Data Science-Curriculum

1.Introduction to Data Science

1.1 Introduction

- Data science combines statistics, programming, and domain expertise to extract insights and knowledge from structured and unstructured data.
- Exploration involves identifying patterns in information
- Prediction involves information we know to make informed guesses about values we wish we knew
- Inference involves quantifying our degree of certainity

1.2 Data Science Pipeline

- Understand Problem Statement
- Data Acquistion
- Data Preprocessing
- Exploratory Analysis - Visualization
- Machine Learning & Deep Learning Modeling
- Final Insights

1.3 Data Science Career

- Data Science Engineer
- Data Analyst(Statistics, Visualisation)
- Machine Learning Engineer
- Deep Learning Engineer
- Data Engineer

2.Python

2.1 Introduction to Python

2.2 Python Functions, Packages, and Routines.

Functions

- Fuctions are blocks of reusable code that perform a specific task.They are defined using the def keyword, allow parameters, and can return results, making code more modular and organised.

Python Packages

- Packages are collections of modules that group related functions, classes, and routines together.

Routines

- Refers to a series of programmed instructions or functions that can be reused to perform common tasks. They help automate processes, improve efficiency, and minimise code duplication.

Data Science-Curriculum

2.3 Data Types, Operators, Variables

Data Types

- Python supports various data types, including integers (int), floating-point numbers (float), strings (str), and complex types like lists, tuples, dictionaries, and sets for managing diverse kinds of data.

Operators

- Python provides operators for performing operations on variables and values, including arithmetic (+, -, *, /), comparison (==, !=, <, >), logical (and, or, not), and assignment (=, +=, -=) operators.

Variables

- Variables are symbolic names assigned to values, acting as containers for storing data. They are dynamically typed in Python, meaning their type can change based on the assigned value.

2.4 Working with Data structure, Arrays, Vectors & Data Frames.

Data structures

- Data structures in Python (e.g., lists, tuples, dictionaries, and sets) are ways to store and organise data efficiently. They allow for easy access, modification, and management of data depending on the structure's properties.

Arrays

- Arrays (using libraries like numpy) and vectors are ordered collections of elements, typically of the same data type. Arrays support fast mathematical operations, while vectors are 1D arrays often used in linear algebra and machine learning.

Data Frame

- It is a two-dimensional, table-like data structure (from libraries like pandas) where data is stored in rows and columns. It's ideal for handling and manipulating structured data, similar to spreadsheets or SQL tables.

2.5 Syntax

- Rules and structure of code in programming.
- Defines correct keyword and symbol usage.
- Ensures code readability and functionality.
- Essential for error-free program execution.

2.6 Working with Numbers & Working with Strings

Working with Numbers

- Arithmetic operations like addition, subtraction, multiplication, and division.
- Handling numeric types like integers, floats, and complex numbers.

Working with Strings

- Manipulating text with functions like concatenation, slicing, and formatting.
- Supporting operations for string comparison, search, and transformation.

Data Science-Curriculum

2.6 Conditional Statements

- Allow decision-making in programming based on conditions.
- Include if, else if, and else clauses.
- Enable branching logic for different outcomes.
- Support complex conditions with logical operators.

2.7 For Loop & While Loop

For Loop

- Iterates over a sequence or range of values.
- Commonly used for executing code a specific number of times.

While Loop

- Repeats code while a condition remains true.
- Useful for indeterminate iterations until a condition changes.

2.8 Lists, Tuples, Sets

Lists

- Ordered, mutable collections that can hold mixed data types; defined with [ ]. Supports indexing, slicing, and dynamic modifications.

Tuples

- Ordered, immutable collections that can hold mixed data types; defined with ( ). Ideal for fixed data that should not be altered.

Sets

- Unordered, mutable collections with unique elements; defined with { }. Used for eliminating duplicates and efficient membership testing.

2.9 Dictionaries & Functions

Dictionaries

- Stores data in key-value pairs.
- Allows fast lookup, insertion, and deletion by keys.

Functions

- Encapsulate reusable blocks of code.
- Can accept parameters and return values.

Data Science-Curriculum

2.10 Pandas, NumPy, Matplotib packages

Pandas

- Powerful library for data manipulation and analysis, Pandas provides data structures like DataFrames, allowing for easy handling, cleaning, and transformation of structured data.

NumPy

- A fundamental package for numerical computations, NumPy offers support for multi-dimensional arrays and a wide range of mathematical functions for operations on arrays and matrices.

Matplotib

- A popular plotting library used for creating static, interactive, and animated visualisations in Python, Matplotlib allows users to generate a wide variety of charts, including line plots, histograms, and scatter plots.

2.11 Advance Data Processing with Numpy and Pandas

- **Vectorized Operations & Broadcasting (NumPy):** Use vectorized operations and broadcasting for high-performance calculations across entire arrays, eliminating slow loops.
- **Multi-dimensional Indexing (NumPy)**: Leverage advanced indexing techniques to manipulate specific array elements, slices, or even masked elements.
- **Data Manipulation with GroupBy (Pandas):** Apply the groupby method to split, apply functions, and combine data efficiently for aggregation, transformation, and filtering.
- **Merging & Joining (Pandas):** Combine datasets with merge, join, and concat for complex data alignment, such as relational data processing, without duplicating data.
- **Handling Missing Data:** Use methods like fillna, dropna, and interpolation to handle missing values seamlessly, ensuring data integrity.
- **Performance Optimization**: Apply apply, map, and vectorized functions over loops, and manage data types carefully to optimize memory usage and computation speed.

2.12 Advance Data Visualization with MatplotLib

- **Subplots and Grid Layouts:** Use subplot and gridspec to organize multiple plots in complex layouts, allowing for effective side-by-side comparisons.
- **Customizing Styles and Themes:** Customize plot aesthetics with plt.style and use themes (like seaborn or ggplot) for a polished look, or define custom color palettes and fonts for brand consistency.
- **Annotations and Text:** Add annotations, arrows, and text labels to highlight data points or trends, using annotate and text to make plots more informative.
- **Interactive Visualizations:** Incorporate mpl_toolkits for 3D plotting and widget modules for sliders and interactive elements, enhancing engagement and interactivity.
- **Advanced Color Mapping:** Utilize color maps (cmap) to represent additional data dimensions, allowing easy visual encoding of numerical ranges or categories.
- **Efficient Data Rendering:** Optimize plots with large datasets using agg, imshow, or downsampling techniques, and save figures in vector formats for high-quality exports.

Data Science-Curriculum

3.Maths required for Data Science

3.1 Linear Algebra
- Matrices and Vectors
- Matrix Multplication
- Eigenvalues and Eigenvectors

3.2 Calculus
- Integrals
- Derivatives
- Gradient Descent

3.3 Introduction to Statistics
- What is Statistics
- Why do we need Statistics
- Statistical Models

3.4 Descriptive statistics
- Measure of Central Tendency
- Mean
- Outliers
- Median
- Mode
- Bimodal

3.5 Measure of Spead
- Variance
- Standard Deviation

3.6 Probability
- What is Probability
- How to Calculate Probability with examples
- Union and Intersection

3.7 Conditional Probability
- What is Conditional Probability
- How to Calculate Conditional Probability with examples

3.8 Data Preprocessing
- Data Clearning
- Missing Values
- Centring and Scalaing
- Resolve Skewness
- Resolve Outliners
- Collinearity
- Sparse Variables
- Re-encoded Dummy Variables

Data Science-Curriculum

4.Introduction to Machine Learning

4.1 Introduction

4.2 Types of ML Algorithms

- Supervised Learning - Classification, Regression
- Unsupervised Learning - Clustering, Dimensionality Reduction
- Semi-Supervised Learning
- Reinforcement Learni ng
- Terminologies of Machine Learning - Model, Feature, Target(Label), Training, Prediction

4.3 Steps in Building ML Model

- Importing the Data
- Exploratory Data Analysis (EDA)
- Data Transformation
- Model Selection
- Training & Testing the Model
- Deplyoment of Model

4.4 Linear Regression

- **Model Overview:** Linear regression models the relationship between a dependent variable and one or more independent variables by fitting a linear equation to observed data.
- **Applications and Assumptions:** Commonly used for predicting continuous outcomes, it assumes a linear relationship, homoscedasticity, independence, and normally distributed residuals.
- **Types:** Simple linear regression uses one independent variable, while multiple linear regression extends this to multiple variables, offering more flexibility and predictive power.

4.5 Logistic Regression

- **Model Overview:** Logistic regression estimates the probability of a binary outcome by applying a logistic function, making it ideal for classification tasks.
- **Applications and Assumptions:** Used in fields like healthcare and finance for binary outcomes (e.g., risk assessment), it assumes no multicollinearity and a linear relationship between predictors and the log-odds.
- **Types:** It includes binary logistic regression (two outcomes), multinomial logistic regression (multiple categories), and ordinal logistic regression (ordered categories).

4.6 KNN K-Nearest Neighbours

- **Purpose:** KNN is a simple, instance-based algorithm that classifies data points based on the majority class of their nearest neighbors in feature space.
- **Applications and Limitations:** Commonly used in recommendation systems and image recognition, KNN can be computationally intensive with large datasets and sensitive to irrelevant features.

Data Science-Curriculum

4.7 Naive Bayes

- **Model Overview:** Naive Bayes is a probabilistic classifier that uses Bayes' Theorem, assuming that features are conditionally independent given the target variable.
- **Applications and Suitability:** Commonly used for text classification, spam detection, and sentiment analysis, it performs well on high-dimensional data but may struggle with correlated features.

4.8 Clustering

Methods

- Partitioning Clustering
- Density Based Clustering
- Distribution Model Based Clustering
- Hierarchical Clustering
- Fuzzy Clustering

Applications

- In Identification of Cancer Cells
- Search Engines
- Customer Segmentation
- Biology
- Land Use

4.9 K-Means Clustering

- What is K-Means Clustering
- How Does K-Means Cluseting Algorithm Work
- How to find optimal value for K in K-Means

4.10 Hierarchical Clustering

- Top Down Approach
- Bottom Up Approach
- How to linkage affect the Dendrogram
- Single Linkage
- Complete Linkage
- Average Linkage

4.11 Dimensionality Reduction

- Advantages & Disadvantages
- Feature Selection
- Feature Extraction
- Common Techniques of Dimensionality Reduction

4.12 Principal Component Analysis

- Terminologies of PCA Algorithm
- Steps for PCA Algorithm

Data Science-Curriculum

4.13 Linear Discriminant Analysis

- What is LDA
- How does LDA work
- How to prepare data from LDA

4.14 Semi-Supervised Learning

- **Model Overview:** Semi-supervised learning combines a small amount of labeled data with a large amount of unlabeled data to improve learning accuracy without extensive labeling costs.
- **Working of the Model**
- **Assumptions:** Continuity Assumptions, Cluster Assumptions, Manifold Assumptions
- **Applications:** Speech Analysis, Web Content Classification, Protien Sequence Classification, Text Document Classifier

4.14 Reinforcement Learning

- **Model Overview:** Reinforcement learning enables an agent to learn optimal actions through trial and error by receiving rewards or penalties from its environment.
- Terminologies of Reinforcement Learning
- Key Features of Reinforcement Learning
- Approaches to implement Reinforcement Learning
- **Types of Reinforcement Learning:** Positive Reinforcement and Negative Reinforcement
- **Algorithms:** Q-Learning, State Action Reward State Action(SARSA), Deep Q Neural Networks(DQN)
- **Applications:** Robotics, Chemistry, Manufacturing, Finance, E-Sports

Data Science-Curriculum

5.Deep Learning and Data Mining

5.1 Introduction to Deep Learning

5.2 Architecture and Application

- **Architecture** - Deep Learning Network, Deep Belief Network
- **Types** - FFNN,CNN,Restricted Boltzmann Machine, Autoencoders

5.3 Deep Learning Algorithms

- Convolutional Neural Networks
- Long Short Term Memory Networks
- Recurrent Neural Networks
- Generative Adversarial Networks
- Radial Basis Function Networks

5.4 Introduction Tensforflow & Keras

5.5 Data Mining

- **Overview:**Data mining is the process of discovering patterns, correlations, and useful information from large datasets using various techniques from statistics, machine learning, and database systems.
- **Techniques:** Common methods include clustering, classification, regression, association rule mining, and anomaly detection, each serving different analytical purposes.
- **Applications:** Widely applied in fields such as marketing for customer segmentation, finance for fraud detection, healthcare for patient outcome analysis, and social media for sentiment analysis.

5.6 Data Pre-processing in Data Mining

- Data Cleaning
- Data Transformation
- Data Reduction

5.7 Natural Language Processing

- **Overview:** NLP is a field of artificial intelligence that focuses on the interaction between computers and human language, enabling machines to understand, interpret, and generate natural language text.
- **Techniques:** Key techniques include tokenization, part-of-speech tagging, named entity recognition, sentiment analysis, and machine translation, allowing for diverse applications.
- **Applications:** NLP is used in chatbots, virtual assistants, text summarization, language translation, and sentiment analysis in social media, enhancing user experiences and automating tasks.

5.8 Data Pre-processing in NLP

- Data Cleaning
- Preprocessing of Data
- Tokenization
- Stemming
- Lemmitization

Data Science-Curriculum

CAPSTONE PROJECTS

1 **Hate-Speech Detection**

- **Data Preprocessing:** Cleaning text by removing URLs, punctuation, and stop words improves model accuracy by reducing noise and standardizing inputs.
- **Feature Extraction:** Use of feature like CountVectorizer to convert text to numerical form, enabling traditional models like Decision Trees to process and classify text data.
- **Model Evaluation:** Metrics like accuracy and confusion matrices provide insights into model performance, revealing strengths and areas for improvement.

2 **Spam SMS Classification**

- **Data Imbalance Management:** Learning effective techniques for handling imbalanced datasets, such as oversampling, to create balanced distributions for model training.
- **Exploratory Data Analysis (EDA):** Gain insights into spam vs. ham characteristics through visualizations like count plots and word count distributions, aiding feature selection.
- **Feature Engineering:** Create new features, like word count, to enhance model accuracy and better capture distinctions between spam and ham messages.

3 **Predicting Mortality of Heart Failure Patients**

- **Healthcare Data Analysis:** Gain expertise in analyzing healthcare datasets, focusing on identifying key indicators that influence heart failure outcomes.
- **Predictive Modeling Skills:** Develop skills in building and tuning predictive models to accurately assess mortality risks for heart failure patients.
- **Feature Importance and Interpretation:** Learn to interpret feature importance, helping in understanding which factors most impact patient survival predictions.

LIVE PROJECT

1 **Credit EDA**

- **Effective Missing Value Handling:** Gain insights into imputation methods to retain data integrity, enhancing model reliability.
- **Data Transformation Techniques:** Learn to create age and credit categories, making analysis of income, credit, and employment patterns clearer.
- **Correlation Analysis in Different Scenarios:** Identify critical differences in financial characteristics across approved, cancelled, and refused loan groups.