U1: Introduction to Data Science and Big Data

Introduction to Data Science and Big Data

Basics and Need of Data Science and Big Data

- Data Science: A field that combines statistics, programming, and domain expertise to extract meaningful insights from data.
- Big Data: Refers to datasets that are too large or complex for traditional data processing techniques.

Applications of Data Science

- Examples include healthcare (predictive analytics), finance (fraud detection), marketing (customer segmentation), and social media (trend analysis).

Data Explosion

- The rapid increase in the amount of data generated, requiring advanced tools to manage and analyze it.

The 5 V's of Big Data

- Volume: Large amounts of data.
- Velocity: Speed at which data is generated and processed.
- Variety: Different types of data (structured, unstructured, semi-structured).
- Veracity: The quality and trustworthiness of the data.
- Value: The usefulness of the data for decision-making.

Relationship between Data Science and Information Science

- Data Science: Focuses on the extraction of insights and knowledge from data.
- Information Science: Focuses on the storage, retrieval, and management of information.

Business Intelligence vs. Data Science

- Business Intelligence: Focuses on querying and reporting historical data.
- Data Science: Focuses on predictive and prescriptive analytics using both historical and real-time data.

Data Science Life Cycle

- Stages: Problem identification, data collection, data cleaning, exploratory data analysis, modeling, evaluation, and deployment.

Types of Data

- Structured Data: Organized data, typically in rows and columns (e.g., databases).
- Unstructured Data: Data without a predefined structure (e.g., text, images).
- Semi-Structured Data: Data that doesn't have a fixed schema but contains some organizational structure (e.g., XML, JSON).

Data Collection

- Gathering data from various sources like databases, sensors, web scraping, APIs, and surveys.

Need for Data Wrangling

- Data wrangling is necessary to clean and prepare raw data for analysis, ensuring it's accurate, consistent, and usable.

Methods of Data Wrangling

- Data Cleaning: Removing errors or inconsistencies in the data.
- Data Integration: Combining data from multiple sources into a unified dataset.
- Data Reduction: Reducing the volume of data while retaining essential information (e.g., dimensionality reduction).
- Data Transformation: Converting data into a format suitable for analysis (e.g., normalization, scaling).
- Data Discretization: Converting continuous data into discrete categories or intervals.

U2:Statistical Inference Statistical Inference

✅ 1. Need of Statistics in Data Science and Big Data Analytics

Statistics helps in collecting, analyzing, interpreting, and presenting data to make informed decisions. It supports model building, pattern detection, hypothesis testing, and data visualization in both Data Science and Big Data.

📊 2. Measures of Central Tendency

- **Mean:** Average of values.
  $$\text{Mean} = \frac{\text{Sum of all values}}{\text{Number of values}}$$
- **Median:** Middle value in sorted data.
- **Mode:** Most frequent value in the dataset.
- **Mid-range:**
  $$\text{Mid-range} = \frac{\text{Max + Min}}{2}$$

🎯 3. Measures of Dispersion

- **Range:**
  $$\text{Range} = \text{Max} - \text{Min}$$
- **Variance:** Average of squared deviations from the mean.
- **Mean Deviation:** Average of absolute deviations from the mean.
- **Standard Deviation (SD):** Square root of variance; shows data spread.

📘 4. Bayes Theorem

Used to find the probability of an event given prior knowledge.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

It helps in spam filtering, medical diagnosis, etc.

❓ 5. Basics and Need of Hypothesis and Hypothesis Testing

- **Hypothesis:** Assumption about a population parameter.
- **Hypothesis Testing:** Validates assumptions using sample data to decide whether to accept or reject a hypothesis.
- Needed for decision-making using sample data (e.g., does a new drug work?).

📈 6. Pearson Correlation

Measures linear relationship between two variables.

r=Cov(X,Y)σXσYr = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y}

- **r = 1:** Perfect positive
- **r = -1:** Perfect negative
- **r = 0:** No correlation

🧪 7. Sample Hypothesis Testing

Tests a hypothesis using a sample instead of the full population to make conclusions.
Involves steps:

1. State null and alternate hypotheses.
2. Choose significance level (α).
3. Calculate test statistic.
4. Compare with critical value or p-value.

🔢 8. Chi-Square Test

Tests independence between categorical variables or goodness of fit.

χ2=∑(O−E)2E\chi^2 = \sum \frac{(O - E)^2}{E}

- O = Observed, E = Expected
- Used in surveys, classification evaluation, etc.

📊 9. t-test

Used to compare means when sample size is small.

- **One-sample t-test:** Compares sample mean to known value.
- **Two-sample t-test:** Compares means of two samples.
- Assumes normally distributed data and unknown population SD.

Let me know if you want **formula sheet**, **example problems**, or **Python code** for any topic!

U3: Big Data Analytics Life Cycle

📌 Introduction to Big Data

Big Data refers to extremely large datasets that are complex and grow rapidly. It cannot be handled using traditional data processing tools.

Key Characteristics (5 Vs):

- **Volume** – Large amounts of data
- **Velocity** – Speed at which data is generated
- **Variety** – Different types (text, image, video, etc.)
- **Veracity** – Data accuracy and trustworthiness
- **Value** – Usefulness of data

🔍 Sources of Big Data

1. **Social Media** – Twitter, Facebook, Instagram
2. **Sensors/IoT Devices** – Smart devices, wearables
3. **Web Logs** – Browsing data, clickstream
4. **Transactional Data** – Online purchases, banking
5. **Mobile Devices** – Call logs, app usage
6. **Machine Data** – Server logs, telemetry
7. **Healthcare Systems** – EHRs, lab records

📊 Data Analytic Lifecycle (6 Phases)

Phase 1: Discovery

- Understand business problem and objectives.
- Identify required data and resources.

- Initial hypothesis setup.

Phase 2: Data Preparation

- Collect, clean, and format data.
- Handle missing values, outliers, duplicates.

Phase 3: Model Planning

- Choose statistical or ML methods.
- Select tools like R, Python, SQL for analysis.

Phase 4: Model Building

- Build models using selected techniques.
- Train and validate models using prepared data.

Phase 5: Communicate Results

- Interpret results in business terms.
- Use visualizations and summary reports.

Phase 6: Operationalize

- Deploy model in real-world environment.
- Set up performance tracking and feedback loop.

Let me know if you'd like **diagrams**, **real-world examples**, or **MCQs** for these topics!

U4: Predictive Big Data Analytics with Python

Introduction

- Focus on how Python is used in big data analytics, particularly for predictive modeling and machine learning.

Essential Python Libraries

- **NumPy**: For numerical computing and handling arrays.
- **Pandas**: For data manipulation and analysis (especially working with DataFrames).
- **Matplotlib**: For data visualization (creating plots and charts).
- **Seaborn**: For statistical data visualization.
- **SciPy**: For scientific and technical computing.
- **Scikit-learn**: For machine learning and predictive analytics.

Basic Examples

- Basic usage of libraries like creating arrays with NumPy, reading datasets with Pandas, and plotting simple graphs with Matplotlib.

Data Preprocessing

- **Removing Duplicates**: Identifying and removing duplicate rows in a dataset to avoid bias in analysis.
- **Transformation of Data using Function or Mapping**: Applying functions or mapping techniques to transform data (e.g., log transformation, applying custom functions to columns).
- **Replacing Values**: Replacing specific values in the dataset, such as replacing NaN values with a default or mean value.
- **Handling Missing Data**: Techniques like removing rows with missing data, imputing missing values (mean, median, or mode), or using algorithms that handle missing data internally.

Analytics Types

- **Predictive Analytics**: Analyzing historical data to make predictions about future outcomes (e.g., regression, classification).
- **Descriptive Analytics**: Analyzing past data to describe what has happened (e.g., summarizing trends).
- **Prescriptive Analytics**: Suggesting possible outcomes and actions to take based on predictive analysis.

Association Rules

- **Apriori Algorithm**: A classic algorithm used for finding frequent itemsets in datasets and deriving association rules from these itemsets.
- **FP-growth Algorithm**: A more efficient algorithm than Apriori for mining frequent itemsets by using a tree structure.

Regression

- **Linear Regression**: A method to model the relationship between a dependent variable and one or more independent variables by fitting a linear equation.
- **Logistic Regression**: Used for binary classification tasks, predicting the probability of one class versus another.

Classification

- **Naïve Bayes**: A classification technique based on Bayes' Theorem, assuming independence between features.
- **Decision Trees**: A tree-like structure where each node represents a decision based on an attribute, and branches represent the outcome of that decision.

Introduction to Scikit-learn

- **Installation**: Steps to install Scikit-learn and other necessary libraries.

- **Dataset**: Overview of datasets available within Scikit-learn (e.g., Iris, Digits).
- **Matplotlib**: Integrating matplotlib with Scikit-learn to visualize model results.

Filling Missing Values

- Using **Pandas** or Scikit-learn's **SimpleImputer** to fill in missing values with the mean, median, or mode.

Regression and Classification using Scikit-learn

- **Regression**: Applying linear regression or logistic regression using Scikit-learn's `LinearRegression` or `LogisticRegression` classes.
- **Classification**: Using classifiers like **Naïve Bayes** (`GaussianNB`), **Decision Trees** (`DecisionTreeClassifier`), and others for predicting class labels.

U5: Big Data Analytics and Model Evaluation

Clustering Algorithms

- **K-Means Clustering**:
    - An unsupervised learning algorithm used to partition data into K clusters based on feature similarity.
    - Objective: Minimize the within-cluster variance by iterating through the following steps:
        1. Assign points to the nearest centroid.
        2. Update centroids based on the mean of points in each cluster.
        3. Repeat the steps until convergence.
- **Hierarchical Clustering**:
    - Builds a tree of clusters (dendrogram) that shows the hierarchy of data points.
    - Agglomerative (bottom-up) and Divisive (top-down) are the two main approaches.
    - No need to predefine the number of clusters, unlike K-Means.
- **Time-Series Analysis**:
    - Analyzing time-ordered data to extract meaningful statistics, trends, and patterns.
    - Methods like moving averages, ARIMA, and exponential smoothing are commonly used.

2. Introduction to Text Analysis

- **Text Preprocessing**:
    - Steps include removing stopwords, stemming, lemmatization, and lowercasing.
    - Helps clean and normalize text data before analysis.
- **Bag of Words (BoW)**:

    - A representation of text where each word is treated as a feature, ignoring grammar and word order.
    - Creates a sparse matrix where each row represents a document and each column a word from the entire vocabulary.
- **TF-IDF (Term Frequency - Inverse Document Frequency)**:
    - A statistical measure to evaluate the importance of a word in a document relative to a corpus.
    - TF = frequency of a term in a document.
    - IDF = logarithm of the inverse of the frequency of the term across documents.
    - Helps in highlighting the words that are more important to the document.
- **Topic Modeling**:
    - Techniques like Latent Dirichlet Allocation (LDA) used to discover hidden topics within a collection of documents.
    - Helps in summarizing and organizing large datasets of textual information.

3. Social Network Analysis (SNA)

- **Introduction to SNA**:
    - Examines relationships (edges) between entities (nodes) in a network.
    - Used in domains like social media, communication networks, and organizational studies.
    - Key metrics: centrality, clustering, shortest paths, and community detection.
    - Popular algorithms: PageRank, betweenness centrality, and k-core decomposition.

4. Business Analysis

- **Introduction to Business Analytics**:
    - Uses data-driven approaches to analyze and improve business performance.
    - Encompasses descriptive, predictive, and prescriptive analytics.

    - Techniques include market basket analysis, customer segmentation, and trend analysis.

5. Model Evaluation and Selection

- **Metrics for Evaluating Classifier Performance**:
    - **Accuracy**: Proportion of correctly classified instances.
    - **Precision**: Proportion of true positive predictions among all positive predictions.
    - **Recall (Sensitivity)**: Proportion of true positive predictions among all actual positives.
    - **F1-Score**: Harmonic mean of precision and recall.
    - **AUC-ROC**: Area Under the ROC Curve, which evaluates classifier performance across different thresholds.
- **Holdout Method and Random Subsampling**:
    - **Holdout Method**: Split the dataset into training and testing sets (e.g., 70%-30%).
    - **Random Subsampling**: Repeatedly split the data into different training and test sets to get a more robust evaluation.
- **Parameter Tuning and Optimization**:
    - Hyperparameters of a model can be optimized using techniques like grid search or random search.
    - **Grid Search**: Exhaustively searches through a predefined set of hyperparameters.
    - **Random Search**: Randomly selects hyperparameters within a defined range.
- **Result Interpretation**:
    - Evaluate the model's performance using metrics, and adjust based on the business or research objectives.
    - Consider overfitting or underfitting during interpretation.

6. Practical Applications

- **Clustering and Time-Series Analysis Using Scikit-Learn**:
    - Use `KMeans`, `AgglomerativeClustering`, and `DBSCAN` for clustering tasks.
    - Time-series analysis tools like ARIMA, Prophet, and seasonal decomposition are also available in Python libraries.
- **Evaluation Using `sklearn.metrics`**:
    - Includes various functions for calculating classification metrics, such as `accuracy_score`, `confusion_matrix`, `roc_auc_score`, etc.
- **Confusion Matrix**:
    - A table to evaluate the performance of a classification model. It shows true positives, true negatives, false positives, and false negatives.
- **AUC-ROC Curves**:
    - A graphical representation of a classifier's ability to distinguish between classes. The area under the curve (AUC) gives an overall performance measure.
- **Elbow Plot**:
    - A method to determine the optimal number of clusters in K-Means clustering.
    - It plots the sum of squared distances from each point to its assigned cluster center (inertia) for different values of K. The "elbow" point indicates the best K.

U6:Data Visualization and Hadoop

Here's a **short and clear summary** of your topics for quick revision:

📌 Introduction to Data Visualization

Data visualization is the graphical representation of data to help people understand trends, patterns, and insights easily using charts, graphs, maps, etc.

⚠️ Challenges to Big Data Visualization

- **Large Volume** of data → hard to process in real-time.
- **Variety** → structured & unstructured formats.
- **Scalability** → performance issues.
- **Interactivity** → maintaining fast, responsive visuals.
- **Integration** → combining data from different sources.

📊 Types of Data Visualization

- **Univariate**: Single variable (e.g., histogram)
- **Bivariate**: Two variables (e.g., scatter plot)
- **Multivariate**: More than two (e.g., heatmaps)
- **Time-Series**: Data over time (e.g., line chart)
- **Geospatial**: Maps and locations

🛠️ Data Visualization Techniques

- Aggregation (e.g., mean, count)
- Filtering & zooming

- Clustering
- Heatmaps
- Dashboards

🌐 Visualizing Big Data

- Use tools that can process large-scale data (like Tableau with Hadoop)
- Real-time visualizations with streaming data
- Interactive dashboards

🧰 Tools Used in Data Visualization

- **Tableau**
- **Power BI**
- **Python (Matplotlib, Seaborn, Plotly)**
- **R**
- **D3.js**
- **Excel (basic level)**

🐘 Hadoop Ecosystem

A framework to store and process big data across distributed systems.

- **HDFS**: Hadoop Distributed File System
- **YARN**: Resource manager
- **MapReduce**: Processing engine

- **Hive**: SQL-like queries on Hadoop
- **Pig**: Scripting for data flow
- **HBase**, **Sqoop**, **Oozie**, **Zookeeper** (other components)

⚙️ MapReduce

Programming model to process large data sets in parallel:

- **Map**: Breaks data into key-value pairs.
- **Reduce**: Aggregates intermediate data into final results.

🐷 Pig

- Scripting platform on Hadoop
- Uses **Pig Latin** language
- Good for data transformation and ETL

🐝 Hive

- Data warehouse tool on Hadoop
- Uses **HiveQL (SQL-like)** for querying large datasets

🔬 Analytical Techniques in Big Data Visualization

- Clustering
- Classification
- Regression

- Association rules
- Trend analysis

🐍 Data Visualization using Python

1. Line Plot

- Shows data trends over time
  `plt.plot(x, y)`

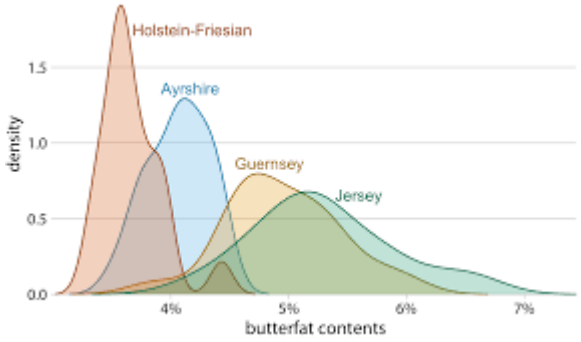
2. Scatter Plot

- Shows relationship between two variables
  `plt.scatter(x, y)`

3. Histogram

- Shows frequency distribution
  `plt.hist(data)`

4. Density Plot

- Smoothed version of histogram
- 

  `sns.kdeplot(data)`

5. Box Plot

- Shows distribution, outliers, and quartiles

```
sns.boxplot(data)
```