AI ENGINEERING PROGRAM

TuxAcademy

Duration: 6 Months : Schedule: 3 Days per Week : Class Duration: 2 hrs

MODULE 1 — Python Programming for AI Engineering

- Python overview, installation, execution, notebooks and IDEs.
- Variables, naming rules, data types and type conversion.
- Input/output, arithmetic, comparison and logical operators.
- Conditional statements: if, elif and else.
- for and while loops; break, continue and pass.

Python Data Structures

- Strings and common string operations.
- Lists, tuples, sets and dictionaries; when to use each.
- Indexing, slicing, nested collections and iteration.
- List and dictionary comprehensions.

Functions and Modular Programming

- Defining and calling functions; parameters, arguments and return values.
- Default and keyword arguments; variable scope.
- Lambda functions and reusable functions.
- Importing modules and packages; creating custom modules.

Essential Python for Applications

- Reading and writing text, CSV and JSON files.
- Exception handling using try, except, else and finally.
- Basic OOP: classes, objects, attributes, methods and constructors.
- Working with dates and paths using datetime, os and pathlib.
- Installing packages with pip; virtual environments and environment variables.
- Introduction to Git and GitHub.

Hands-on Implementation

- Student record management program.
- CSV reader and processor.
- JSON-based record management application.
- File-processing utility.

Tools and Technologies

Python 3, Jupyter Notebook / Google Colab, PyCharm or VS Code, pip, venv, Git, GitHub

TuxAcademy

MODULE 2 — Mathematics, Statistics and Data Preparation

Statistics and Probability

- Data types, population and sample.
- Mean, median, mode, range, variance and standard deviation.
- Quartiles, percentiles, outliers and data distributions.
- Normal distribution, correlation and covariance.
- Probability, events, independent/dependent events and conditional probability.
- Conceptual introduction to Bayes' theorem.

Linear Algebra Essentials

- Scalars, vectors and matrices.
- Vector and matrix operations; dimensions and shapes.
- Dot product and why vectors are used in machine learning and embeddings.

Data Handling with NumPy and Pandas

- NumPy arrays, dimensions, indexing, slicing and array operations.
- Pandas Series and DataFrames.
- Reading CSV and Excel files; inspecting and selecting data.
- Filtering, sorting, grouping and aggregation.
- Handling missing values, duplicates and data types.
- Merging and concatenating datasets.

Exploratory Data Analysis

- Descriptive statistics and understanding distributions.
- Univariate and bivariate analysis.
- Histograms, box plots, scatter plots, bar charts and heatmaps.
- Correlation analysis and identifying patterns.

Hands-on Implementation

- Student performance data analysis.
- Sales dataset exploration.
- House price dataset EDA.

Tools and Technologies

NumPy, Pandas, Matplotlib, Seaborn, Jupyter Notebook / Google Colab

MODULE 3 — Machine Learning

Machine Learning Foundations

- AI, Machine Learning and Deep Learning: relationships and differences.
- Traditional programming versus machine learning.
- Supervised, unsupervised and reinforcement learning overview.
- Features, target variables, training data and test data.
- Typical machine learning project workflow.

Data Preprocessing

- Data cleaning and missing-value treatment.
- Label encoding and one-hot encoding.
- Standardization and normalization.
- Train-test split, feature selection and data leakage.
- Scikit-learn transformers and preprocessing pipelines.

TuxAcademy

Regression

- Regression problems and prediction.
- Simple and multiple linear regression.
- Interpreting coefficients and intercepts.
- Predictions, residuals and error analysis.
- Evaluation: MAE, MSE, RMSE and R-squared.

Classification

- Classification problems and decision boundaries.
- Logistic Regression.
- Decision Trees and Random Forest.
- K-Nearest Neighbors (KNN).
- Introduction to Support Vector Machines and Naive Bayes.

Model Evaluation and Improvement

- Confusion matrix, accuracy, precision, recall and F1-score.
- False positives, false negatives, ROC curve and AUC overview.
- Overfitting, underfitting and bias-variance trade-off.
- Cross-validation and regularization.
- Hyperparameter tuning with GridSearchCV and RandomizedSearchCV.
- Feature importance, model comparison and selection.
- Saving and loading models with Joblib.

Unsupervised Learning

- Clustering and customer segmentation.
- K-Means and the elbow method.
- Hierarchical clustering overview.
- Dimensionality reduction and introductory PCA.

Hands-on Implementation

- House price prediction using regression.
- Titanic survival classification.
- Customer segmentation using clustering.
- Loan approval prediction.

Tools and Technologies

Scikit-learn, Pandas, NumPy, Matplotlib, Seaborn, Joblib, Jupyter Notebook / Google Colab

MODULE 4 — Deep Learning and Neural Networks

Neural Network Fundamentals

- Deep Learning and its relationship with Machine Learning.
- Artificial neurons, layers, weights and biases.
- Forward propagation and loss functions.
- Activation functions: sigmoid, ReLU, Leaky ReLU and softmax.
- Gradient descent and backpropagation at a conceptual level.
- Vanishing and exploding gradients.

Building and Training Neural Networks

- TensorFlow and Keras overview.
- Sequential models and Dense layers.
- Compiling a model: optimizer, loss and metrics.

TuxAcademy

- Training with fit; epochs, batch size and validation data.
- Evaluating models and making predictions.
- Training/validation curves, dropout and early stopping.
- Model checkpointing and saving/loading models.

Convolutional Neural Networks

- Image representation: pixels, channels and dimensions.
- Convolution, filters and feature maps.
- Pooling and max pooling.
- Flattening and fully connected layers.
- Building a CNN for image classification.

Recurrent Neural Networks

- Sequential data and the need for sequence models.
- RNN architecture and hidden states.
- Limitations of basic RNNs.
- Introduction to LSTM and GRU.
- Building a simple sentiment classification model.

Hands-on Implementation

- MNIST handwritten digit recognition using CNN.
- IMDB movie review sentiment classification using RNN.
- Compare training and validation performance.

Tools and Technologies

TensorFlow, Keras, NumPy, Matplotlib, Google Colab, MNIST and IMDB datasets

MODULE 5 — Natural Language Processing

NLP Foundations and Text Processing

- What NLP is and where it is used.
- Text normalization and cleaning.
- Sentence and word tokenization.
- Removing punctuation and stopwords.
- Stemming and lemmatization.

Text Representation

- Why text must be represented numerically.
- Bag of Words and CountVectorizer.
- TF-IDF and document-term representations.
- Word embeddings and semantic similarity.
- Word2Vec: CBOW and Skip-Gram concepts.

Common NLP Tasks

- Text classification and sentiment analysis.
- Spam detection and text similarity.
- Introduction to named entity recognition.
- Introduction to language modelling.

Hands-on Implementation

- Email spam classifier.
- Sentiment analysis application.
- Text classification mini-project.

TuxAcademy

Tools and Technologies

NLTK, spaCy, Scikit-learn, Gensim (Word2Vec), Pandas

MODULE 6 — Transformers and Modern AI Architectures

Transformer Architecture

- Why sequence models led to attention-based architectures.
- Encoder and decoder structure.
- Self-attention and Query, Key and Value concepts.
- Scaled dot-product attention and multi-head attention.
- Positional encoding.
- Feed-forward layers, residual connections and normalization.
- Masked self-attention and cross-attention.

Transformer Model Families

- Encoder-only, decoder-only and encoder-decoder architectures.
- BERT, GPT and T5: purpose and architectural differences.
- How decoder-only models generate text.

Using Pre-trained Models

- Hugging Face ecosystem and Model Hub.
- Tokenizers, model loading and inference pipelines.
- Text classification, summarization, translation and text generation.
- Generation settings: temperature, top-p, sampling and max new tokens.
- Beam search and length penalty at an introductory level.

Hands-on Implementation

- Text summarization using T5.
- Text classification using BERT.
- Text generation using a GPT-style model.

Tools and Technologies

Hugging Face Transformers, Hugging Face Hub, PyTorch , Google Colab

MODULE 7 — Generative AI and Large Language Models

Generative AI Foundations

- Generative AI versus traditional predictive AI.
- Text, image, audio and video generation overview.
- Common use cases and limitations.

Large Language Models

- What LLMs are and next-token prediction at a high level.
- Pre-training, fine-tuning and inference.
- Tokens, tokenization and context windows.
- Parameters, model size and model capability.
- Temperature and sampling.
- Hallucinations, reliability and model limitations.
- Open-weight and proprietary models.

LLM Ecosystem

- GPT, Gemini, Claude and Llama model families.

TuxAcademy

- Hosted models versus locally available models.
- Choosing models based on quality, cost, latency and task.

Prompt Engineering

- Prompt structure: role, task, context and constraints.
- Zero-shot and few-shot prompting.
- Role prompting and example-based prompting.
- Structured prompts and JSON output requirements.
- Prompt refinement, testing and common failure patterns.
- Responsible use and verification of model outputs.

Hands-on Implementation

- Summarization and rewriting tasks.
- Structured information extraction.
- Generate and validate JSON responses.
- Compare outputs from different LLMs.

Tools and Technologies

ChatGPT, Google Gemini, Claude, Hugging Face, Ollama

MODULE 8 — LLM APIs and AI Application Development

API and HTTP Fundamentals

- What APIs are and how client-server communication works.
- REST concepts, HTTP methods and status codes.
- Requests, responses, headers and JSON payloads.
- API authentication and secure API-key handling.
- Using Python requests; handling errors, timeouts and rate limits.

Integrating LLM APIs

- Calling OpenAI, Gemini and Anthropic APIs from Python.
- Sending system, user and assistant messages.
- Managing conversation history and multi-turn chat.
- Streaming responses.
- Structured outputs and JSON responses.
- Function/tool calling: connecting model requests to Python functions.
- Retries, token usage and API cost awareness.

Building Interfaces with Streamlit

- Streamlit application structure and execution.
- Text input, buttons, select boxes and layout elements.
- Displaying model responses and chat interfaces.
- Session state and conversation history.
- File upload and basic document input.
- Connecting Streamlit to Python application logic and LLM APIs.

Introduction to LLM Application Frameworks

- Why application frameworks are useful.
- LangChain chat model integration.
- Prompt templates and output parsers.
- Chains and LCEL overview.
- Connecting tools to an LLM and basic tool binding.

TuxAcademy

Introduction to Retrieval-Augmented Generation

- Why external knowledge may be needed by an LLM.
- RAG architecture: documents, chunks, embeddings, retrieval and generation.
- Vector databases and similarity search concepts.
- Building a basic document question-answering flow.

Tools and Technologies

OpenAI API, Google Gemini API, Anthropic API, Python requests, Pydantic , Streamlit, LangChain, ChromaDB , GitHub

Hands-on Implementation

- LLM-powered chatbot with conversation history.
- AI content generator.
- Structured information extraction application.
- Basic PDF/document Q&A application.

MODULE 9 — Testing, Deployment and Capstone

Application Engineering Practices

- Organizing Python application folders and modules.
- Managing dependencies and virtual environments.
- Environment variables and API-key security.
- Logging, debugging, exception handling and input validation.

Testing and Evaluation

- Testing fundamentals and unit testing with Pytest.
- Testing API integrations and application functions.
- Evaluating machine learning model performance.
- Checking LLM response quality, consistency and failure cases.
- Basic latency, token and cost monitoring.

Deployment

- Git and GitHub workflow; repositories and version control.
- Deploying Streamlit applications using Streamlit Community Cloud.
- Introduction to FastAPI and REST endpoints.
- Docker concepts and basic Dockerfile workflow.
- Cloud deployment concepts; overview of AWS, Azure and Google Cloud.
- Postman for API testing.

Introduction to AI Agents

- Difference between an LLM application and an AI agent.
- Agent loop and tool-use concepts.
- Basic agent workflow and examples.
- Introduction to LangGraph and agent orchestration.

Capstone Development

- Project selection, problem definition and requirements.
- Application architecture and implementation plan.
- Build, test, document and deploy the application.
- Project presentation and technical review.

TuxAcademy

Hands-on Implementation

- Deploy an AI chatbot or document assistant.
- Deploy an ML prediction application.
- Complete and present an end-to-end capstone project.



Tools and Technologies

Git, GitHub, Pytest, Postman, Streamlit Community Cloud, FastAPI , Docker ,Render , LangGraph

Capstone Project Options

- **AI Customer Support Application** — An LLM-powered chatbot with conversation history, a Streamlit interface and secure API integration.
- **AI Resume Analyzer** — A document-processing application that extracts resume information and produces structured summaries or skill comparisons.
- **AI Document Assistant** — A basic RAG application that processes uploaded documents, retrieves relevant passages and answers questions using an LLM.
- **Image Classification Application** — A CNN-based image classifier with an upload interface and deployed inference workflow.
- **End-to-End Predictive AI Application** — A dataset-to-deployment workflow including EDA, preprocessing, model comparison, evaluation and a prediction interface.

TuxAcademy