Python

1. Beginner Level

Basic Concepts

- Data type conversions
- String operations and methods
- Input/output operations (print, input)

Control Structures

- if-elif-else conditional statements
- for and while loops
- break, continue and pass statements
- Building simple algorithms

Data Structures

- Lists and list methods
- Tuples
- Dictionaries
- Sets

Functions

- Defining and calling functions
- Parameters and return statements
- Variable scope

- Lambda functions

Application Projects

- **Smart Shopping List**: An application that categorizes the user's shopping items, tracks prices and offers budget analysis. Will save data with file operations

2. Middle Level

Object Oriented Programming (OOP)

- Classes and objects
- Inheritance (inheritance)
- Encapsulation
- Polymorphism
- Abstract classes and methods

Modules and Packages

- Standard library modules (os, sys, datetime, math)
- Module creation and import
- package management with pip
- Use of virtual environment

File Operations

- File reading and writing
- CSV, JSON operations
- Index operations

Error Management

- try, except, finally blocks
- Custom error generation
- Debugging techniques

Application Projects

- **Personal Finance Manager**: A system that tracks income and expenses using OOP, generates graphical reports and offers savings suggestions. Modular structure that stores data in JSON format and can offer data analysis

3. Advanced Level

Advanced Python Features

- Decorators
- Generators and iterators
- Context managers
- Regular expressions
- Concurrency (threading, multiprocessing)
- Asynchronous programming (asyncio)

Database Programming

- Working with SQLite, MySQL or PostgreSQL
- ORM usage (SQLAlchemy)
- Database design and optimization

Testing and Quality Assurance

- Unit testing (unittest, pytest)
- TDD (Test Driven Development)
- Code quality tools (pylint, flake8)
- Documentation (docstrings, Sphinx)

Application Projects

- **Data Analytics Automation**: A comprehensive system that pulls data from different sources (CSV, web API), cleans, analyzes and generates automated reports. It will include

asynchronous operations and data visualization. Can also make simple predictions with ML algorithms

4. Ways to Specialize

Basic Level

Data Processing and Analysis

- Pandas basic operations (data loading, cleaning, filtering)
- **Polars** input - high-performance data processing
- Scientific computing with NumPy
- SQL queries with**DuckDB** and Pandas integration

Visualization Basics

- Using Matplotlib and Seaborn
- Drag-and-drop analysis with**PyGWalker**
- Automated data discovery with**Lux**

Statistics and ML Fundamentals

- Basic statistical concepts
- Introduction to Scikit-learn (regression, classification)
- Model evaluation metrics
- Low-code ML with**PyCaret**

Project

Data Discovery Automation: A tool that analyzes any data set, generates automatic statistics and graphs, and identifies key relationships

Middle Level

Advanced Data Processing

- Processing large data sets with**Vaex**
- Multi-core data processing with**Modin**
- Data feature engineering
- Time series data processing**(sktime**)

Comprehensive ML

- Hyperparameter optimization**(Optuna**)
- Ensemble methods (XGBoost, LightGBM, CatBoost)
- Automatic model selection with**AutoGluon**
- Model explainability (SHAP, LIME)

Deployment and MLOps Basics

- Application development with Streamlit and Gradio
- Data processing pipelines with**Kedro**
- Model management with**MLflow**
- Model servitization with FastAPI

Project

Prediction Modeling Platform: A web application where users can upload data, create automated models, benchmark performance and export the model as an API

Advanced Level

Deep Learning

- PyTorch basics
- Scalable DL with**PyTorch Lightning**
- CNN, RNN, Transformers
- **fastai** applications

NLP and LLM

- Hugging Face ecosystem
- Embed text with**Sentence-Transformers**
- LLM applications with**LangChain** and **LlamaIndex**
- Model fine-tuning with**PEFT/LoRA**
- Efficient LLM deployment with**vLLM** and **Ollama**

Visual Data Processing

- OpenCV and computer vision
- Object detection with**Ultralytics (YOLO)**
- Image data augmentation with**Albumentations**

MLOps Advanced

- ML pipelines with**ZenML**
- Model tracking with**Evidently AI**
- Data and model versioning with**DVC**
- Production workflows with**Metaflow**

Project

Multimodal AI Assistant: An interactive assistant that can process text, images and structured data and work on specialized data sources using RAG (Retrieval Augmented Generation)

Areas of Specialization

Scalable Data Processing

- Apache Spark and PySpark
- GPU-accelerated data science with**RAPIDS**
- Parallel calculation with Dask

GenAI

- Diffusion models and image generation

- Multimodal models (image-text)
- Structured LLM programming with**LMQL**

Reinforced Learning

- RL basics
- Stable Baselines3 and acme libraries

Advanced MLOps

- ML workflows with Kubeflow
- Model presentation optimization
- Model packaging with**BentoML**

Projects

1. **Personalized RAG System**: An information retrieval system that works on specific data sources and adapts to user behavior
2. **GenAI Content Studio**: A multimodal content production platform that generates text, images and audio
3. **Time Series Forecasting System**: Forecasting system that processes real-time data streams, fed from multiple sources
4. **MLOps Orchestration Platform**: Platform that monitors models in production, triggers automatic retraining and provides model management

At the end of each phase we will develop practical projects using modern technologies and solutions to real-world problems.