# Machine Learning Roadmap

**by**

**Md Shariar Emon Shaikat**

# Class 01: Intro (Class - 1)

- Data Science overview
- AI vs ML vs Deep Learning
- Supervised vs Unsupervised Learning
- Regression, Classification, Clustering
- Real-world use cases
- Career paths for ML/AI domain (ML+Analytics vs. Academic vs. ML+SWE)
- Required tools
- Required platforms (GitHub, Kaggle, Colab)
- Additional materials on Linear Algebra and Statistics
- Q&A Session

# Module 1: Python Fundamentals (Classes 2 - 4)

## Class 2: Python Setup & Basics

**Topics:**

- Setting up IDE (VS Code, PyCharm, Jupyter)
- Python syntax and basic operations
- Variables and data types
- Input/output operations
- Basic arithmetic and string operations

## Class 3: Control Flow & Functions

**Topics:**

- Conditional statements (if, elif, else)
- Loops (for, while)
- Break, continue, and pass statements
- Functions (def, return, parameters)
- Default arguments and *args, **kwargs

- List (creation, indexing, slicing, methods)
- Tuple (immutability, use cases)
- Set (unique elements, operations)
- Dictionary (key-value pairs, methods)
- When and why to use each structure
- List comprehensions

**Assignment 1: Python Problem Solving**

- Write a prime number checker
- Create a Fibonacci series generator
- Build a factorial calculator using recursion
- Design a pattern printer (pyramid, diamond)
- Make a grade calculator based on marks
- Create a simple calculator with all operations
- List (creation, indexing, slicing, methods)
- Tuple (immutability, use cases)
- Set (unique elements, operations)
- Dictionary (key-value pairs, methods)
- When and why to use each structure
- List comprehensions

## Class 4: Advanced Python Basics & File Handling

**Topics:**

- Lambda functions
- Map, Filter, Reduce functions
- List, Dictionary, Set comprehensions
- File handling (open, read, write, append)
- Working with CSV files
- JSON file operations
- Error handling with try-except
- String formatting (f-strings, format())
- Working with modules (import, from)
- Command line arguments

- Regular expressions basics (re module)

**Assignment 2: Advanced Python Practice**

- Create a file-reading/writing program for student records
- Use list comprehension to filter and transform data
- Read a CSV file and perform operations (count, sum, average)
- Write data to JSON format
- Use lambda with map/filter on a list of numbers
- Implement error handling for file not found scenarios
- Create a text file analyzer (count words, lines, characters)
- Use regex to extract emails/phone numbers from text

# Module 2: Data Manipulation & Analysis (Classes 5 - 8)

## Class 5: NumPy Fundamentals

**Topics:**

- NumPy arrays vs Python lists
- Array creation and operations
- Indexing and slicing
- Mathematical operations
- Broadcasting
- Random number generation
- Array reshaping

## Class 6: Pandas for Data Manipulation

**Topics:**

- Series and DataFrame
- Reading/writing data (CSV, Excel, JSON)
- Data selection and filtering

- Handling missing values
- Data aggregation and groupby
- Merging and joining datasets
- Basic time series operations

**Assignment 3: Data Manipulation with Pandas**

- Load a real dataset (Titanic/any Kaggle dataset)
- Perform data filtering based on multiple conditions
- Handle missing values using 3 different techniques
- Create aggregations using groupby
- Merge/join two related datasets
- Export cleaned data to CSV

## Class 7: Data Cleaning & Preprocessing

**Topics:**

- Understanding different data types
- Identifying and handling missing values (mean, median, mode, forward/backward fill)
- Outlier detection (IQR, Z-score)
- Outlier treatment (capping, removal)
- Data type conversion
- Duplicate removal
- String operations for text cleaning

## Class 8: Exploratory Data Analysis (EDA) & Visualization

**Topics:**

- EDA methodology
- Statistical summaries (describe, info)
- Distribution analysis
- Correlation analysis
- Matplotlib basics (line, bar, scatter plots)
- Seaborn for statistical plots

- Plotly for interactive visualizations

**Assignment 4: Comprehensive EDA & Visualization**

- Choose a dataset (e.g., house prices, sales, customer data)
- Perform complete data cleaning
- Create 10+ different visualizations
- Analyze correlations and patterns
- Identify key insights from the data
- Write a detailed EDA report with findings

# Module 3: Feature Engineering & ML Preparation (Classes 9 - 10)

## Class 9: Feature Engineering & Encoding

**Topics:**

- Feature scaling (StandardScaler, MinMaxScaler, RobustScaler)
- Normalization techniques
- Encoding categorical variables (Label, One-Hot, Ordinal)
- Feature creation and extraction
- Binning and discretization
- Handling date-time features

## Class 10: Train-Test Split & Data Leakage Prevention

**Topics:**

- Importance of data splitting
- Train-test-validation split
- Stratified sampling
- Cross-validation (K-Fold, Stratified K-Fold)
- Data leakage concepts and prevention
- Handling imbalanced datasets (SMOTE, SMOTETomek, class weights)

- Pipeline creation for preprocessing

**Assignment 5: Feature Engineering & Data Preparation**

- Take a raw dataset and apply all preprocessing steps
- Implement different scaling techniques and compare results
- Encode categorical variables properly
- Create new features from existing ones
- Split data with stratification
- Handle imbalanced data using SMOTE
- Build a complete preprocessing pipeline

# Module 4: Introduction to Machine Learning (Classes 11 -12)

## Class 11: ML Fundamentals & First Model

**Topics:**

- AI vs ML vs Deep Learning
- Supervised vs Unsupervised Learning
- Regression vs Classification
- ML workflow overview
- Introduction to scikit-learn
- Building first Linear Regression model
- Model evaluation basics

## Class 12: Model Evaluation & Metrics

**Topics:**

- Regression metrics (MAE, MSE, RMSE, $R^2$)
- Classification metrics (Accuracy, Precision, Recall, F1)
- Confusion matrix
- ROC curve and AUC

- Overfitting vs Underfitting
- Bias-Variance tradeoff
- Learning curves

**Assignment 6: First ML Model & Evaluation**

- Build a Linear Regression model from scratch
- Train on a dataset (e.g., house prices, salary prediction)
- Evaluate using multiple metrics
- Create visualizations of predictions vs actual
- Implement a classification model (Logistic Regression)
- Calculate and interpret the confusion matrix, ROC-AUC
- Identify overfitting/underfitting

# Module 5: Regression Algorithms (Classes 13 - 15)

## Class 13: Linear & Polynomial Regression

**Topics:**

- Simple Linear Regression theory
- Multiple Linear Regression
- Loss functions (MSE, MAE)
- Polynomial Regression
- When to use polynomial features
- Feature importance in regression

**Project 1: House Price Prediction System**

- Dataset: House prices dataset
- Implement both Linear and Polynomial Regression
- Compare model performances
- Feature importance analysis

## Class 14: Regularization Techniques

**Topics:**

- Ridge Regression (L2 regularization)
- Lasso Regression (L1 regularization)
- ElasticNet (L1 + L2)
- When to use each technique
- Feature selection with Lasso
- Hyperparameter tuning for alpha

### Class 15: KNN & Distance-Based Learning

**Topics:**

- K-Nearest Neighbors algorithm
- Distance metrics (Euclidean, Manhattan)
- Choosing optimal K
- KNN for regression and classification
- Advantages and limitations
- Computational complexity

**Assignment 7: Regression Models Comparison**

- Apply Linear, Polynomial, Ridge, Lasso, and KNN regression
- Use a dataset (e.g., car prices, medical costs)
- Compare all models using $R^2$, RMSE, MAE
- Plot predictions for each model
- Find optimal hyperparameters
- Create a comparison report with recommendations

## Module 6: Classification Algorithms (Classes 16 - 19)

### Class 16: Logistic Regression

**Topics:**

- Binary classification

- Sigmoid function
- Decision boundary
- Multiclass classification (One-vs-Rest, Softmax)
- Probability predictions
- Classification threshold tuning

**Project 2: Disease Prediction System**

- Dataset: Diabetes/Heart Disease dataset
- Build a binary classification model
- Tune classification threshold
- Create a comprehensive evaluation report
- Implement a web interface
- Add probability predictions

## Class 17: Support Vector Machines (SVM)

**Topics:**

- Margin and hyperplane concepts
- Support vectors
- Kernel trick (Linear, RBF, Polynomial)
- C parameter (regularization)
- Gamma parameter
- SVM for binary and multiclass

## Class 18: Naive Bayes & Text Classification

**Topics:**

- Probability fundamentals
- Bayes theorem
- Naive Bayes assumptions
- Types (Gaussian, Multinomial, Bernoulli)
- Text preprocessing (lowercase, remove punctuation, stopwords)
- Text vectorization (CountVectorizer, TfidfVectorizer)

**Project 3: Email Spam Classifier**

- Dataset: Spam email dataset
- Text preprocessing pipeline
- TF-IDF vectorization
- Train Naive Bayes classifier
- Evaluate with accuracy, precision, and recall
- Build an interactive web app
- Test with custom emails

### Class 19: Decision Trees

**Topics:**

- Decision tree structure
- Information gain and Gini impurity
- Tree building process
- Pruning techniques
- Feature importance
- Visualization of decision trees
- Hyperparameters (max_depth, min_samples_split)

**Assignment 8: Classification Models Comparison**

- Dataset: Multi-class classification (Iris, Wine Quality)
- Implement Logistic Regression, SVM (all kernels), Naive Bayes, and Decision Tree
- Create a confusion matrix for each
- Compare accuracy, precision, recall, and F1-score
- Visualize decision boundaries (for 2D projections)
- Document which model performs best and why

## Module 7: Ensemble Learning (Classes 20 - 21)

### Class 20: Ensemble Methods - Bagging & Boosting

**Topics:**

- Ensemble learning concept
- Bagging (Bootstrap Aggregating)
- Random Forest algorithm
- Boosting concept
- Gradient Boosting basics
- XGBoost, CatBoost, LightGBM
- Feature importance in ensemble methods
- When to use which ensemble method

**Project 4: Customer Churn Prediction**

- Dataset: Telecom/Bank customer churn
- Build a Random Forest classifier
- Implement XGBoost and CatBoost
- Feature importance analysis
- Compare Decision Tree vs Random Forest vs XGBoost
- Hyperparameter tuning
- Create a business insights dashboard

## Class 21: Hyperparameter Tuning & Model Selection

**Topics:**

- Grid Search
- Random Search
- Bayesian Optimization concept
- Automated tuning with Optuna
- Cross-validation with tuning
- Model selection strategies
- Best practices for hyperparameter tuning

**Assignment 9: Ensemble Methods & Hyperparameter Tuning**

- Dataset: Complex classification problem (Credit Card Fraud, Wine Quality)
- Implement Random Forest, XGBoost, and CatBoost
- Apply Grid Search and Random Search

- Use Optuna for advanced tuning
- Compare all models with optimal parameters
- Create a detailed performance comparison report
- Visualize feature importance

# Module 8: Unsupervised Learning & Time Series (Classes 22-23)

## Class 22: Clustering Algorithms

**Topics:**

- Unsupervised learning overview
- K-Means clustering algorithm
- Elbow method for optimal K
- Silhouette score
- Hierarchical clustering
- DBSCAN introduction
- Cluster evaluation

**Project 5: Customer Segmentation System**

- Dataset: Customer data (demographics, purchase behavior)
- Apply K-Means clustering
- Find the optimal number of clusters
- Analyze cluster characteristics
- Create customer personas
- Build an interactive Streamlit dashboard
- Generate business recommendations

## Class 23: Time Series Forecasting

**Topics:**

- Time series basics

- Components (trend, seasonality, residual)
- Stationary vs non-stationary data
- ARIMA model
- SARIMA for seasonal data
- Prophet by Facebook
- Evaluation metrics for time series

**Project 6: Stock Market Price Forecasting**

- Dataset: Historical stock prices
- Implement ARIMA and SARIMA
- Use Facebook Prophet
- Compare forecast accuracy
- Visualize predictions with confidence intervals
- Deploy forecasting app
- Add a multiple stock selection feature

# Module 9: Advanced NLP & Projects (Classes 24 - 25)

## Class 24: Advanced NLP & Sentiment Analysis

**Topics:**

- Advanced text preprocessing
- N-grams
- Word embeddings concept
- Sentiment analysis techniques
- Building an end-to-end NLP pipeline
- Model deployment for text classification

**Project 7: Fake News Detection System**

- Dataset: Fake news dataset
- Complete text preprocessing pipeline
- Feature extraction with TF-IDF
- Train multiple classifiers (Naive Bayes, SVM, Random Forest)

- Model comparison and selection
- Deploy with the Gradio interface
- Add confidence scores

### Class 25: End-to-End ML Project Workflow

**Topics:**

- Complete ML project lifecycle
- Problem definition and scoping
- Data collection strategies
- Comprehensive EDA
- Model selection criteria
- Evaluation and comparison
- Documentation best practices
- Presentation skills

**Project 8: Amazon Product Review Sentiment Analysis**

- Dataset: Amazon reviews
- End-to-end NLP pipeline
- Sentiment classification (Positive/Negative/Neutral)
- Feature engineering for text
- Multiple model comparison
- Deploy web application
- Add review rating prediction

## Module 10: Model Deployment (Classes 26-27)

### Class 26: Web Apps with Streamlit & Gradio

**Topics:**

- Introduction to Streamlit
- Building interactive dashboards
- Model integration with Streamlit

- Gradio for quick ML interfaces
- User input handling
- File upload functionality
- Deployment on Streamlit Cloud/Hugging Face Spaces
- Best practices for UI/UX

**Project 9: Multi-Model ML Dashboard**

- Create a Streamlit app with multiple ML models
- Include regression, classification, and clustering
- Add data upload functionality
- Interactive visualizations
- Model comparison feature
- Deploy on Streamlit Cloud
- Share publicly accessible link

## Class 27: FastAPI for ML Deployment

**Topics:**

- What is FastAPI and why for ML
- REST API basics (GET, POST)
- Creating ML inference API
- Request/response schema with Pydantic
- Model serialization (pickle, joblib)
- Loading and using saved models
- Testing with Swagger UI
- Error handling and validation
- CORS configuration

**Assignment 10: ML Model API Development**

- Choose a trained model from previous projects
- Create a FastAPI application
- Implement a POST endpoint for predictions
- Add proper request/response validation

- Include error handling
- Test all endpoints using Swagger UI
- Document API with examples
- (Optional) Deploy on Render/Railway

# Module 11: Deep Learning with PyTorch (Classes 28 - 33)

## Class 28: Neural Networks Fundamentals

**Topics:**

- What is Deep Learning?
- Neuron and Perceptron model
- Artificial Neural Networks (ANN) architecture
- Forward propagation
- Activation functions (Sigmoid, Tanh, ReLU, Leaky ReLU, Softmax)
- Loss functions (MSE, Binary Cross-Entropy, Categorical Cross-Entropy)
- Backpropagation concept (intuition)
- Gradient Descent and learning rate
- Optimizers (SGD, Momentum, Adam, RMSprop)
- Batch, Mini-batch, and Stochastic gradient descent
- Overfitting in neural networks
- Regularization techniques (Dropout, L1/L2)

**Hands-on:**

- Build perceptron from scratch using NumPy
- Visualize activation functions
- Understand gradient descent with simple examples

## Class 29: PyTorch, Keras Basics & Building Neural Networks

**Topics:**

- Introduction to PyTorch and Keras
- Why PyTorch for Deep Learning?

- Tensors and tensor operations
- PyTorch vs NumPy
- Autograd and automatic differentiation
- Building neural networks with nn.Module
- Defining layers (Linear, Activation)
- Dataset and DataLoader
- Training loop structure
- Evaluation loop
- Saving and loading models
- GPU acceleration basics

**Project 10: Image Classification with ANN**

- Dataset: MNIST or Fashion-MNIST
- Build ANN using PyTorch
- Implement training and validation loops
- Use different optimizers (SGD, Adam)
- Apply dropout for regularization
- Plot training/validation loss curves
- Evaluate on test set
- Visualize predictions
- Compare with traditional ML models

## Class 30: Convolutional Neural Networks (CNN)

**Topics:**

- Limitations of ANN for images
- Introduction to CNN
- Convolution operation
- Filters and feature maps
- Pooling layers (Max pooling, Average pooling)
- CNN architecture (Conv → Pool → FC)
- Famous architectures overview (LeNet, AlexNet, VGG)
- Transfer Learning concept
- Data augmentation techniques

- Building a CNN with PyTorch

**Project 11: Advanced Image Classification with CNN**

- Dataset: CIFAR-10 or Custom image dataset
- Build a CNN from scratch using PyTorch
- Implement data augmentation
- Apply transfer learning (ResNet/VGG pretrained)
- Compare CNN vs ANN performance
- Fine-tune pretrained model
- Deploy with Gradio
- Test with real images

**Assignment 11: CNN Implementation & Comparison**

- Build a CNN for image classification
- Implement at least 2 different CNN architectures
- Use data augmentation techniques
- Apply transfer learning with a pretrained model
- Compare performance: ANN vs CNN vs Transfer Learning
- Create a visualization of learned filters
- Document architecture decisions

## Class 31: Recurrent Neural Networks (RNN) & Sequence Models

**Topics:**

- Sequential data and its challenges
- Introduction to RNN
- RNN architecture and working
- Vanishing gradient problem
- Long Short-Term Memory (LSTM)
- Gated Recurrent Unit (GRU)
- Bidirectional RNN
- Sequence-to-sequence models
- Applications: Text generation, time series, translation

- Building RNN/LSTM with PyTorch

**Project 12: Text Generation / Sentiment Analysis with LSTM**

- **Option A - Text Generation:**
    - Dataset: Text corpus (Shakespeare, poems, code)
    - Build a character-level LSTM
    - Train the text generation model
    - Generate new text samples
- **Option B - Multilingual Sentiment Analysis:**
    - Dataset: IMDb reviews or Twitter sentiment
    - Build an LSTM for sentiment classification
    - Compare with traditional ML (Naive Bayes)

**Assignment 12: Deep Learning Projects & Documentation**

- Complete one of the LSTM projects
- Build an end-to-end deep learning pipeline
- Experiment with LSTM vs GRU
- Tune hyperparameters (layers, units, dropout)
- Create comprehensive project documentation
- Write a technical report comparing all DL models learned

## Class 32: Transformers & Attention Mechanisms

**Topics:**

- **Limitations of RNNs and motivation for Transformers**
- **Self-attention mechanism**
    - Query, Key, Value concepts
    - Scaled dot-product attention
    - Multi-head attention
- **Transformer architecture**
    - Encoder-decoder structure

    - Positional encoding
    - Feed-forward networks
    - Layer normalization and residual connections
- **Pre-trained Transformer models**
    - BERT (Bidirectional Encoder Representations from Transformers)
    - GPT (Generative Pre-trained Transformer)
    - T5 (Text-to-Text Transfer Transformer)
- **Transfer learning and fine-tuning**
- **Hugging Face Transformers library**
- **Applications**: Text classification, NER, question answering, summarization
- **Vision Transformers (ViT)** - brief introduction

**Project 13: Transformer-based NLP Application**

## Option A - Text Classification with BERT:

- Dataset: AG News, 20 Newsgroups, or custom dataset
- Load pre-trained BERT model
- Fine-tune for the classification task
- Evaluate performance vs LSTM
- Analyze attention weights

**Option B - Question Answering System:**

- Dataset: SQuAD (Stanford Question Answering Dataset)
- Use pre-trained transformer (BERT/RoBERTa)
- Build an extractive QA system
- Create an interactive demo
- Compare different model sizes

**Assignment 13: Transformer Experimentation**

- Fine-tune a pre-trained transformer model
- Compare BERT, RoBERTa, and DistilBERT performance
- Experiment with different learning rates and epochs
- Visualize attention patterns

- Document computational requirements (time, memory)
- Write comparative analysis: RNN vs Transformer

## Class 33: Reinforcement Learning Fundamentals

**Topics:**

- **Introduction to Reinforcement Learning**
    - Agent, environment, state, action, reward
    - Markov Decision Process (MDP)
    - Policy, value function, Q-function
- **Exploration vs Exploitation tradeoff**
    - Epsilon-greedy strategy
    - Upper Confidence Bound (UCB)
- **Value-based methods**
    - Q-Learning algorithm
    - State-action value function
    - Bellman equation
- **Deep Q-Networks (DQN)**
    - Combining Q-learning with neural networks
    - Experience replay
    - Target networks
- **Policy gradient methods**
    - REINFORCE algorithm
    - Actor-Critic methods
    - Proximal Policy Optimization (PPO)
- **RL Frameworks**: OpenAI Gym, Stable-Baselines3
- **Applications**: Game playing, robotics, resource optimization

**Project 14: Reinforcement Learning Agent**

## Option A - Game Playing Agent:

- Environment: CartPole, MountainCar, or Atari game

- Implement Q-Learning or DQN
- Train the agent to play the game
- Visualize learning progress (rewards over episodes)
- Compare different hyperparameters (learning rate, discount factor)

### Option B - Custom Environment RL:

- Design a custom grid-world or navigation problem
- Implement an environment following the OpenAI Gym interface
- Train an agent using Q-learning or policy gradient
- Visualize the agent's learned policy
- Analyze convergence and stability

#### Assignment 14: RL Algorithm Comparison

- Implement basic Q-learning from scratch
- Use Stable-Baselines3 for DQN and PPO
- Compare performance across algorithms
- Tune hyperparameters (discount factor γ, learning rate α, exploration ε)
- Create visualizations of:
    - Reward curves over training
    - Policy heatmaps (for grid worlds)
    - Action distributions
- Write a comprehensive report on RL concepts and results

# Module 12: Thesis, Publication, Documentation & Career Guideline (Class 34)

## Class 34: LaTeX, Thesis Writing & Career Guidelines

#### Part 1: LaTeX & Thesis Writing (Overleaf)

**Topics:**

- Introduction to LaTeX

- Why LaTeX for academic writing
- Setting up an Overleaf account
- Document structure (sections, subsections)
- Mathematical equations formatting
- Tables and figures
- Citations and bibliography (BibTeX)
- Cross-referencing
- Research paper template
- Thesis formatting guidelines
- Conference/Journal Publication Guideline

**Hands-on:**

- Create a research paper template
- Write mathematical equations
- Add tables and figures
- Manage references
- Export to PDF

**Part 2: Career Guidelines**

**Topics:**

- ML/AI career paths:
    - **ML + Analytics**: Data Scientist, ML Analyst
    - **Academic Research**: Research Scientist, PhD path
    - **ML + Software Engineering**: ML Engineer, AI Engineer
- Building professional resume for ML roles
- GitHub portfolio optimization
- LinkedIn profile setup and optimization
- Kaggle competitions strategy
- Interview preparation (technical + behavioral)
- Networking in ML community
- Continuous learning resources
- Freelancing opportunities
- Building personal brand

**Career Action Items:**

- Resume building workshop
- GitHub portfolio review
- Mock interview practice
- LinkedIn profile optimization
- Kaggle competition participation

**Assignment 15: Documentation & Career Preparation**

- Write a complete project documentation in LaTeX
- Include: Abstract, Introduction, Methodology, Results, Conclusion
- Add mathematical formulas for models used
- Create professional tables and figures
- Manage citations properly
- Update resume with all 10 projects
- Organize GitHub repository professionally
- Write README files for all projects
- Optimize LinkedIn profile
- Participate in 1 Kaggle competition

# FINAL CAPSTONE PROJECT

## Final Project: End-to-End ML Solution (Any Domain)

**Requirements:** Choose one of the following domains or propose your own:

1. **Healthcare**: Disease Prediction System with multiple models
2. **Finance**: Credit Risk Assessment & Fraud Detection
3. **E-commerce**: Product Recommendation System
4. **Social Media**: Content Moderation System
5. **Agriculture**: Crop Yield Prediction
6. **Education**: Student Performance Prediction
7. **Real Estate**: Property Price Prediction & Analysis
8. **Climate**: Weather Forecasting System

9. **Transportation**: Traffic Prediction System
10. **Custom Domain**: Your own innovative idea

**Project Components:**

**1. Data Phase:**

- Collect/use real-world dataset
- Comprehensive data cleaning
- Advanced EDA with insights
- Feature engineering

**2. Modeling Phase:**

- Implement minimum 5 different models
- Compare performance metrics
- Hyperparameter tuning
- Select best model with justification

**3. Deployment Phase:**

- Build web application (Streamlit/Gradio)
- Create REST API (FastAPI)
- Deploy on cloud platform
- Add user authentication (optional)

**4. Documentation Phase:**

- Complete project report in LaTeX
- Technical documentation
- GitHub README with:
    - Problem statement
    - Dataset description
    - Methodology
    - Results and comparison
    - How to run
    - Demo screenshots/video

- Presentation slides

**5. Presentation:**

- 15-20 minute presentation
- Demo of working application
- Q&A session

**Evaluation Criteria:**

- Code quality and organization (20%)
- Model performance (25%)
- Deployment and UI/UX (20%)
- Documentation (20%)
- Innovation and insights (15%)

# Complete Assignment Summary

| Assignmet | Topic | Type |
|---|---|---|
| 1 | Python Problem Solving | Coding |
| 2 | OOP Implementation | Coding |
| 3 | Data Manipulation with Pandas | Data Analysis |
| 4 | EDA & Visualization | Data Analysis |
| 5 | Feature Engineering & Pipeline | Preprocessing |
| 6 | First ML Model & Evaluation | ML Basic |
| 7 | Regression Models Comparison | ML Models |

| | | |
|---|---|---|
| 8 | Classification Models Comparison | ML Models |
| 9 | Ensemble & Hyperparameter Tuning | Advanced ML |
| 10 | ML Model API Development | Deployment |
| 11 | CNN Implementation & Comparison | Deep Learning |
| 12 | Deep Learning Projects | Deep Learning |
| 13 | Documentation & Career Prep | Professional |

## Complete Project Summary (12 + 1 Final)

| Project Name | Domain | Algorithms |
|---|---|---|
| House Price Prediction | Real Estate | Linear, Polynomial Regression |
| Disease Prediction System | Healthcare | Logistic Regression |
| Email Spam Classifier | NLP | Naive Bayes |
| Customer Churn Prediction | Business | Random Forest, XGBoost |
| Customer Segmentation | Business | K-Means Clustering |
| Stock Market Forecasting | Finance | ARIMA, SARIMA, Prophet |
| Fake News Detection | NLP | Multiple Classifiers |

| | | |
|---|---|---|
| Amazon Review Analysis | NLP | Sentiment Analysis |
| Multi-Model Dashboard | General | Multiple Models |
| Image Classification (ANN) | Computer Vision | Neural Networks |
| Advanced Image Classification (CNN) | Computer Vision | CNN, Transfer Learning |
| Text Generation/Sentiment (LSTM) | NLP/DL | RNN, LSTM, GRU |
| **End-to-End ML Solution - Final Project** | **Students Choice** | **Multiple Models** |

# Tools & Platforms

### Must-Have:

- **Python IDE**: VS Code / PyCharm / Jupyter
- **Cloud Notebooks**: Google Colab (free GPU)
- **Version Control**: Git & GitHub
- **Dataset Source**: Kaggle
- **Deployment**: Streamlit Cloud, Hugging Face Spaces
- **Documentation**: Overleaf (LaTeX)

### Libraries to Master:

- **Core**: NumPy, Pandas, Matplotlib, Seaborn
- **ML**: Scikit-learn, XGBoost, CatBoost
- **DL**: PyTorch (basics)
- **NLP**: NLTK, TextBlob
- **Deployment**: Streamlit, Gradio, FastAPI
- **Time Series**: Statsmodels, Prophe

# Additional Learning Resources

## Documentation:

- Scikit-learn documentation
- PyTorch tutorials
- Kaggle Learn courses
- Anthropic Claude prompting guide

## Practice Platforms:

- **LeetCode**: Python practice
- **Kaggle**: Competitions & datasets
- **HackerRank**: Problem-solving
- **GitHub**: Open source contributions

## Stay Updated:

- Papers with Code
- ArXiv ML papers
- Towards Data Science (Medium)
- ML subreddit