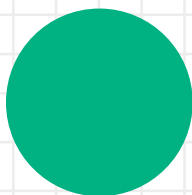
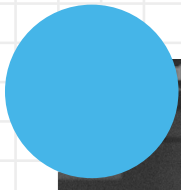




Data Academy

Comprehensive training in data and technology for diverse data careers





Overview

The TeamEpic Data Academy provides comprehensive and tailored training in the dynamic field of data and technology. Our carefully crafted curriculum caters to the diverse needs of individuals aspiring to excel in various data-driven careers. Whether graduates have ambitions of becoming a Data Scientist, Data Engineer, Prompt Engineer, or pursuing specialized roles, our academy equips them with the necessary training and resources to develop the essential skills.

At TeamEpic Data Academy, we follow a career-specific roadmap for each path, ensuring that our graduates gain a comprehensive understanding of their chosen field and emerge with the expertise required to thrive. Our intensive training program is designed to be completed within a four-month timeframe, with graduates dedicating 25-30 hours per week to course content and project work.

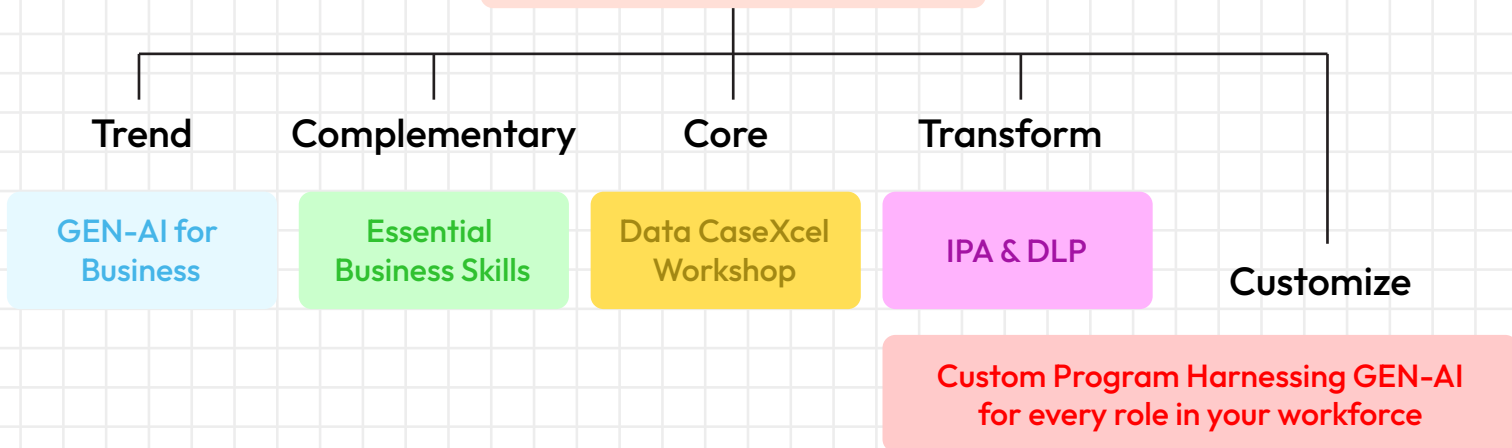
Our training approach combines technical prowess with a user-centric and problem-solving mindset. Graduates are equipped with structured frameworks and templates to analyze and devise innovative solutions, enabling them to effectively tackle complex business challenges. We also emphasize the utilization of cutting-edge technologies, including AI, ML, Hyper Automation, Microservices, and Cloud computing, to develop impactful solutions that drive tangible business outcomes.

In addition to technical training, the TeamEpic Data Academy places significant emphasis on developing essential business skills. Our graduates receive training in Design Thinking, Product Success, UI/UX, Next-Gen AI, and Storytelling, enabling them to grasp the broader context of their work and effectively communicate their ideas and solutions.

To deliver a comprehensive learning experience, we leverage various platforms such as the Content Engine, Learning Management System, and eBooks Platform. Our courses are structured in bite-sized modules, offering a blend of reading materials, recommended external courses, auto-evaluated assessments, and project opportunities. This approach ensures that our participants acquire theoretical knowledge while engaging in practical application.

In summary, the TeamEpic Data Academy offers a robust and meticulously designed training program that prepares individuals for successful careers in the data and technology industry. Our comprehensive curriculum, focus on practical application, and emphasis on both technical expertise and business acumen equip our graduates with the knowledge and capabilities needed to excel in today's data-driven landscape.

TE Academy



Key Features

The Skills Acquired

Over the course of four months, our grads learn the following:

- In-depth understanding of data analysis, engineering, and science principles for effective decision-making.
- Proficiency in leveraging AI, ML, and Automation technologies to solve complex business problems and drive innovation.
- Familiarity with Intelligent Process Automation (IPA) in a dynamic multi-Tech eBusiness environment, enabling graduates to optimize operational efficiency.
- Business skills development in critical areas such as Design Thinking, Product Success, UI/UX, Next-Gen AI, and Storytelling, empowering graduates to communicate effectively and deliver impactful solutions.
- Mastery of structured frameworks and templates for analyzing and ideating solutions, ensuring graduates possess a systematic approach to problem-solving.

The Learning Process

- Our online curriculum, meticulously curated by industry experts, provides a deep understanding of key AI concepts through a blend of comprehensive materials. These include engaging projects, insightful lectures, theory-based modules, coding exercises, and relevant reading/viewing exercises, along with career-focused coursework.
- Project-based learning forms a cornerstone of our program, allowing graduates to apply their knowledge through 12 mini-projects and 3 capstone projects. This practical approach hones their skills and solidifies their understanding of the learning process.
- Competency tracks offer graduates the opportunity to specialize in specific areas of expertise. By selecting unique skills within the AI domain, they can differentiate themselves as highly skilled professionals in their chosen field.

The Valuable Benefits

- **Personalized 1-on-1 Support:** Each graduate will be paired with a dedicated expert who will guide them through the curriculum, offer valuable feedback, and address their inquiries. This mentorship ensures personalized attention and provides an insider's perspective on the industry.
- **Career Development Assistance:** Our career coach will assist graduates in navigating career-specific modules, including strategic planning, resume and LinkedIn profile development, networking, and conducting mock interviews. This comprehensive support equips graduates with the necessary skills to excel in their professional journey.

Key Training Details



4 months
Training Length



25-30
Hours per week



15 total projects
12 mini projects 3 guided capstones



Tools & Skills

 python™

 pandas

SQL

 GitHub

 APACHE
Spark™

 + a b l e a u™

Power BI 


TensorFlow

 PyTorch

 Keras

 scikit
learn

 jupyter

Building a data career portfolio



Building a Data Career Portfolio

Required for any Data Career Path

Git - version control

Github

Gitlab

Semantic versioning

Keep a changelog

Notebooks

Jupyter Notebook

Google Collab

Project deployment

AWS Cloud

Microsoft Azure

Google Cloud

Heroku

Read Papers

Zeta Alpha

Papers with Code

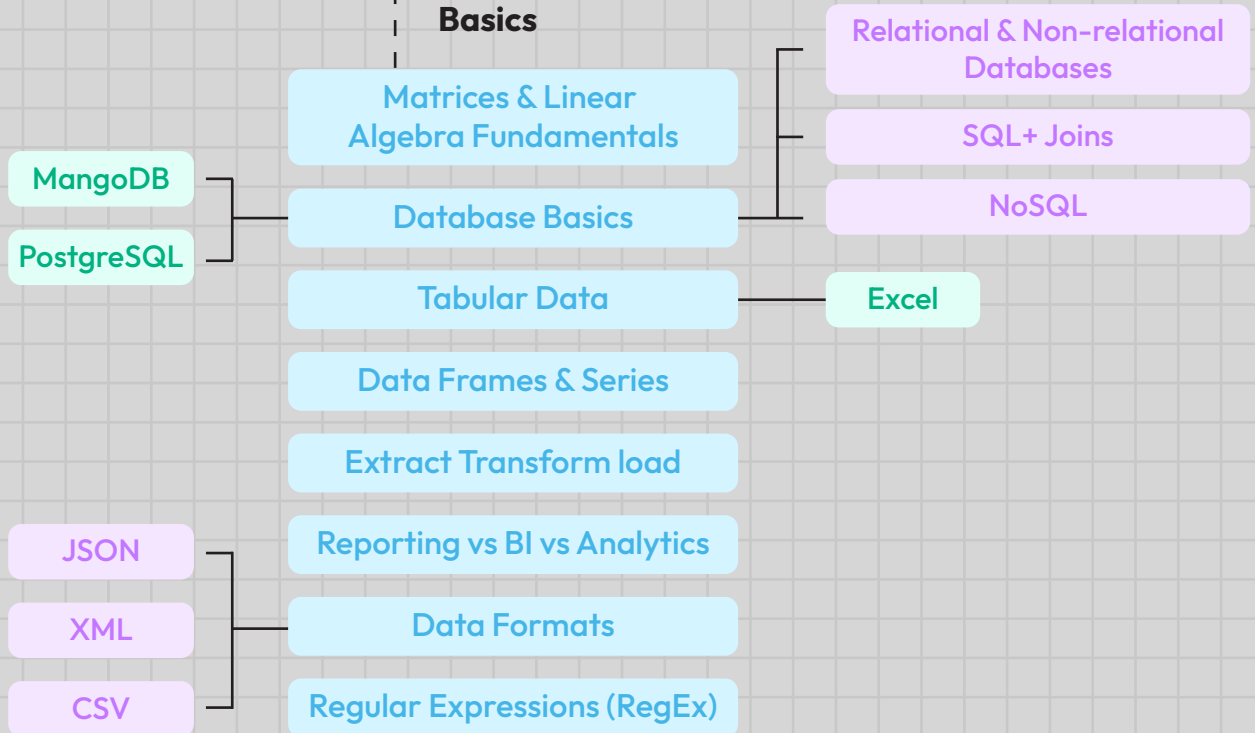
Fundamentals (Data Role)

Choose Your Path

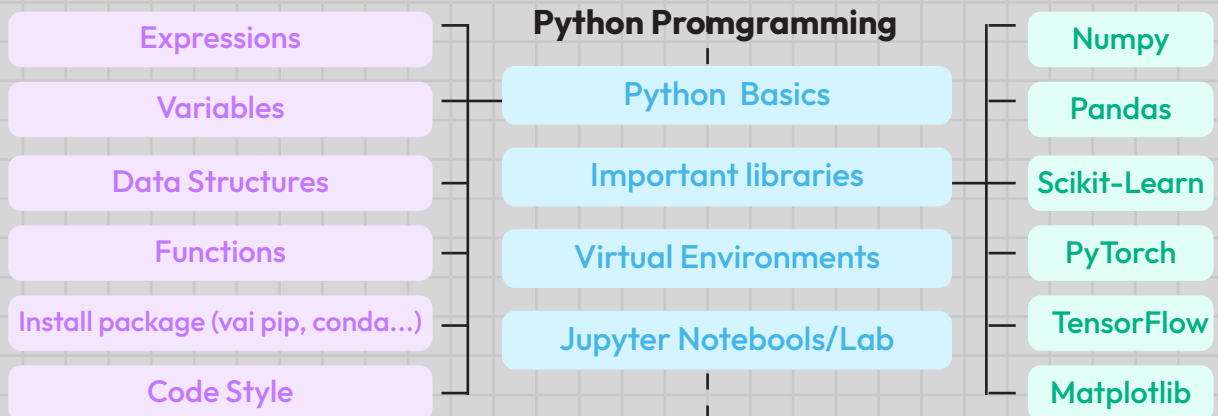
Analytics/AI Job

Fundamental (Data Role)

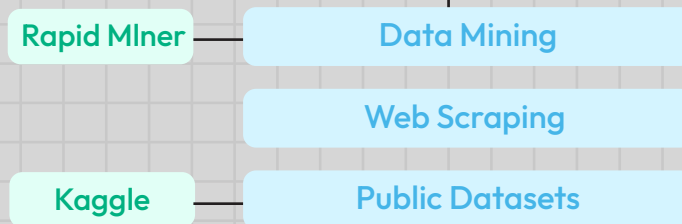
Basics



Python Programming



Data Sources



Exploratory Data Analysis

Principal Component Analysis

Dimensionality Reduction

Normalization

Data Cleaning, Handling Missing Values

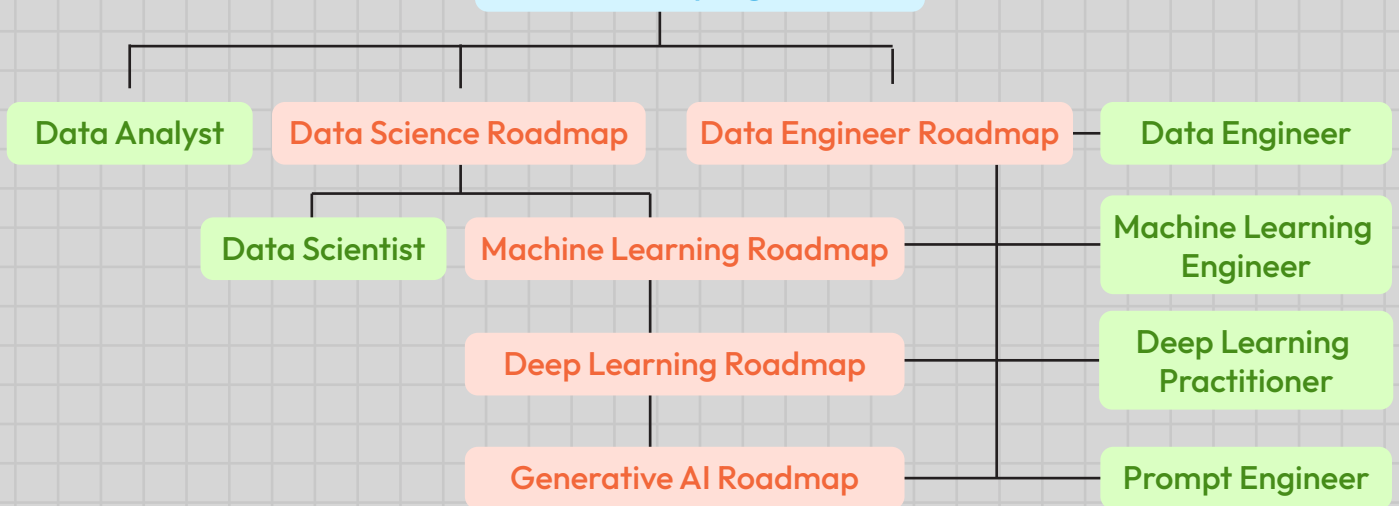
Estimators

Binning Sparse Values

Feature Extraction

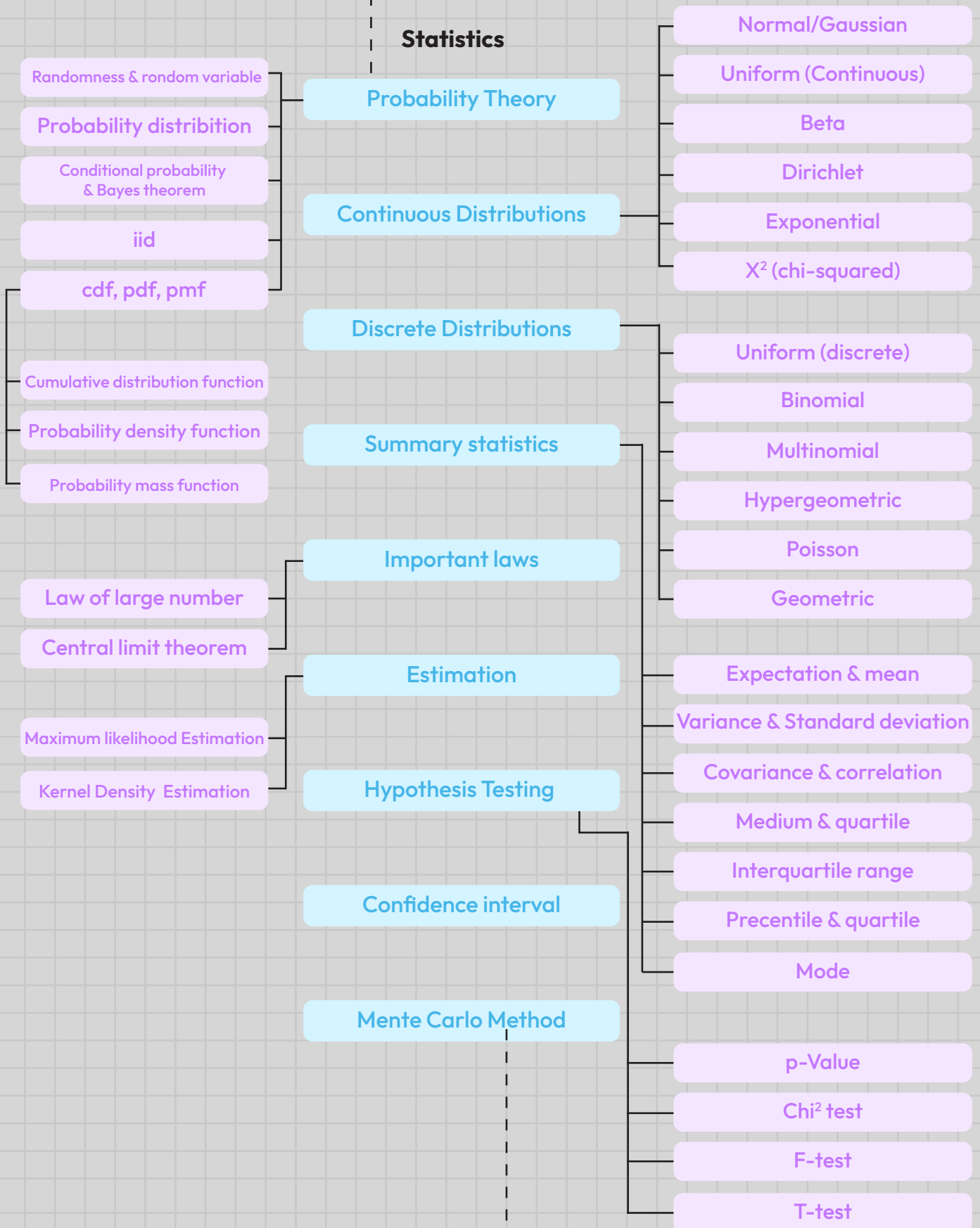
Denoising

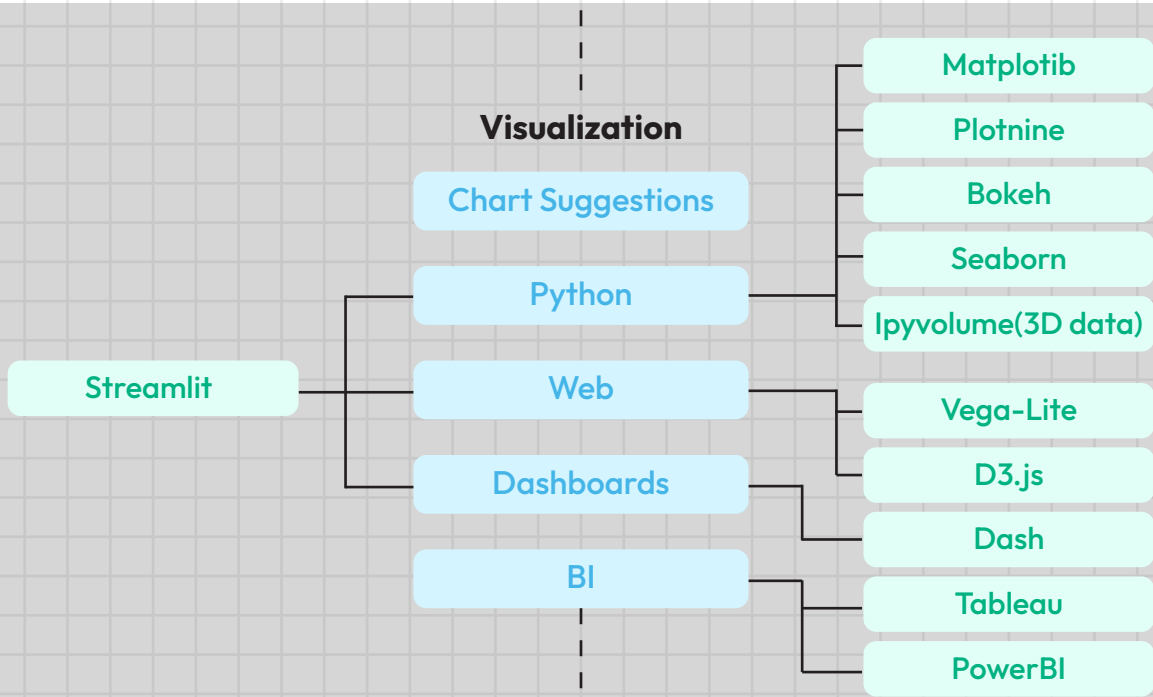
Sampling



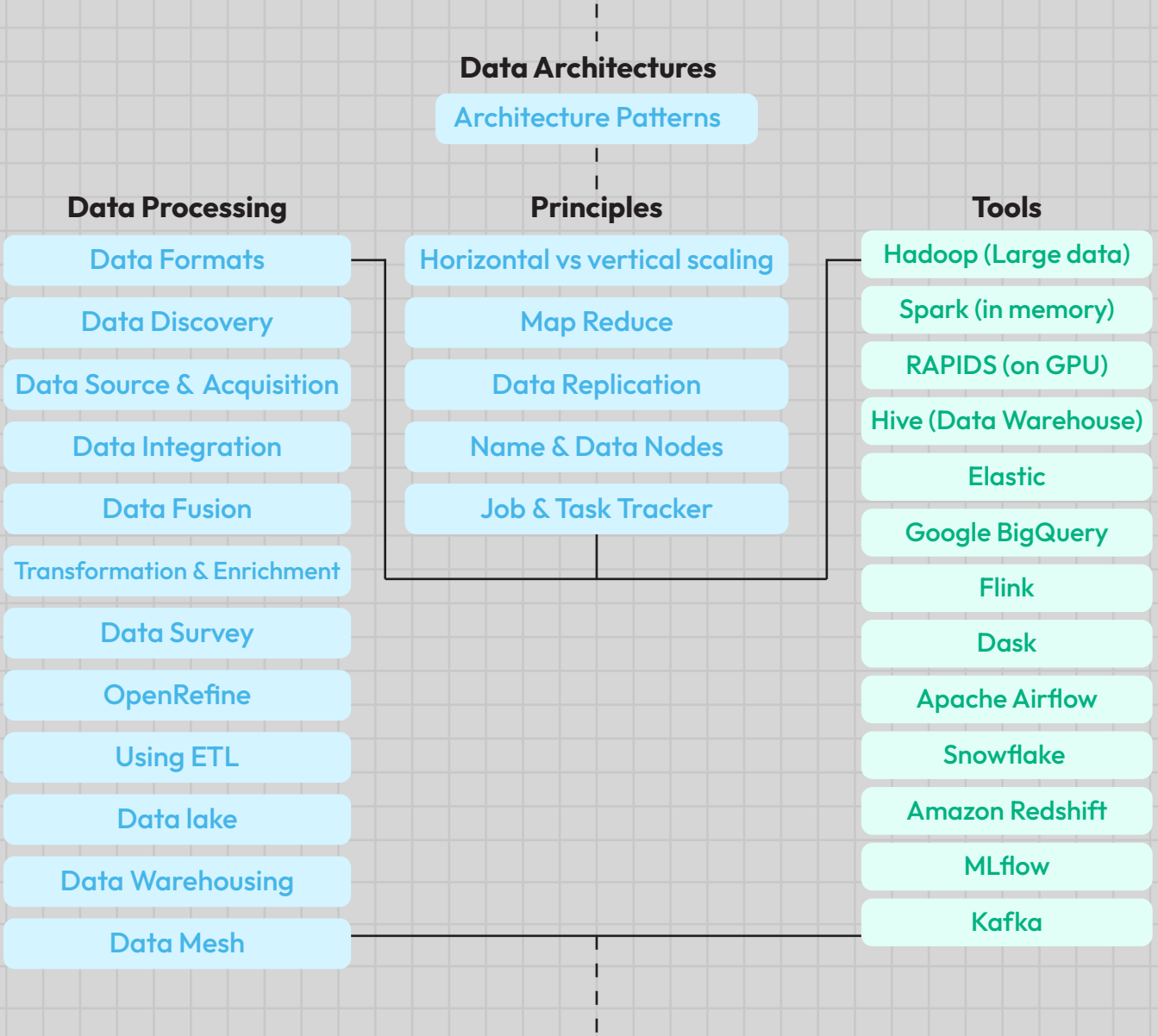
Data Scientist Road Map

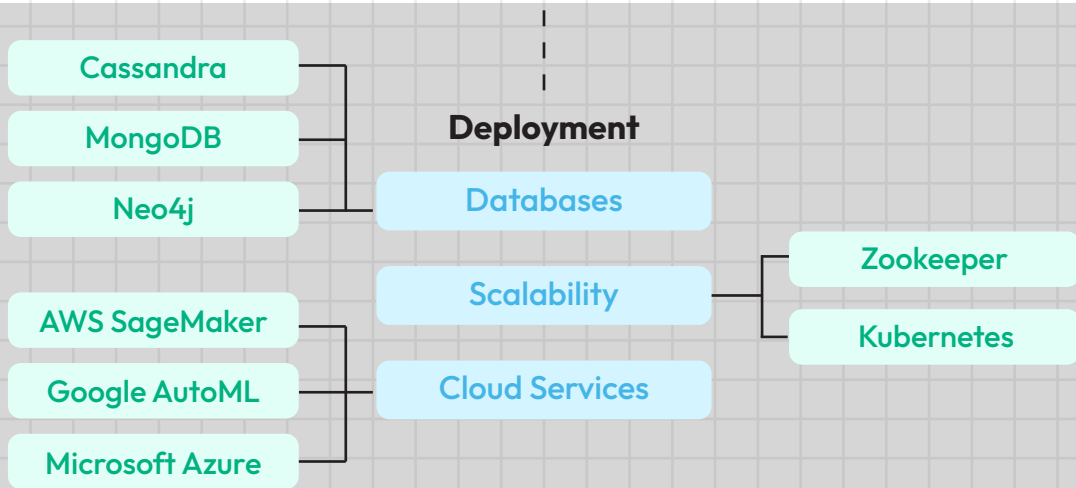
Statistics



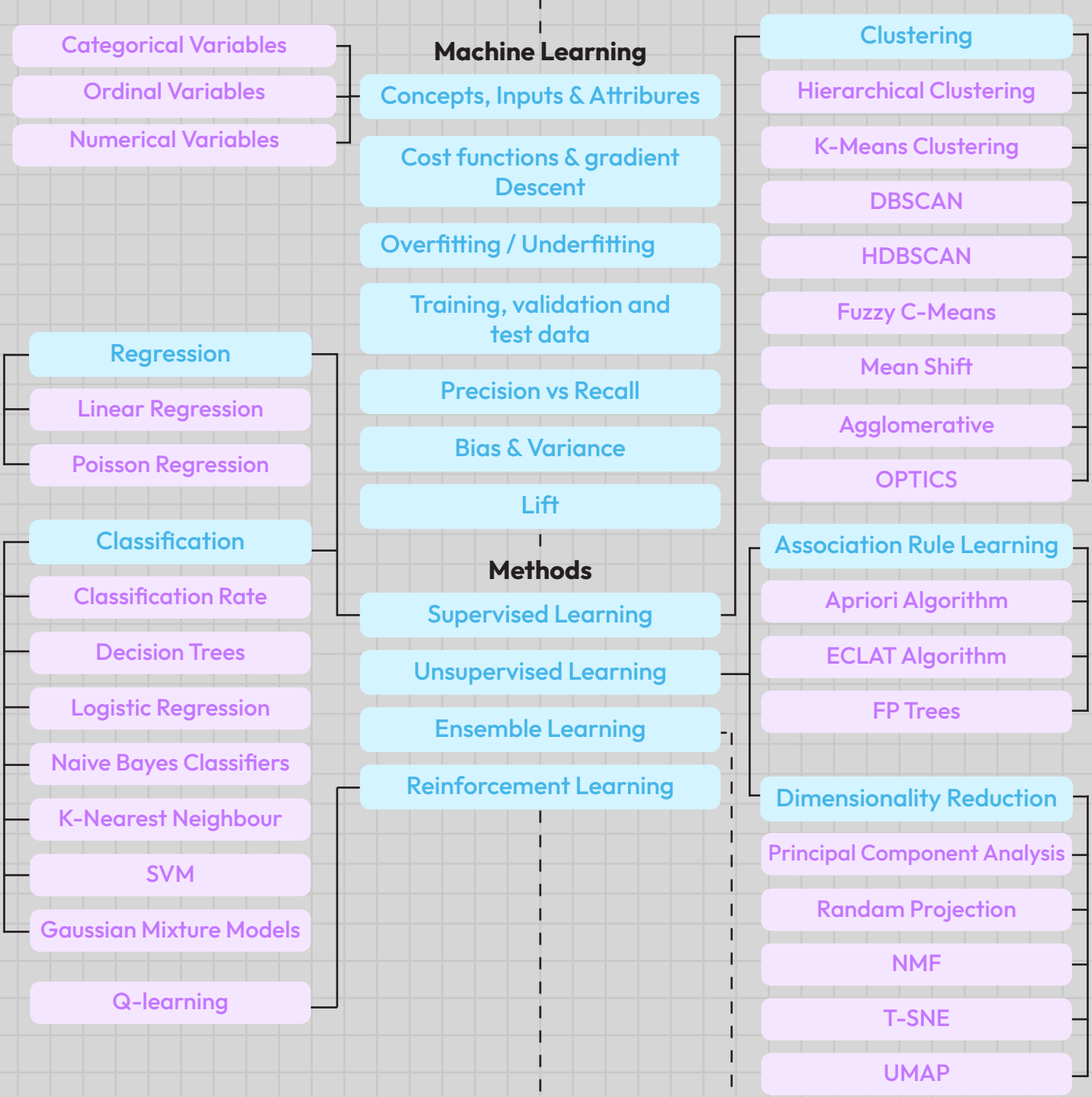


Data Engineer Road Map





Machine Learning Road Map



- Sentiment Analysis
- Collabarative Filtering
- Tagging
- Prediction

- Bagging
- Stacking
- Boosting

Machine Learning with Python

Framework

- Flask
- Django
- Keras
- Bottle
- CherryPy

Tools

- Docker
- Kubernetes
- Gradio
- MLflow

- Important libraries
- Model Deployment

- Scikit-Learn
- TensorFlow
- Spacy
- Pandas
- Numpy
- PyTorch
- Matplotlib

Deep Learning Road Map

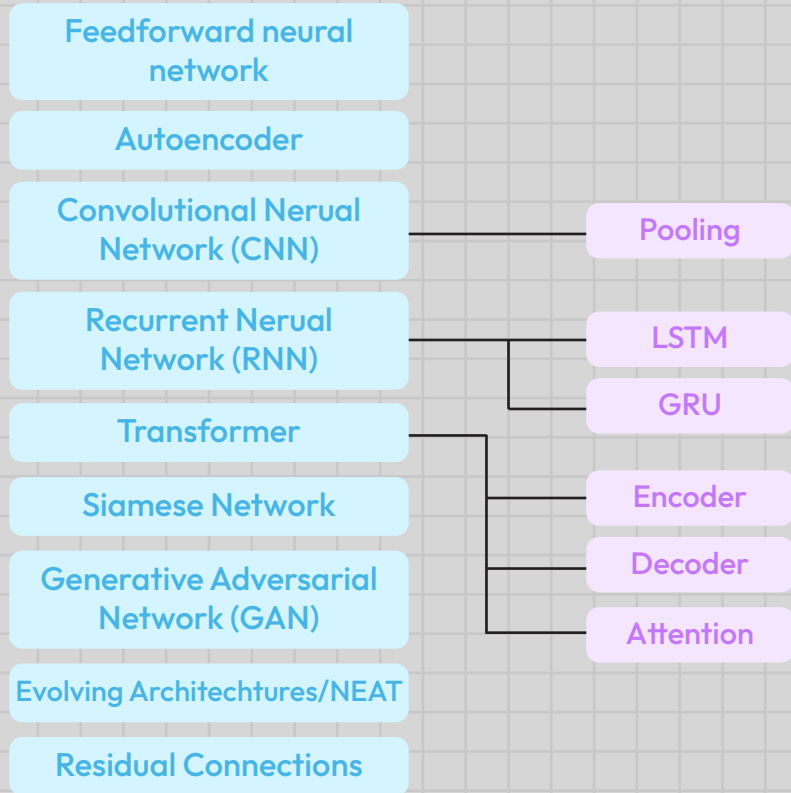
Papers

- Deep Learning Papers Reading Roadmap
- Zeta Alpha Search Engine for papers

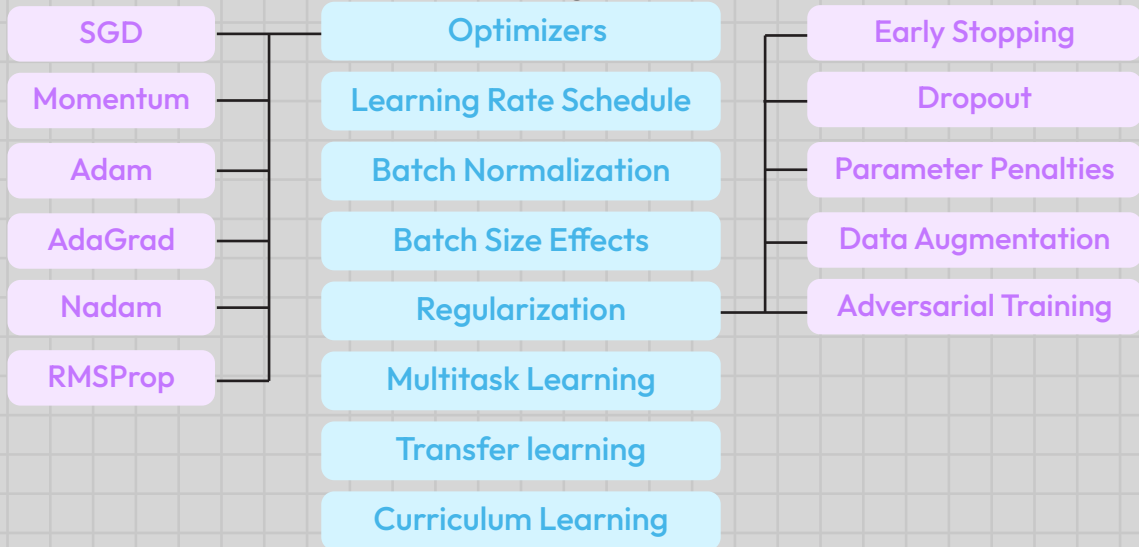
Neural Networks

- Understanding Neural Networking
- Loss Functions
- Activation Function
- Weight Initialization
- Vanishing / Exploding Gradient Problem

Architectures



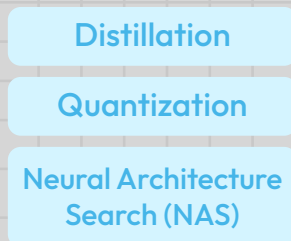
Training



Tools



Model optimization (Advanced)



Generative AI Road Map

Prompt Engineer

TE Data Academy: Competency Development

TE Training Plan 4 months

40 case-studies + 12 Mini Projects
+ 3 Capstone Projects

M1

Data Casexcel
Workshop-Core

M3

Gen-AI & Essential
Business Skills

M2

Data Casexcel
Workshop-Specialization

M4

IPA, DLP, Client-Specific
Training

Data CaseXcel Workshop

The Data CaseXcel offers a wide range of workshops designed with an experiential learning approach and a bite-size learning structure. Each course features case studies driven by AI-voiced videos, available in different modes such as Expert Videos, Scenario-based videos, Screen-cast videos, and Demo videos. In addition to the videos, there are audio podcasts, reading materials, auto-evaluated assessments and capstone projects to reinforce the learning.

Title	Workshop Duration	Recommended ILT Hours	Project Hours
TE Workshop for Data Analyst	240 Hours	20 Hours	100 Hours
TE Workshop for Data Engineer	240 Hours	20 Hours	100 Hours
TE Workshop for Data Scientist	240 Hours	20 Hours	100 Hours
TE Workshop for Machine Learning Expert	240 Hours	20 Hours	100 Hours
TE Workshop for MLOPs Engineer	240 Hours	20 Hours	100 Hours
TE Workshop for Deep Learning Specialist	240 Hours	20 Hours	100 Hours
TE Workshop for NLP Practitioner	240 Hours	20 Hours	100 Hours
TE Workshop for Prompt Engineer	240 Hours	20 Hours	100 Hours
TE Workshop for XAI Professional	240 Hours	20 Hours	100 Hours
TE Workshop for Generative AI Scientist	240 Hours	20 Hours	100 Hours
TE Workshop for AI Ethics Professional	240 Hours	20 Hours	100 Hours
TE Workshop for RPA Professional	240 Hours	20 Hours	100 Hours

Essential Business Skills

Business Skill Programs

The Business Skill Program offer a comprehensive courses designed to enhance professionals' expertise across various key areas. Participants will develop a deep understanding of design thinking, product success strategies, UI expertise, storytelling techniques, and the evolving landscape of next-gen marketing. Through this program, individuals will acquire a versatile skill set, empowering them to drive innovation, create captivating user experiences, and shape impactful marketing campaigns in today's dynamic business environment.

Course Title	Course Duration	Recommended ILT Hours
NP Certified Design Thinking Practitioner	50 Hours	20 Hours
NP Certified Product Success Leader	100 Hours	20 Hours
NP Certified UI/UX Expert	50 Hours	20 Hours
NP Certified Storytelling with Data Visualisation	100 Hours	30 Hours
NP Certified Next-Gen AI Marketer	50 Hours	10 Hours

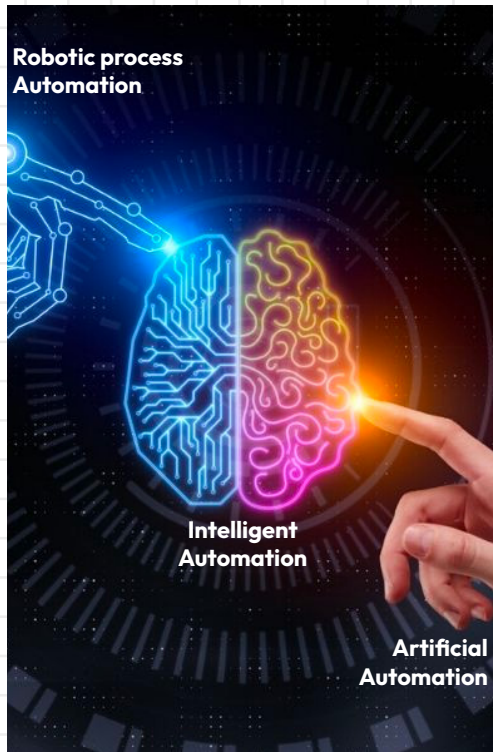
Gen-AI

Gen AI for Business

The Gen AI for Business program offers a wide range of courses that equip participants to navigate the AI landscape and unlock transformative opportunities in the rapidly evolving business landscape. Through a curated curriculum covering AI fundamentals, noteworthy applications, automation strategies, and ethical considerations, participants gain insights into harnessing Gen-AI for strategic decision-making, process optimization, and innovation.

Course Title	Course Duration	Recommended ILT Hours
Overview of Gen-AI: Imperatives for Digital Business	50 Hours	10 Hours
Discriminative Vs Generative AI for Business	50 Hours	10 Hours
Application Fields & Potential Trends	50 Hours	10 Hours
AI- driven Product Strategy (Product Analysis)	120 Hours	60 Hours
NLP & Chatbots	120 Hours	60 Hours
Computer Vision & Image Recognition	120 Hours	60 Hours
Reinforcement Learning for Business Optimization	120 Hours	60 Hours
Noteworthy GAN Applications: 3D Object/Interactive Image Generation, Data Augmentation	120 Hours	60 Hours
Managing AI Transformation & Automation	120 Hours	60 Hours
Gen AI: The Good, Bad and Ugly	50 Hours	10 Hours
Strategy to develop your own application using Gen AI	120 Hours	60 Hours

Up-Skilling Programs for Corporate Success



10 Live Hour

Intelligent Process Automation Program in a Multi-Tech eBusiness Environment

A program to help participants:

- Demonstrate a user-centric approach and solution mindset to problem-solving that is
- less reactive and more preventive and restorative
- Use a structured framework and templates to analyze, ideate solutions, resolve business problems
- Become aware of how technologies AI ML, Hyper Automation, Microservices, and Cloud computing can be leveraged for creating the best solution that will deliver business impact
- Relate and suggest the most appropriate technology for the solution keeping the end goal and user in mind



Digital Leader Program (Using Gen AI) in a Multi-Tech eBusiness Environment

A program to help participants:

- Discover the top digital disruptions that you can leverage for sustainable
- innovation
- Think about what new solutions might make sense for the future
- Consider and selectively adopt new technologies for business impact and growth
- Adopt new technologies for innovating and creating new value to business and clients

Project List

- **Product Recommendation for E-Commerce**
- **Customer LTV Prediction for Financial Institutions**
- **Mass Approval Detection and Resource Allocation**
- **Network Anomaly Detection using ML Technique**
- **Predictive Analytics for Sports**



GPT PROJECTS

- **Text Summarization**
- **Q & A – Ask Me Anything Bot**
- **Code Assistant**
- **Stack Overflow Assistant**
- **Mental Health Slack Bot**



Product Recommendation for E-Commerce

Goal

Implement a recommender system for an e-commerce store to provide personalized product recommendations using clustering, content-based filtering, and incorporating product reviews, and deploy the models in a web application.

Data

The project utilizes a dataset containing product information, including attributes, descriptions, and reviews. Additional data may include customer information, ratings, and historical purchase data. The data will be sourced from the e-commerce store's database or publicly available datasets.

Approach

- **Data Preparation & EDA**
- **Content-based Filtering**
- **Collaborative Filtering**
- **Neural-Network based Recommender**
- **Hybrid Recommender**

Extract and preprocess relevant data from the e-commerce store's database.

Perform exploratory data analysis (EDA) to gain insights into the dataset and understand product and customer preferences.

Utilize natural language processing techniques to analyze product attributes, descriptions, and other textual data.

Apply feature extraction methods to represent products in a meaningful feature space.

Develop a content-based recommendation system based on product attributes and descriptions to suggest similar items.

Use customer ratings and historical purchase data to build collaborative filtering models.

Implement user-based and item-based collaborative filtering techniques for personalized recommendations.

Explore advanced techniques like matrix factorization or neighborhood-based approaches to improve recommendation accuracy.

Implement deep learning approaches, such as neural networks, utilizing techniques like embedding layers, CNNs, or RNNs.

Create a hybrid recommender system by combining content-based filtering, collaborative filtering, and neural network-based approaches. Develop a mechanism to weight and combine recommendations for more diverse and accurate suggestions.

Tech Stack

- **Programming Languages:** Python 3.9, SQL
- **Libraries:** Pandas, Numpy, Matplotlib, Sci-kit-learn, PyCaret
- **Coding Environment:** Google Colab, Anaconda (Jupyter Notebook)
- **Web Framework:** FastAPI (for building the API)
- **Deep Learning Framework:** TensorFlow, Keras (for neural network-based recommender)
- **Database Management:** SQL (for data storage and retrieval)

Insights

This project enhances the user experience in an e-commerce store by providing personalized product recommendations. Through various recommendation techniques, including customer attributes, descriptions, reviews, ratings, and purchase data, the store tailors recommendations to customer preferences. The gained insights enable understanding of customer behavior, leading to enhanced satisfaction and increased sales.

Success Metrics

- **Recommendation Accuracy:** Measure the accuracy of the recommendation models by evaluating how well the recommended products align with customers' preferences and actual purchases.
- **User Engagement:** Track user engagement metrics, such as click-through rates, conversion rates, and time spent on the site, to assess the impact of personalized recommendations on user behavior.
- **Customer Satisfaction:** Gather feedback from customers to gauge their satisfaction with the recommendations received and their overall shopping experience.
- **Business Impact:** Measure the impact of the recommender system on key business metrics, such as sales revenue, average order value, and customer retention.
- **API Performance:** Monitor the performance of the deployed web application, including response times, uptime, and user feedback, to ensure a seamless user experience.

Customer LTV Prediction for Financial Institutions

Goal

Develop a predictive model for financial institutions to estimate loan-to-value (LTV) ratios using behavioral machine learning techniques. The model will be deployed as an API for international financial institutions as a credit risk tool.

Data

Demography Table: Contains customer demographic information.
Bureau Data: Includes data from credit bureaus and other relevant financial sources.

Contributor ID: Identifies the type of financial institution providing the loan.

Bank Name: Name of the bank associated with the account.

Account Type: Indicates the type of loan or credit facility.

Description: Additional description or category for the account type.

Approach

- **Data Collection**
- **Data Preparation**
- **Feature Engineering**
- **Model Development**
- **Model Validation**
- **Model Deployment**

Gather relevant customer data from the demography table, bureau data, and additional sources.

Clean and transform the data, handle missing values, and address any data quality issues.

Extract relevant features from the data to enhance the predictive power of the model.

Utilize behavioral machine learning techniques to train a predictive model for LTV estimation.

Assess the performance of the model using appropriate evaluation metrics and validation techniques.

Create an API to deploy the trained model, allowing users to access LTV predictions.

EDA | ML (Regression) | Feature Selection | Model Evaluation (MAE, R2) | Model Enhancement (Ensemble Techniques, Finetune-hyperparameters)

Tech Stack

Programming Languages:
Python

Libraries:
Pandas, Numpy, Matplotlib, Seaborn, Sci-kit-learn, PyCaret

Coding Environment:
Google Colab, Anaconda (Jupyter Notebook)

Web Framework:
Flask (for building the API)

Deep Learning Framework:
TensorFlow, Keras (for neural network-based recommender)

Insights:

This project analyze and model the loan-to-value (LTV) ratio, a vital credit risk indicator for financial institutions. By uncovering influential factors and patterns through data analysis, it enables informed lending decisions and effective credit risk management.

Success Metrics

Prediction Accuracy: Measure the accuracy of the LTV predictions by comparing them to actual LTV ratios.

Model Performance: Evaluate the model's performance using appropriate metrics such as mean absolute error or R-squared.

Business Impact: Assess the impact of deploying the LTV prediction model on financial institutions, considering factors like improved risk management, cost reduction, and enhanced decision-making processes.

User Feedback: Gather feedback from users of the deployed API to gauge its usefulness and effectiveness in real-world scenarios.

Mass Approval Detection and Resource Allocation

Goal

The project aims to detect mass approvals in the IT/software request approval process and optimize resource allocation. By analyzing log data, it identifies suspicious approval patterns and recommends default resources and applications based on individual requirements.

Data

Log data containing information about software/IT requests, including requestee details, request IDs, request types, request status, account names, entitlements, approval types, approvers, approval statuses, and timestamps. Additionally, user data will be used, including employee IDs, employee types, departments, divisions, and HR organization codes.

Approach

- **Timeline of Development Phase**
- **EDA**
- **Data Preprocessing**
- **Mass Approval Detection**
- **Recommendation System**

Define development timeline for tasks: data collection, preprocessing, model development, and evaluation.

Perform exploratory data analysis on log data to gain insights and identify patterns or anomalies.

Analyze distribution of approval types, statuses, and request types for overall understanding.

Clean and preprocess log data, handling missing values, normalization, and feature engineering.

Merge log data with user data based on employee IDs to enrich the dataset.

Define criteria for identifying mass approvals, such as high number of approvals by the same approver within a time period or for the same request type.

Develop algorithms to detect suspicious approval patterns and flag potential mass approvals.

Apply machine learning techniques like clustering or anomaly detection for identifying unusual approval behavior.

Build a recommendation system to suggest default resources and applications based on employee requirements and roles.

Utilize employee data to understand needs and preferences, and apply collaborative or content-based filtering techniques for personalized recommendations.

Tech Stack

Programming Languages:
Python

Libraries:
Pandas, Numpy, Matplotlib,
Seaborn, Sci-kit-learn, PyCaret

Algorithms: Clustering, Anomaly
detection

Coding Environment:
Google Colab, Anaconda
(Jupyter Notebook)

Web Framework:
Flask/Django (for building the API)

Deep Learning Framework:
TensorFlow, Keras (for neural
network-based recommender)

Insights:

This project enhances resource allocation and improves the approval process in the IT/software domain. By analyzing log data and detecting mass approvals, it ensures accuracy and integrity. The recommendation system suggests customized default resources and applications, leading to improved efficiency and reduced unauthorized access.

Success Metrics

Mass Approval Detection Accuracy: Measure the accuracy of the mass approval detection algorithm by comparing the flagged mass approvals with known instances of suspicious approval patterns.

Resource Allocation Efficiency: Evaluate the efficiency of the recommendation system by tracking the utilization of default resources and applications and assessing whether they align with individual requirements.

Reduction in Unauthorized Access: Measure the reduction in unauthorized access to resources by monitoring approval patterns and tracking instances where inappropriate approvals are avoided.

User Satisfaction: Gather feedback from employees regarding the usefulness and relevance of the recommended default resources and applications to assess user satisfaction and adoption.

Network Anomaly Detection using ML Technique

Goal

The project aims to develop a machine learning model for network anomaly detection to differentiate normal network connections from malicious attacks. By accurately classifying connections, the model enhances network security by identifying and preventing unauthorized access and intrusions.

Data

The project will utilize a dataset containing network connection information. Each connection is labeled as either "normal" or a specific type of attack. The dataset includes features such as source IP address, target IP address, protocol, and TCP packet information.

Approach

- **Overview, Scope & Objectives**
- **Workflow**
- **EDA**
- **Proposed Solution**
- **Model Building & Evaluation**

Define the problem statement and emphasize the importance of network anomaly detection for network security.

Specify the project scope and objectives to be achieved.

Outline the high-level workflow, including data preprocessing, feature engineering, model development, and evaluation.

Perform exploratory data analysis on the network connection dataset.

Analyze the distribution of normal connections and different types of attacks.

Identify patterns and anomalies in the data by exploring feature relationships.

Select appropriate machine learning techniques and algorithms for network anomaly detection.

Preprocess the data by applying normalization or scaling methods.

Determine relevant features and evaluate their importance for model development.

Split the dataset into training and testing sets and train a predictive model using algorithms like decision trees, random forests, or support vector machines.

Tech Stack

Programming Languages:
Python 3.9

Libraries:
Pandas, Numpy, Matplotlib,
Seaborn, Sci-kit-learn, PyCaret

Algorithms: Decision Trees,
Random Forest, SVM

Data Preprocessing:
Normalization, Feature Scaling

Model Evaluation: Accuracy,
Precision, Recall, F1 Score

Insights

By developing a machine learning model, the project identifies anomalies and unauthorized access attempts, providing valuable insights for network administrators to implement security measures and protect computer networks.

Success Metrics

Model Accuracy: Measure the accuracy of the machine learning model in classifying network connections.

Attack Detection Rate: Evaluate the model's performance in detecting different types of attacks.

False Positive Rate: Assess the rate of incorrectly classifying normal connections as attacks.

False Negative Rate: Evaluate the rate of incorrectly classifying actual attacks as normal connections.

Robustness and Efficiency: Assess the model's robustness through cross-validation and evaluate its computational efficiency for real-time or near-real-time detection.

Predictive Analytics for Sports

Goal

The project aims to create a player performance tracking and evaluation system for a sports franchise, initially focusing on cricket. The system allows the organization to monitor player performance, select players for auctions, evaluate skills, identify development areas, and track training routines.

Data

The project will utilize a dataset containing cricket player information, including performance metrics, player skills, training routines, and development areas. The dataset may include features such as batting average, strike rate, bowling average, fielding statistics, player rankings, and match-specific data.

Approach

- **Problem Statement**
- **Player Analysis**
- **Preliminary Model Development**
- **Model Enhancement**
- **Dashboard**

Define problem statement and project objectives for cricket player performance tracking and evaluation system.

Plan to expand the system to other sports teams in the future.

Perform data preprocessing tasks: cleaning, handling missing values, and transformation.

Explore and analyze dataset, address data quality issues.

Analyze performance metrics like strike rate, overall value, and critical situation handling.

Identify top performers based on batting average, bowling average, and fielding statistics.

Conduct statistical analysis to identify correlations between performance metrics and player skills.

Develop preliminary model using machine learning techniques for player performance prediction.

Evaluate model performance using appropriate metrics and cross-validation.

Develop interactive dashboard for visualizing player performance, evaluation results, and insights.

Tech Stack

Programming Languages:
Python 3.9

Libraries:
Pandas, Numpy, Matplotlib,
Seaborn, Plotly, Tableau

Algorithms: Regression,
Classification, Ensemble Methods,
Deep Learning

Deep Learning Framework:
TensorFlow, Keras (for neural
network-based recommender)

Insights

This project provides insights into cricket player performance, helping the sports franchise organization make informed decisions regarding player selection, auctions, and player development. The insights gained from the system optimize team composition and enhance player performance.

Success Metrics

Prediction Accuracy: Measure the accuracy of the player performance prediction model in estimating various performance metrics, such as batting average, strike rate, or bowling average.

Player Ranking: Evaluate the model's ability to rank players based on their overall value and suitability for auctions.

Development Area Identification: Assess the model's effectiveness

GPT PROJECTS

Text Summarization

Goal

The goal of this project is to develop an AI application that can summarize long and complex text documents into concise and easy-to-understand summaries. The application will leverage NLP techniques to automatically generate summaries, saving time and effort for users who need to process large volumes of textual information.

Tech Stack

OpenAI API (GPT-3.5)

Python

Langchain (language modeling framework)

VectorDB (database management system)

Data

The project will utilize a diverse dataset consisting of long text documents from various domains. The dataset include articles, research papers, news reports, or any other type of text that requires summarization.

Insights

The project develops an AI solution for automated text summarization, aiding users dealing with large volumes of text. By summarizing complex documents, the application extracts key information swiftly. It showcases NLP techniques, particularly text summarization, and highlights the capabilities of language models like GPT-3.5.

Approach

- **Problem Statement**
- **Data Collection & Preparation**
- **Text Summarization**
- **Model Development**
- **Application Deployment**

1. Define the problem statement and project objectives for developing a text summarization AI application.
2. Discuss the importance and use cases of text summarization in various domains.
3. Collect a diverse dataset of long text documents suitable for summarization.
4. Preprocess the data by cleaning, normalizing, and transforming the text.
5. Explore extractive and abstractive text summarization techniques.
6. Implement extractive techniques to identify important sentences and create summaries.
7. Implement abstractive techniques to generate summaries by understanding the context.
8. Utilize OpenAI's GPT-3.5 model and fine-tune it with the dataset.
9. Use Langchain language modeling framework for development and integration.
10. Build a user-friendly interface or integrate the summarization model into an existing application, leveraging VectorDB for efficient data storage and retrieval.

Success Metrics

Summary Accuracy: Evaluate the accuracy of the generated summaries by comparing them with human-generated summaries or reference summaries.

Coherence and Readability: Assess the coherence and readability of the generated summaries by analyzing factors such as grammar, sentence structure, and flow of information.

Time Efficiency: Measure the time saved by using the text summarization application compared to manually reading and summarizing the original text.

User Feedback: Gather user feedback to assess the usefulness and effectiveness of the text summarization application in real-world scenarios.

Q & A – Ask Me Anything Bot

Goal

Develop an AI chatbot capable of accurately and contextually answering a wide range of queries on various topics. The chatbot should provide human-like conversational experiences and demonstrate a high level of natural language understanding.

Data

Collect and preprocess a diverse dataset of questions and answers from various domains.

Augment the dataset with relevant online resources, such as FAQs, forums, and knowledge bases

Incorporate user interactions and feedback to improve the chatbot's performance over time

Approach

- **Problem Statement**
- **Data Collection & Preparation**
- **Model Development**
- **Application Deployment**

Design and train a deep learning model using a sequence-to-sequence architecture.

Utilize transformer-based models for their ability to handle long-range dependencies and capture context.

Employ transfer learning techniques to leverage pre-trained language models like GPT-3.5 to bootstrap the chatbot's knowledge.

Implement a retrieval-based or generative-based approach, depending on the nature of the questions and available resources.

Natural Language Processing (NLP): Apply NLP techniques to preprocess and clean the input data, including tokenization, stemming, and lemmatization.

Named Entity Recognition (NER): Extract important entities from user queries to provide more relevant responses.

Intent Classification: Classify user intents to better understand the purpose of the queries and generate appropriate responses.

Sentiment Analysis: Analyze user sentiment to tailor responses accordingly, ensuring a more personalized interaction.

Tech Stack

Programming Languages: Python for model development and training

Deep Learning Frameworks: TensorFlow or PyTorch for building and training the chatbot model.

Web Development: HTML, CSS, and JavaScript for creating a user-friendly chatbot interface.

Deployment: Docker containers and cloud platforms like AWS or Google Cloud for hosting the chatbot.

Insights

Monitor and analyze user interactions and feedback to identify areas for improvement.

Track user satisfaction levels and sentiment analysis to assess the chatbot's performance.

Analyze conversational data to identify patterns, common queries, and user preferences for continuous enhancement.

Success Metrics

Accuracy: Measure the chatbot's ability to provide correct answers to user queries.

Contextual Understanding: Evaluate how well the chatbot comprehends user intent and context.

User Satisfaction: Assess user feedback and ratings to determine satisfaction levels.

Response Time: Measure the chatbot's speed in providing responses to ensure real-time processing.

Scalability: Evaluate the chatbot's performance under increased user load and concurrent interactions.

Code Assistant

Goal

Develop an AI assistant that can understand, debug, and assist with code, aiming to increase efficiency and solve programming problems. The assistant should provide accurate suggestions, identify bugs, and offer solutions to improve code quality.

Data

Collect a diverse dataset of code snippets, programming problems, and their corresponding solutions.

Augment the dataset with code repositories, online programming forums, and open-source projects to capture real-world code scenarios.

Include code annotations, comments, and documentation to enhance the assistant's understanding of programming language semantics.

Approach

- **Natural Language Processing**
- **Semantic Analysis**
- **Symbolic Execution**
- **Code Generation**

Design and train a deep learning model using techniques like sequence-to-sequence modeling or code2vec.

Incorporate techniques from program synthesis and code analysis to enable the assistant to comprehend and reason about code.

Integrate the assistant within popular development environments to provide real-time code suggestions, debugging assistance, and contextual documentation.

Implement a retrieval-based or generative-based approach, depending on the nature of the questions and available resources.

Apply NLP techniques to preprocess and tokenize code snippets, enabling the model to understand code structure and semantics.

Utilize techniques such as abstract syntax trees (ASTs) and type inference to capture code semantics and enable code comprehension.

Employ symbolic execution to analyze code paths, identify potential bugs, and provide relevant suggestions for improvement.

Use program synthesis techniques to automatically generate code snippets or refactor existing code based on user requirements.

Tech Stack

Programming Languages: Python for model development and training

Deep Learning Frameworks: TensorFlow or PyTorch for building and training the chatbot model.

Web Development: HTML, CSS, and JavaScript for creating a user-friendly chatbot interface.

Deployment: Docker containers and cloud platforms like AWS or Google Cloud for hosting the chatbot.

Insights

Analyze user interactions and feedback to identify common programming challenges and areas for improvement.

Monitor code quality metrics, such as code complexity and bug occurrences, to assess the assistant's impact on code improvement.

Track user productivity metrics, such as time saved or reduced debugging time, to measure the assistant's efficiency.

Success Metrics

Code Understanding: Measure the assistant's ability to accurately comprehend code snippets and programming problems.

Bug Detection: Evaluate the assistant's capability to identify and suggest fixes for code bugs.

Code Quality Improvement: Assess the impact of the assistant on improving code quality, including metrics like reduced code complexity and improved performance.

User Satisfaction: Gather user feedback and ratings to determine satisfaction levels with the assistant's suggestions and assistance.

Time Saved: Measure the time saved by developers through the assistant's efficient code suggestions and debugging assistance.

Stack Overflow Assistant

Goal

Develop an AI tool that can navigate Stack Overflow's knowledge base and provide relevant solutions to user queries. The tool should effectively retrieve and present answers from Stack Overflow to assist users in finding solutions to their programming problems.

Data

Utilize Stack Overflow's publicly available data, including question and answer content, tags, user profiles, and voting data.

Preprocess and clean the data, removing irrelevant or duplicate content while preserving the contextual information.

Approach

- **Natural Language Processing**
- **Information Retrieval**
- **Semantic Matching**
- **Question Classification**
- **Intent Recognition**

Design and implement a search system that leverages natural language processing and information retrieval techniques to match user queries with relevant content from Stack Overflow.

Apply NLP techniques for query preprocessing, tokenization, and entity recognition.

Utilize techniques such as TF-IDF, BM25, or word embeddings for efficient retrieval of relevant content from Stack Overflow's knowledge base.

Employ deep learning-based models or transformer models for improved semantic understanding and matching of user queries with relevant answers.

Implement models or techniques to categorize user queries into relevant programming topics or domains.

Utilize techniques to determine the intent behind user queries, enabling more accurate retrieval of relevant content.

Tech Stack

Programming Languages: Python for model development and training

Deep Learning Frameworks: TensorFlow or PyTorch for building and training the chatbot model.

Web Development: HTML, CSS, and JavaScript for creating a user-friendly chatbot interface.

Deployment: Docker containers and cloud platforms like AWS or Google Cloud for hosting the chatbot.

Insights

Analyze user interactions and feedback to understand common programming challenges and improve the relevance of retrieved answers.

Monitor user satisfaction with the provided solutions to identify areas for improvement.

Track the effectiveness of the tool in reducing the time and effort required for users to find solutions on Stack Overflow.

Success Metrics

Accuracy: Measure the tool's ability to provide relevant and accurate solutions to user queries.

Relevance: Evaluate the relevance of the retrieved answers based on user feedback and ratings.

User Satisfaction: Assess user feedback to determine satisfaction levels with the tool's performance and solutions provided.

Time Saved: Measure the time saved by users in finding solutions compared to manual search on Stack Overflow.

Effectiveness: Evaluate the tool's effectiveness in improving productivity and reducing the effort required to find solutions.

Mental Health Slack Bot

Goal

Develop an AI bot for Slack that can recognize signs of mental health issues in text conversations and provide appropriate resources, coping strategies, and supportive messages. The bot should aim to assist users in managing their mental health and provide valuable support.

Tech Stack

Programming Languages: Python for model development and integration with slack

Deep Learning Frameworks: TensorFlow or PyTorch for building and training the AI models.

NLP Libraries: NLTK (Natural Language Toolkit), spaCy, or Hugging Face's Transformers for NLP preprocessing and analysis.

Slack API: Utilize Slack's API to integrate the AI bot within the platform and enable real-time analysis of text conversations.

Deployment: Docker containers and cloud platforms like AWS or Google Cloud for hosting and deploying the AI bot.

Data

Collect a diverse dataset of text conversations that include discussions related to mental health.

Annotate the dataset with labels indicating the presence of mental health issues, emotions, and relevant resources or coping strategies.

Ensure the dataset is properly anonymized and follow strict privacy protocols to protect user information.

Insights

Monitor and analyze user interactions and feedback to identify areas for improvement and assess the bot's effectiveness in providing support.

Collaborate with mental health professionals to gain insights and guidance in developing appropriate responses and coping strategies.

Continuously update the bot's knowledge base with the latest mental health resources and best practices.

Approach

- **Natural Language Processing**
- **Sentiment Analysis**
- **Emotion Detection**
- **Topic Modeling**
- **Contextual Understanding**

Design and train a deep learning model using techniques like sentiment analysis, emotion detection, and topic modeling to understand the mental health context of text conversations.

Employ transfer learning approaches by utilizing pre-trained language models like GPT-3.5 to bootstrap the bot's understanding of mental health-related language.

Integrate the AI model within the Slack platform to allow real-time analysis of text conversations and provide appropriate responses.

Utilize NLP techniques to preprocess and tokenize text conversations, including sentiment analysis, emotion detection, and named entity recognition.

Determine the overall sentiment of text conversations to identify signs of distress or negative emotions.

Recognize and classify emotions expressed in text conversations to gauge the user's emotional state.

Identify key topics or issues related to mental health in text conversations to provide targeted resources and coping strategies.

Utilize techniques like deep learning and transformer models to capture the context and nuances of mental health discussions.

Success Metrics

Accuracy: Measure the bot's ability to accurately identify signs of mental health issues and emotions in text conversations.

Resource Effectiveness: Evaluate the relevance and usefulness of the provided mental health resources and coping strategies.

User Satisfaction: Assess user feedback and ratings to determine satisfaction levels with the bot's support and responses.

Impact: Track the bot's impact on users' well-being, such as increased awareness, improved coping skills, or seeking further professional help.

Privacy and Ethics: Ensure strict adherence to privacy protocols and ethical considerations to protect user data and maintain confidentiality.

Case Studies

Business Cases Repository

Team Epic utilizes open access business case repositories to drive knowledge sharing and innovation. We curate a wide range of real-world business case studies, enabling learners to access practical insights and enhance critical thinking skills. With a user-friendly platform, Team Epic promotes active learning and problem-solving.



Course Title	No of Case Studies
Acadia Institute of Case Studies	10
Case Centre	25
NP Certified UI/UX Expert	10
Ethics Wnrapped	15
Insead Publishing	20
Ivey Business Schools	5
Journal of Business Cases & Applications	25
Stanford Graduate School of Business Cases	20
UBC Open Cases	10
Yale School of Management	15
Corporate Case Sites	30



TeamEpic

Data Academy

www.teamepic.in

