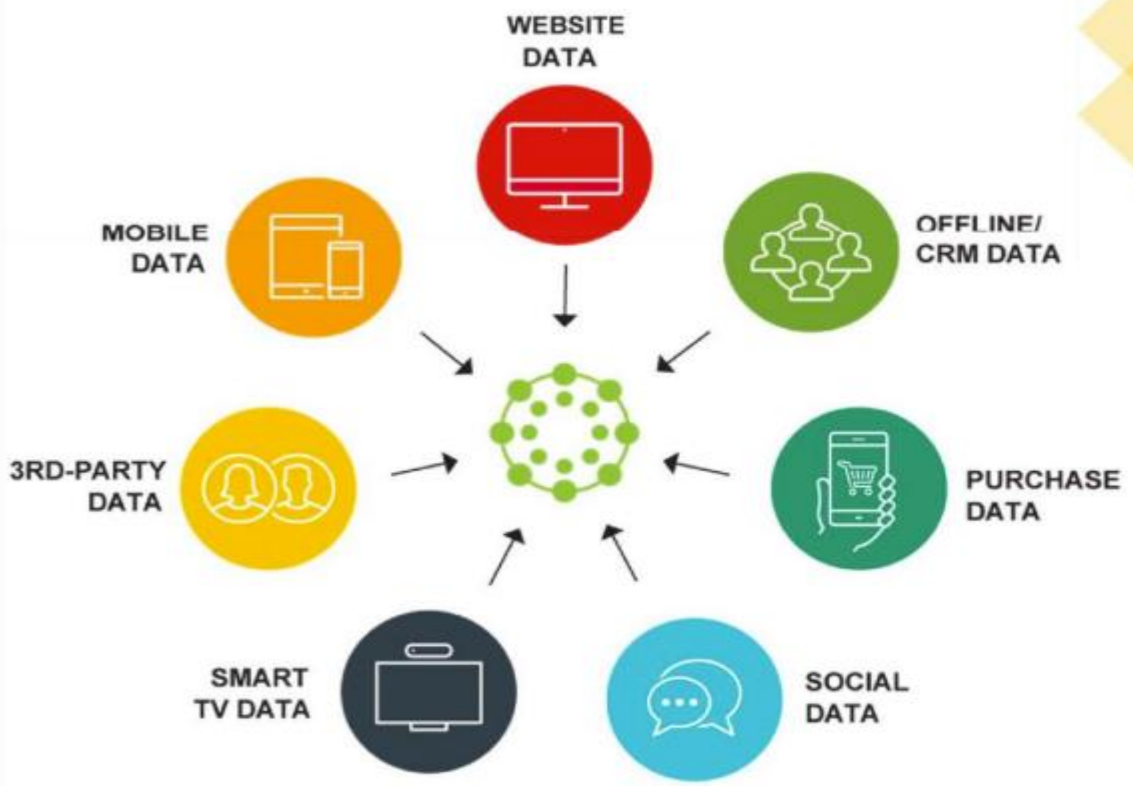
Stage	Key Focus	Tools & Technologies
1. Data Acquisition	Collecting raw data	Kafka, APIs, IoT, Web Scraping
2. Data Preprocessing	Cleaning & Transforming data	Pandas, NumPy, PySpark
3. Data Storage	Storing structured & unstructured data	Hadoop, S3, PostgreSQL
4. Data Processing	Processing data (Batch & Real-time)	Spark, Hadoop, Flink
5. Data Analysis	Statistical & Machine Learning models	Scikit-learn, TensorFlow, R
6. Data Visualization	Presenting insights	Tableau, Power BI, Matplotlib
7. Decision Making	Business applications & automation	AI, BI tools, Forecasting

Topics

- Types of Raw/Dirty Data
- Problems associated to raw data
- Diagnosing data problems
- Data Wrangling Goals
- Data Wrangling Steps
- Data Wrangling in Python
- Data Sampling: Strategies for Sampling
- Missing Data Handling

Types of Raw /Dirty Data

- Data comes in all shapes and sizes
- CSV files, PDFs, texts, .jpg…
- Different files have different formatting
- Spaces instead of NULLs, extra rows
- “Dirty” data
- Unwanted anomalies
- Duplicates



Problems Associated with Raw Data

- Missing data
- Incorrect data
- Inconsistent representations of the same data
- About 75% of data problems require human intervention
- Cleaning data vs overly--sanitizing data

Diagnosing Data Problems

- Visualizations can convey "raw" data
- Different visual representations/querying techniques highlight different types of data issues
  – Outliers often stand out in a plot
  – Missing data will cause gap or zero value
- Becomes increasingly difficult as data gets larger
  – Visual design coupled with interaction
  – Sampling

Data Wrangling: Formal Definition

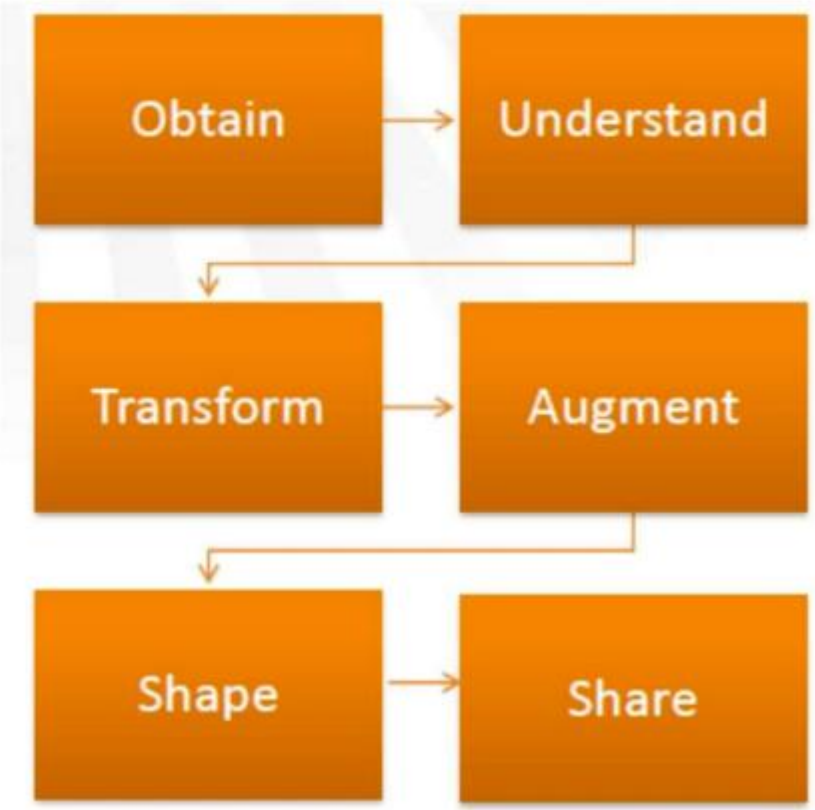
- The process of transforming “raw” data into data that can be analyzed to generate valid actionable insights
- Data Wrangling :
- Data preprocessing
- Data preparation
- Data Cleansing
- Data Scrubbing
- Data Munging
- Data Transformation

Data Wrangling Goals

- Goal: extract and standardize the raw data
  – Combine multiple data sources
  – Clean data anomalies
  – Avoid poor outcomes because of bad data
- Combine automation with interactive visualizations to aid in cleaning
- Improve efficiency and scale of data importing

Data Wrangling Steps

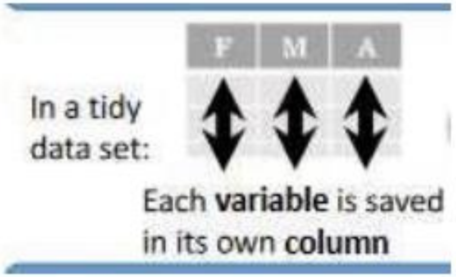
- Iterative process of
  - Obtain
  - Understand
  - Explore
  - Transform
  - Augment/enrich
  - Validate/shape
  - Visualize



Est. 1841 | YORK ST JOHN UNIVERSITY
London Campus

Data Wrangling Steps

- Data Import/Ingestion: CSV, Pdf, API/JSON/HTML Web Scraping
- Data Exploration: Visual inspection and Graphing
- Data Cleansing: -Missing value handling, formatting, outlier removal, Correction data errors per domain
- Data Augmenting: Aggregate Data Sources : merge, concat, Fuzzy/exact match
- Data Shaping: Tidying the data



Types of data

There are two basic types of data: numerical and categorical data.

Numerical data: data to which a number is assigned as a quantitative value.

Categorical data: data defined by the classes or categories into which an individual member falls.

Structured Vs Unstructured Data

"Looks like my V8 Chevy is running low on fuel. Didn't I fill up just the day before?"

UNSTRUCTURED

STRUCTURED

Owner	Vehicle	Type	Fuel Level	Engine	Last Fill
AK	Chevy	Gas	5%	V8	05/04/16

Continuous or Non-continuous data

A continuous variable is one in which it can theoretically assume any value between the lowest and highest point on the scale on which it is being measured

- (e.g. weight, speed, price, time, height)

Non-continuous variables, also known as discrete variables, that can only take on a finite number of values

- Discrete data can be numeric -- like numbers of apples -- but it can also be categorical -- like red or blue, or male or female, or good or bad.

Qualitative vs. Quantitative Data

- A qualitative data is one in which the “true” or naturally occurring levels or categories taken by that variable are not described as numbers but rather by verbal groupings
- Open ended answers

- Quantitative data on the other hand are those in which the natural levels take on certain quantities (e.g. price, travel time)
- That is, quantitative variables are measurable in some numerical unit (e.g. pesos, minutes, inches, etc.)
- Likert scales, semantic scales, yes/no, check box

Est. 1841 YORK ST JOHN UNIVERSITY London Campus

Data Wrangling in Python

- Numpy (aka Numerical Python): It's the most basic python package for data science. One can perform operations on n-arrays and matrices in Python using Numpy. It provides vectorization of mathematical operations on the NumPy array type, which helps improve performance and accordingly speeds up the execution of the python code.
- Pandas: It makes data analysis operations faster and easier. Useful for data structures with labeled axes. Some data alignment prevents common errors that can be extracted from misaligned data during data scraping.
- Matplotlib: It's the most common python visualization module. One can create line graphs, pie charts, histograms, and other professional-grade figures.

- Plotly: for interactive, publication-quality graphs. Great for creating line plots, scatter plots, area charts, bar charts, error bars, box plots, histograms, heatmaps, subplots, multiple-axis, polar graphs, and bubble charts.

Est. 1841 YORK ST JOHN UNIVERSITY London Campus

Exploring Your Data

- The simplest case is when you have a structured data set, which is just a collection of numbers. For example,
- daily average number of minutes each user spends on your site,
- the number of times each of a collection of data science tutorial videos was watched,
- the number of pages of each of the data science books in your data science library.
- An obvious first step is to compute a few summary statistics.
- You'd like to know how many data points you have, the smallest, the largest, the mean, and the standard deviation.

CSV Data Import/Ingestion

In [3]:
```
df_ffire = pd.read_csv('./dataset/module3/brazilianfire.csv')
order_col = ['Date Reported', 'Year', 'Month', 'State', 'Number of Fires']
df_ffire['Number of Fires'] = df_ffire['Number of Fires'].astype(int)
df_ffire = df_ffire[order_col]
df_ffire.head(10)
```

Out[3]:

	Date Reported	Year	Month	State	Number of Fires
0	1/01/1998	1998	January	Acre	0
1	1/01/1999	1999	January	Acre	0
2	1/01/2000	2000	January	Acre	0
3	1/01/2001	2001	January	Acre	0
4	1/01/2002	2002	January	Acre	0
5	1/01/2003	2003	January	Acre	10
6	1/01/2004	2004	January	Acre	0
7	1/01/2005	2005	January	Acre	12
8	1/01/2006	2006	January	Acre	4
9	1/01/2007	2007	January	Acre	0

- Lets take an example dataset
- Load the Brazilian Fire Dataset

https://www.kaggle.com/gustavomodelli/forest-fires-in-brazil/version/1

Python options for Summarizing Data

Summarize Data

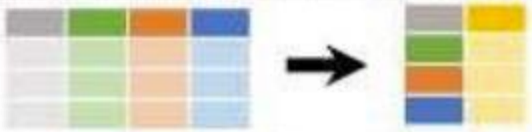
`df['w'].value_counts()`
Count number of rows with each unique value of variable

`len(df)`
# of rows in DataFrame.

`df.shape`
Tuple of # of rows, # of columns in DataFrame.

`df['w'].nunique()`
# of distinct values in a column.

`df.describe()`
Basic descriptive and statistics for each column (or GroupBy).



pandas provides a large set of summary functions that operate on different kinds of pandas objects (DataFrame columns, Series, GroupBy, Expanding and Rolling (see below)) and produce single values for each of the groups. When applied to a DataFrame, the result is returned as a pandas Series for each column. Examples:

`sum()`
Sum values of each object.

`count()`
Count non-NA/null values of each object.

`median()`
Median value of each object.

`quantile([0.25,0.75])`
Quantiles of each object.

`apply(function)`
Apply function to each object.

`min()`
Minimum value in each object.

`max()`
Maximum value in each object.

`mean()`
Mean value of each object.

`var()`
Variance of each object.

`std()`
Standard deviation of each object.

Describe method in Pandas for Summary

YORK ST JOHN UNIVERSITY
London Campus

```
df_ffire.describe(include='all')
```

	Date Reported	Year	Month	State	Number of Fires
count	6454	6454.000000	6454	6454	6454.000000
unique	20	NaN	12	27	NaN
top	1/01/2007	NaN	January	Alagoas	NaN
freq	324	NaN	541	240	NaN
mean	NaN	2007.461729	NaN	NaN	108.235358
std	NaN	5.746654	NaN	NaN	190.843947
min	NaN	1998.000000	NaN	NaN	0.000000
25%	NaN	2002.000000	NaN	NaN	3.000000
50%	NaN	2007.000000	NaN	NaN	24.000000
75%	NaN	2012.000000	NaN	NaN	113.000000
max	NaN	2017.000000	NaN	NaN	998.000000

Summary Statistics: Mean/average vs median vs mode

•(Arithmetic) Mean: the "average" value of the data

$$\mu = \frac{1}{n} \sum_i x_i$$

```
def mean(a): return sum(a) / float(len(a))
```

```
def mean(a): return reduce(lambda x, y: x+y, a) / float(len(a))
```

- Average: can be ambiguous
- The average household income in this community is $60,000
- The average (mean) income for households in this community is $60,000
- The income for an average household in this community is $60,000
- What if most households are earning below $30,000 but one household is earning $1M
- Median: the "middlest" value, or mean of the two middle values
- Can be obtained by sorting the data first • Does not depend on all values in the data.
- More robust to outliers

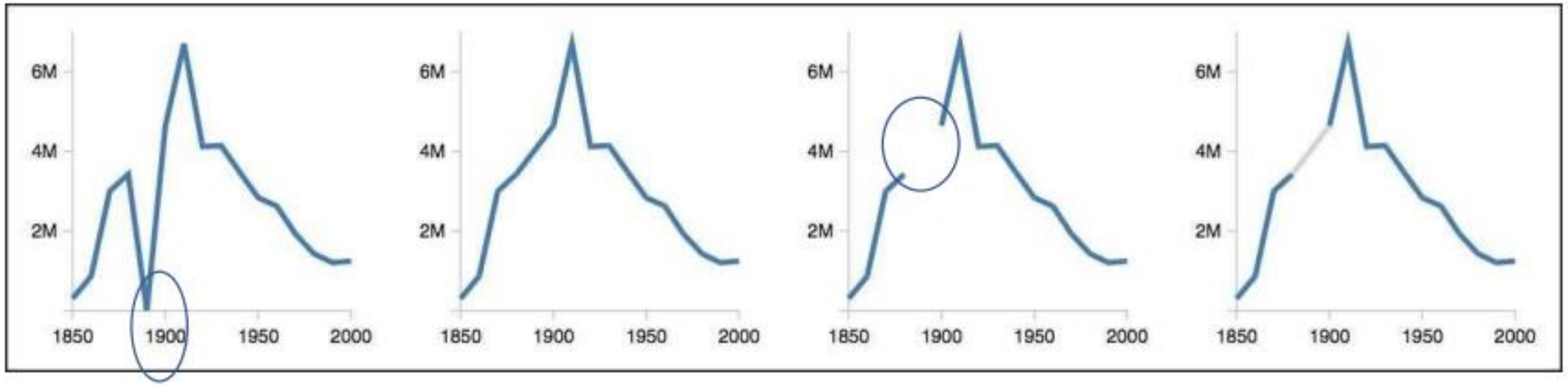
Handling the Missing Data

Reasons for Having Missing data

- Data can be missing from databases or be unreliable for a number of reasons:
- Human error when entering the data
- Inaccuracies or errors of instruments recording values
- Changes in procedures, or in requirements
- Difficulties with the integration of data from different sources.
- Rigidity of the structure of database systems, which may require fields to exist even when they make no sense for a particular record.

Handling the Missing Data

- Set values to zero?
- Interpolate based on existing data?
- Omit missing data?



Cleansing: removal of records

- The extent of the problem of missing values, as well as the mechanism for missing, may be important in determining which strategy to adopt for dealing with them.

- Some databases may contain a few records for which some or many of the attribute values are unknown.
- In such case, an approach for dealing with missing value could be to discard those records.

Cleansing: imputation

- The next alternative to discarding missing values is to try to produce "guesses" for the missing data. This is called data completion or data imputation.
- Data imputation can be achieved by various methods including:
- Complete with default values which are embedded as domain knowledge
- Complete with values calculated as the most common value for an attribute (e.g. an arithmetic mean or mode).
- Complete using k-Nearest Neighbour techniques.
- Create a model to predict the missing data.

YORK ST JOHN UNIVERSITY Est 1841 London Campus

Missing Data Handling in python

- Total number of missing values per column:

```
print(df.isnull().sum())
```

	First Name	Gender	Salary	Bonus %	Senior Management	Team
0	Douglas	Male	97308	6.945	TRUE	Marketing
1	Thomas	Male	61933	NaN	TRUE	NaN
2	Maria	Female	130590	11.858	FALSE	Finance
3	Jerry	Male	NaN	9.34	TRUE	Finance
4	Larry	Male	101004	1.389	TRUE	Client Services

Missing Data Handling in python

```
# drop all rows with NaN values
df.dropna(axis=0,inplace=True)
```

Replacing NaNs with a single constant value:

```
df['Salary'].fillna(0, inplace=True)
```

Replacing NaNs using Median/Mean of the column

```
# using median
df['Salary'].fillna(df['Salary'].median(), inplace=True)


#using mean
df['Salary'].fillna(int(df['Salary'].mean()), inplace=True)
```

YORK ST JOHN UNIVERSITY London Campus Est. 1841

Replace and interpolate

```
# will replace NaN value in Salary with value 0
df['Salary'].replace(to_replace = np.nan, value = 0,inplace=True)
```

```
df['Salary'].interpolate(method='linear', direction = 'forward',
inplace=True)
print(df['Salary'].head(10))
```

Learning Outcomes

- We have learnt about:
- Problems with raw data
- Produce summary statistics of a dataset
- Dataset sampling and balancing
- Missing data handling

Learning check

Which of the following is NOT typically a step in the data wrangling process?

a) Data collection

b) Data cleaning

c) Data mining

d) Data transformation

True/False:

It is generally better to remove missing values entirely rather than trying to impute them. (True/False)

Activity

What are the key challenges and opportunities in Big Data analytics? List as many relevant keywords, concepts, and technologies as possible.

Why the functions???
df.fillna()
drop_duplicates()
to_datetime()

Conclusion/feedforward activity(s)

Conclusion: The development of Big Data technologies has progressed through various stages, from data collection and storage to processing, analysis, and visualization. Each stage plays a crucial role in transforming raw data into actionable insights, which drive informed decision-making and strategic planning. By integrating advanced algorithms and tools, organizations can unlock the full potential of Big Data, enabling a deeper understanding of patterns and trends that were previously inaccessible.

Feedforward: Moving forward, it is essential to continue refining each stage of Big Data development by embracing emerging technologies such as AI, machine learning, and real-time analytics. Organizations should also focus on improving data governance and security to ensure compliance and protect sensitive information. Additionally, further investment in skill development for data professionals will help bridge the gap between technology and practical application, enhancing the effectiveness of Big Data initiatives.

Conclusion of objectives

In conclusion, achieving efficient data collection and management is foundational for ensuring high-quality, reliable datasets.

By focusing on effective pre-processing techniques, data quality is maintained, enhancing the accuracy and consistency of subsequent analyses.

Scalable data processing and analysis enable the handling of large datasets, ensuring that insights can be generated swiftly and efficiently, even as data volumes grow.

Effective data visualization and reporting are essential for communicating complex findings in an accessible manner, facilitating decision-making.

Support available

- Academic Quality to check in with Iain Pullar following recruitment summer 2023

References

1. Bi-Kring. (n.d.). The 4 Steps of the Big Data Life Cycle. Retrieved from https://bi-kring.nl/29-business-intelligence/1353-the-4-steps-of-the-big-data-life-cycle?utm_source=chatgpt.com

2. Harvard Business School Online. (n.d.). 8 Steps in the Data Life Cycle. Retrieved from https://online.hbs.edu/blog/post/data-life-cycle?utm_source=chatgpt.com

3. Towards Data Science. (2021, April 29). The Five Stages of Big Data. Retrieved from https://towardsdatascience.com/the-five-stages-for-big-data-b89ad1e8e156/?utm_source=chatgpt.com

4. GeeksforGeeks. (2021, February 9). Big Data Analytics Life Cycle. Retrieved from https://www.geeksforgeeks.org/big-data-analytics-life-cycle/?utm_source=chatgpt.com

5. DASCA. (n.d.). Big Data Processing: Transforming Data into Actionable Insights. Retrieved from https://www.dasca.org/world-of-data-science/article/big-data-processing-transforming-data-into-actionable-insights?utm_source=chatgpt.com

Note to educator: please adhere to [YSJU Harvard Referencing]