16-Week AI/ML Mastery Roadmap: From Zero to Job-Ready

📋 Executive Summary: 4-Week Block Overview

Weeks 1-4: Foundations

Goal: Master Python programming, essential mathematics, and data manipulation basics
Key Skills: Python fluency, NumPy/Pandas, basic statistics, Git/GitHub workflow
Capstone: End-to-end data analysis pipeline with EDA and visualization
Job Relevance: These are the core skills every data scientist uses daily for data exploration and preprocessing

Weeks 5-8: Classical Machine Learning

Goal: Build and evaluate ML models, understand algorithms deeply
Key Skills: Scikit-learn mastery, feature engineering, model evaluation, hyperparameter tuning
Capstone: Multi-model comparison system with automated evaluation pipeline
Job Relevance: By week 8, you can train and tune classical models—core tasks for junior ML engineers

Weeks 9-12: Deep Learning & Neural Networks

Goal: Design and train neural networks for vision, text, and structured data
Key Skills: PyTorch/TensorFlow, CNNs, RNNs, Transfer Learning, model optimization
Capstone: End-to-end deep learning application with deployment
Job Relevance: You'll be able to build production-ready neural networks for real-world applications

Weeks 13-16: MLOps & Production Systems

Goal: Deploy, monitor, and maintain ML systems at scale
Key Skills: Docker, CI/CD, model serving, monitoring, A/B testing, cloud deployment
Capstone: Full MLOps pipeline with automated training, versioning, and deployment
Job Relevance: These skills differentiate ML Engineers from data scientists—crucial for industry roles

---

📚 Detailed Weekly Breakdown

Week 1: Python Fundamentals & Development Environment

Topics to Cover:

- Python syntax, data types, control flow
- Functions, modules, and packages
- Virtual environments and package management
- Git basics and GitHub setup

- Jupyter notebooks and Google Colab

Estimated Time: 13 hours

Tasks:

- Setup Python environment (1.5 hrs)
- Complete Python basics tutorial (4 hrs)
- Practice Python exercises (3 hrs)
- Learn Git/GitHub basics (2 hrs)
- Setup Jupyter and run first notebook (1.5 hrs)
- Complete mini-project (1 hr)

Resources (Free Only):

- Python Official Tutorial
- Kaggle Python Course
- Git Tutorial by Atlassian
- Google Colab Getting Started

My Resources:

- (Leave blank if no specific resources from your list)

Mini Project:

- **Dataset:** Iris Dataset from UCI
- **Task:** Build a Python data analyzer that reads CSV, calculates statistics, and generates a report
- **Deliverables:** GitHub repo, Jupyter notebook, README with setup instructions
- **Acceptance Criteria:** Code runs without errors, handles edge cases, includes 5+ statistical calculations

Ethics & Best Practices:

- [ ] Add comments to all functions
- [ ] Include data source attribution
- [ ] Check for PII in datasets
- [ ] Use meaningful variable names
- [ ] Include error handling

Weekly Outcomes:

- Can write Python functions and classes
- Can use Git for version control

- Can create and share Jupyter notebooks
- Understands virtual environments

Stretch Goals:

- Implement unit tests using pytest
- Add type hints to functions

Week 2: Data Manipulation with NumPy & Pandas

Prerequisites: Week 1 Python fundamentals

Topics to Cover:

- NumPy arrays and operations
- Pandas DataFrames and Series
- Data loading, cleaning, and transformation
- Handling missing data
- Basic data aggregation

Estimated Time: 13 hours

Tasks:

- NumPy fundamentals tutorial (3 hrs)
- Pandas basics and data loading (3 hrs)
- Data cleaning exercises (2.5 hrs)
- Practice aggregation and groupby (2 hrs)
- Complete mini-project (2.5 hrs)

Resources (Free Only):

- NumPy Official Documentation
- Pandas Getting Started Tutorials
- Kaggle Pandas Course
- Real Python NumPy Tutorial

My Resources:

- (Leave blank if no specific resources from your list)

Mini Project:

- **Dataset:** COVID-19 Dataset from Kaggle
- **Task:** Clean and analyze COVID data, create summary statistics by country
- **Deliverables:** GitHub repo, analysis notebook, cleaned dataset, visualization dashboard
- **Acceptance Criteria:** Handle all missing values, create 10+ insights, include time series analysis

Ethics & Best Practices:

- [ ] Document data cleaning decisions
- [ ] Preserve raw data separately
- [ ] Check for data biases
- [ ] Validate transformations
- [ ] Include data dictionary

Weekly Outcomes:

- Can manipulate arrays and dataframes efficiently
- Can clean and prepare real-world datasets
- Understands vectorized operations
- Can perform exploratory data analysis

Stretch Goals:

- Optimize code for large datasets
- Create custom aggregation functions

Week 3: Data Visualization & Statistical Foundations

Prerequisites: Week 2 NumPy/Pandas skills

Topics to Cover:

- Matplotlib and Seaborn for visualization
- Statistical distributions and hypothesis testing
- Correlation and causation
- Descriptive vs inferential statistics
- Interactive visualizations with Plotly

Estimated Time: 13 hours

Tasks:

- Matplotlib/Seaborn tutorial (3 hrs)

- Statistics fundamentals (3 hrs)
- Create visualization portfolio (3 hrs)
- Statistical testing exercises (2 hrs)
- Complete mini-project (2 hrs)

Resources (Free Only):

- Matplotlib Official Tutorials
- Seaborn Tutorial
- Khan Academy Statistics
- Think Stats Free Book

My Resources:

- (Leave blank if no specific resources from your list)

Mini Project:

- **Dataset:** World Happiness Report from Kaggle
- **Task:** Create interactive dashboard with statistical analysis
- **Deliverables:** GitHub repo, Jupyter notebook with visualizations, statistical report
- **Acceptance Criteria:** 15+ visualizations, 5+ statistical tests, interactive elements

Ethics & Best Practices:

- [ ] Avoid misleading visualizations
- [ ] Include confidence intervals
- [ ] Label axes clearly
- [ ] Use colorblind-friendly palettes
- [ ] Document statistical assumptions

Weekly Outcomes:

- Can create publication-quality visualizations
- Understands basic statistical concepts
- Can perform hypothesis testing
- Can identify patterns in data

Stretch Goals:

- Create animated visualizations
- Build Streamlit dashboard

Week 4: Mathematics for ML & First Capstone

Prerequisites: Weeks 1-3 foundations

Topics to Cover:

- Linear algebra essentials
- Calculus for ML (derivatives, chain rule)
- Probability theory
- Optimization basics
- Mathematical intuition for ML

Estimated Time: 14 hours

Tasks:

- Linear algebra fundamentals (3 hrs)
- Calculus essentials (2.5 hrs)
- Probability exercises (2.5 hrs)
- Review and integration (2 hrs)
- Complete capstone project (4 hrs)

Resources (Free Only):

- 3Blue1Brown Linear Algebra Series
- Khan Academy Calculus
- Mathematics for Machine Learning Book
- MIT OCW Linear Algebra

My Resources:

- (Leave blank if no specific resources from your list)

Capstone Project:

- **Dataset:** Titanic Dataset from Kaggle
- **Task:** Complete end-to-end EDA pipeline with statistical analysis and feature engineering
- **Deliverables:** GitHub repo, comprehensive notebook, presentation slides, Docker setup
- **Acceptance Criteria:** Complete EDA, feature engineering, statistical tests, professional documentation

Ethics & Best Practices:

- [ ] Consider survival bias in analysis
- [ ] Document all assumptions
- [ ] Include reproducibility instructions
- [ ] Address class imbalance
- [ ] Discuss ethical implications

Weekly Outcomes:

- Understands mathematical foundations of ML
- Can implement basic algorithms from scratch
- Ready for machine learning concepts
- Portfolio has first complete project

Self-Assessment Checklist:

- [ ] Can manipulate matrices and vectors
- [ ] Understands derivatives and gradients
- [ ] Can calculate probabilities
- [ ] Comfortable with Python and data tools
- [ ] Has 4+ mini-projects on GitHub

Interview Prep Questions:

1. Explain the difference between NumPy arrays and Python lists
2. How do you handle missing data?
3. What is the central limit theorem?
4. Write a function to normalize a dataset
5. Explain overfitting in simple terms

Week 5: Introduction to Machine Learning

Prerequisites: Week 4 mathematics and Python skills

Topics to Cover:

- ML fundamentals and terminology
- Supervised vs unsupervised learning
- Training, validation, and test sets
- Overfitting and underfitting
- Bias-variance tradeoff

Estimated Time: 13 hours

Tasks:

- ML theory and concepts (3 hrs)
- Scikit-learn basics (3 hrs)
- Train first models (3 hrs)
- Model evaluation exercises (2 hrs)
- Complete mini-project (2 hrs)

Resources (Free Only):

- Andrew Ng's Machine Learning Course (Week 1-2)
- Scikit-learn Official Tutorials
- Google's Machine Learning Crash Course
- Fast.ai Practical Deep Learning

My Resources:

- (Leave blank if no specific resources from your list)

Mini Project:

- **Dataset:** California Housing Dataset
- **Task:** Build and compare 3+ regression models
- **Deliverables:** GitHub repo, model comparison notebook, performance metrics dashboard
- **Acceptance Criteria:** Cross-validation, learning curves, feature importance analysis

Ethics & Best Practices:

- [ ] Check for demographic biases
- [ ] Document model limitations
- [ ] Include confidence intervals
- [ ] Use appropriate metrics
- [ ] Consider fairness metrics

Weekly Outcomes:

- Can train and evaluate ML models
- Understands core ML concepts
- Can use scikit-learn effectively
- Knows model selection basics

Stretch Goals:

- Implement linear regression from scratch
- Add automated hyperparameter tuning

Week 6: Classification Algorithms & Evaluation

Prerequisites: Week 5 ML fundamentals

Topics to Cover:

- Logistic regression
- Decision trees and random forests
- Support vector machines
- Classification metrics (precision, recall, F1, ROC)
- Class imbalance handling

Estimated Time: 13 hours

Tasks:

- Classification algorithms study (3 hrs)
- Implement multiple classifiers (3 hrs)
- Metrics and evaluation deep dive (2.5 hrs)
- Handle imbalanced data (2 hrs)
- Complete mini-project (2.5 hrs)

Resources (Free Only):

- Scikit-learn Classification Tutorial
- StatQuest YouTube Channel
- Google ML Course - Classification
- Towards Data Science Articles

My Resources:

- (Leave blank if no specific resources from your list)

Mini Project:

- **Dataset:** Credit Card Fraud Detection from Kaggle
- **Task:** Build fraud detection system with imbalanced data
- **Deliverables:** GitHub repo, model pipeline, performance report, ethical considerations doc
- **Acceptance Criteria:** Handle class imbalance, achieve >0.9 AUC, include false positive analysis

Ethics & Best Practices:

- [ ] Analyze false positive impact
- [ ] Consider demographic fairness
- [ ] Document decision thresholds
- [ ] Include explainability
- [ ] Test on different populations

Weekly Outcomes:

- Masters classification algorithms
- Can handle imbalanced datasets
- Understands evaluation metrics deeply
- Can choose appropriate algorithms

Stretch Goals:

- Implement ensemble methods
- Add SHAP explanations

Week 7: Feature Engineering & Model Selection

Prerequisites: Weeks 5-6 ML algorithms

Topics to Cover:

- Feature creation and transformation
- Feature selection techniques
- Dimensionality reduction (PCA, t-SNE)
- Cross-validation strategies
- Hyperparameter optimization

Estimated Time: 13 hours

Tasks:

- Feature engineering techniques (3 hrs)
- Feature selection methods (2.5 hrs)
- Dimensionality reduction (2.5 hrs)
- Hyperparameter tuning (3 hrs)
- Complete mini-project (2 hrs)

Resources (Free Only):

- Feature Engineering Book (Free)
- Kaggle Feature Engineering Course
- Scikit-learn Feature Selection
- AutoML with TPOT Tutorial

My Resources:

- (Leave blank if no specific resources from your list)

Mini Project:

- **Dataset:** Kaggle House Prices Competition
- **Task:** Engineer features to improve model performance
- **Deliverables:** GitHub repo, feature engineering pipeline, Kaggle submission
- **Acceptance Criteria:** Create 20+ features, document feature importance, achieve top 50% score

Ethics & Best Practices:

- [ ] Avoid data leakage
- [ ] Document feature logic
- [ ] Consider feature interpretability
- [ ] Validate on holdout set
- [ ] Check for proxy discrimination

Weekly Outcomes:

- Can engineer impactful features
- Understands dimensionality reduction
- Can optimize hyperparameters
- Knows validation strategies

Stretch Goals:

- Implement automated feature engineering
- Create feature store design

Week 8: Advanced ML & Second Capstone

Prerequisites: Weeks 5-7 ML expertise

Topics to Cover:

- Ensemble methods (boosting, stacking)
- Time series forecasting
- Anomaly detection
- Recommender systems basics
- ML system design

Estimated Time: 14 hours

Tasks:

- Ensemble methods deep dive (3 hrs)
- Time series analysis (3 hrs)
- Anomaly detection techniques (2 hrs)
- System design practice (2 hrs)
- Complete capstone project (4 hrs)

Resources (Free Only):

- XGBoost Documentation
- Facebook Prophet Tutorial
- Time Series Analysis Course
- ML System Design Template

My Resources:

- (Leave blank if no specific resources from your list)

Capstone Project:

- **Dataset:** Kaggle Store Sales Forecasting
- **Task:** Build complete ML pipeline with multiple models and AutoML
- **Deliverables:** GitHub repo, model comparison system, API endpoint, Docker container
- **Acceptance Criteria:** 5+ models compared, automated pipeline, API documentation, deployment ready

Ethics & Best Practices:

- [ ] Include model cards
- [ ] Document assumptions
- [ ] Add monitoring hooks
- [ ] Include fairness metrics
- [ ] Version control models

Weekly Outcomes:

- Can build production-grade ML pipelines
- Understands advanced algorithms
- Can design ML systems
- Ready for deep learning

Self-Assessment Checklist:

- [ ] Can implement 10+ ML algorithms
- [ ] Understands feature engineering
- [ ] Can handle any dataset type
- [ ] Has 8+ projects on GitHub
- [ ] Can explain model decisions

Interview Prep Questions:

1. Explain gradient boosting vs random forests
2. How do you handle time series data?
3. Design a recommendation system
4. What is k-fold cross-validation?
5. How do you detect overfitting?

Week 9: Deep Learning Fundamentals

Prerequisites: Week 8 ML mastery, Week 4 mathematics

Topics to Cover:

- Neural network architecture
- Backpropagation and gradients
- Activation functions
- Loss functions and optimizers
- Introduction to PyTorch/TensorFlow

Estimated Time: 13 hours

Tasks:

- Neural network theory (3 hrs)
- PyTorch basics tutorial (3 hrs)
- Build first neural network (3 hrs)

- Optimization techniques (2 hrs)
- Complete mini-project (2 hrs)

Resources (Free Only):

- PyTorch Official Tutorials
- Deep Learning Specialization (Week 1-2)
- Neural Networks and Deep Learning Book
- 3Blue1Brown Neural Network Series

My Resources:

- (Leave blank if no specific resources from your list)

Mini Project:

- **Dataset:** MNIST Handwritten Digits
- **Task:** Build neural network from scratch and with PyTorch
- **Deliverables:** GitHub repo, comparison notebook, training curves visualization
- **Acceptance Criteria:** Achieve >95% accuracy, implement backprop manually, compare with PyTorch

Ethics & Best Practices:

- [ ] Monitor for overfitting
- [ ] Document architecture choices
- [ ] Include training reproducibility
- [ ] Save model checkpoints
- [ ] Track experiments

Weekly Outcomes:

- Understands neural network theory
- Can implement networks from scratch
- Comfortable with PyTorch basics
- Knows optimization techniques

Stretch Goals:

- Implement different optimizers
- Add tensorboard logging

Week 10: Convolutional Neural Networks (CNNs)

Prerequisites: Week 9 deep learning basics

Topics to Cover:

- CNN architecture and convolutions
- Pooling and stride
- Transfer learning
- Data augmentation
- Popular architectures (ResNet, VGG)

Estimated Time: 13 hours

Tasks:

- CNN theory and architectures (3 hrs)
- Implement CNN for images (3 hrs)
- Transfer learning practice (3 hrs)
- Data augmentation techniques (2 hrs)
- Complete mini-project (2 hrs)

Resources (Free Only):

- CS231n Stanford CNN Course
- PyTorch Vision Tutorial
- Keras CNN Tutorial
- Papers with Code - Vision

My Resources:

- (Leave blank if no specific resources from your list)

Mini Project:

- **Dataset:** CIFAR-10 Image Classification
- **Task:** Build custom CNN and fine-tune pretrained model
- **Deliverables:** GitHub repo, model comparison, augmentation study, deployment notebook
- **Acceptance Criteria:** Custom CNN >80% accuracy, transfer learning >90%, augmentation analysis

Ethics & Best Practices:

- [ ] Check for dataset biases
- [ ] Document augmentation choices
- [ ] Include model interpretability

- [ ] Test on diverse images
- [ ] Consider privacy implications

Weekly Outcomes:

- Can design CNN architectures
- Masters transfer learning
- Understands computer vision tasks
- Can handle image datasets

Stretch Goals:

- Implement attention mechanisms
- Add GradCAM visualizations

Week 11: Recurrent Networks & NLP

Prerequisites: Week 9-10 deep learning

Topics to Cover:

- RNN, LSTM, and GRU architectures
- Natural language processing basics
- Word embeddings (Word2Vec, GloVe)
- Sequence-to-sequence models
- Introduction to Transformers

Estimated Time: 13 hours

Tasks:

- RNN/LSTM theory (3 hrs)
- NLP preprocessing pipeline (2.5 hrs)
- Build text classifier (3 hrs)
- Word embeddings practice (2.5 hrs)
- Complete mini-project (2 hrs)

Resources (Free Only):

- Understanding LSTM Networks
- PyTorch NLP Tutorial
- Hugging Face Course

- spaCy 101 Tutorial

My Resources:

- (Leave blank if no specific resources from your list)

Mini Project:

- **Dataset:** IMDB Movie Reviews from Kaggle
- **Task:** Build sentiment analysis system with RNN and Transformer
- **Deliverables:** GitHub repo, model API, performance comparison, word embedding visualizations
- **Acceptance Criteria:** RNN >85% accuracy, implement attention, deploy as API

Ethics & Best Practices:

- [ ] Check for bias in embeddings
- [ ] Document preprocessing steps
- [ ] Include diverse test cases
- [ ] Consider multilingual support
- [ ] Address toxic content handling

Weekly Outcomes:

- Can build NLP pipelines
- Understands sequence models
- Can work with text data
- Knows embedding techniques

Stretch Goals:

- Fine-tune BERT model
- Implement beam search

Week 12: Advanced Deep Learning & Third Capstone

Prerequisites: Weeks 9-11 deep learning

Topics to Cover:

- Generative models (VAE, GAN basics)
- Reinforcement learning introduction
- Model optimization and quantization
- Edge deployment considerations

- Multi-modal learning

Estimated Time: 14 hours

Tasks:

- Generative models study (3 hrs)
- RL basics and examples (2.5 hrs)
- Model optimization techniques (2.5 hrs)
- Integration and review (2 hrs)
- Complete capstone project (4 hrs)

Resources (Free Only):

- GAN Tutorial by Ian Goodfellow
- OpenAI Spinning Up in RL
- Model Optimization Toolkit
- PyTorch Mobile Tutorial

My Resources:

- (Leave blank if no specific resources from your list)

Capstone Project:

- **Dataset:** Kaggle Competition - Choose Current
- **Task:** End-to-end deep learning solution with deployment
- **Deliverables:** GitHub repo, Docker container, model card, API documentation, monitoring setup
- **Acceptance Criteria:** Top 30% performance, fully deployed, includes A/B test plan

Ethics & Best Practices:

- [ ] Include bias evaluation
- [ ] Document model limitations
- [ ] Add robustness testing
- [ ] Include privacy measures
- [ ] Create model card

Weekly Outcomes:

- Can build advanced DL systems
- Understands generative models
- Can optimize for deployment

- Ready for MLOps

Self-Assessment Checklist:

- [ ] Can implement CNNs and RNNs
- [ ] Understands attention mechanisms
- [ ] Has deployed 3+ models
- [ ] Can optimize model performance
- [ ] Portfolio has 12+ projects

Interview Prep Questions:

1. Explain batch normalization
2. What is gradient vanishing/exploding?
3. Design a text classification system
4. How do you prevent overfitting in deep learning?
5. Explain transfer learning benefits

Week 13: MLOps Fundamentals

Prerequisites: Weeks 1-12 ML/DL skills

Topics to Cover:

- MLOps principles and workflow
- Experiment tracking (MLflow, Weights & Biases)
- Model versioning with DVC
- Docker for ML
- CI/CD basics for ML

Estimated Time: 13 hours

Tasks:

- MLOps concepts and tools (3 hrs)
- Setup MLflow tracking (2.5 hrs)
- Docker containerization (3 hrs)
- CI/CD pipeline setup (2.5 hrs)
- Complete mini-project (2 hrs)

Resources (Free Only):

- MLflow Official Documentation
- DVC Get Started
- Docker for Data Scientists
- GitHub Actions for ML

My Resources:

- (Leave blank if no specific resources from your list)

Mini Project:

- **Dataset:** Previous house prices dataset
- **Task:** Add MLOps to existing ML project
- **Deliverables:** GitHub repo with CI/CD, MLflow tracking, DVC pipeline, Docker container
- **Acceptance Criteria:** Automated testing, experiment tracking, model versioning, containerized

Ethics & Best Practices:

- [ ] Version data and models
- [ ] Include reproducibility info
- [ ] Document dependencies
- [ ] Add security scanning
- [ ] Include rollback plan

Weekly Outcomes:

- Can track ML experiments
- Understands version control for ML
- Can containerize ML applications
- Knows CI/CD for ML

Stretch Goals:

- Setup Kubernetes deployment
- Add automated retraining

Week 14: Model Deployment & Serving

Prerequisites: Week 13 MLOps basics

Topics to Cover:

- Model serving architectures

- FastAPI for ML APIs
- TorchServe and TensorFlow Serving
- Batch vs real-time inference
- Cloud deployment (AWS/GCP/Azure free tier)

Estimated Time: 13 hours

Tasks:

- API development with FastAPI (3 hrs)
- Model serving frameworks (3 hrs)
- Cloud deployment practice (3 hrs)
- Load testing and optimization (2 hrs)
- Complete mini-project (2 hrs)

Resources (Free Only):

- FastAPI Tutorial
- TorchServe Documentation
- Google Cloud AI Platform
- AWS SageMaker Free Tier

My Resources:

- (Leave blank if no specific resources from your list)

Mini Project:

- **Dataset:** Use trained sentiment analysis model
- **Task:** Deploy model as production API with monitoring
- **Deliverables:** GitHub repo, deployed API, load testing results, monitoring dashboard
- **Acceptance Criteria:** <100ms latency, handles 100 req/s, includes monitoring, auto-scaling config

Ethics & Best Practices:

- [ ] Include rate limiting
- [ ] Add input validation
- [ ] Log predictions
- [ ] Include fallback behavior
- [ ] Document API thoroughly

Weekly Outcomes:

- Can deploy models to production
- Understands serving architectures
- Can build ML APIs
- Knows cloud deployment basics

Stretch Goals:

- Implement A/B testing
- Add feature flags

Week 15: Monitoring & Maintenance

Prerequisites: Week 14 deployment skills

Topics to Cover:

- Model monitoring and drift detection
- Performance metrics in production
- Logging and debugging ML systems
- A/B testing for ML
- Model retraining strategies

Estimated Time: 13 hours

Tasks:

- Monitoring setup and tools (3 hrs)
- Drift detection implementation (3 hrs)
- A/B testing framework (2.5 hrs)
- Retraining pipeline (2.5 hrs)
- Complete mini-project (2 hrs)

Resources (Free Only):

- Model Monitoring Guide
- Evidently AI Tutorial
- Prometheus and Grafana Setup
- A/B Testing for Data Science

My Resources:

- (Leave blank if no specific resources from your list)

Mini Project:

- **Dataset:** Production data simulator
- **Task:** Build complete monitoring system for deployed model
- **Deliverables:** GitHub repo, monitoring dashboard, alert system, drift detection, retraining trigger
- **Acceptance Criteria:** Detects 3+ drift types, automated alerts, visualization dashboard, retraining pipeline

Ethics & Best Practices:

- [ ] Monitor for bias drift
- [ ] Include privacy in logs
- [ ] Document alert thresholds
- [ ] Test failover scenarios
- [ ] Include audit trail

Weekly Outcomes:

- Can monitor production models
- Detects performance degradation
- Can implement A/B tests
- Knows retraining strategies

Stretch Goals:

- Implement shadow deployment
- Add automated rollback

Week 16: Final Capstone & Career Preparation

Prerequisites: All previous weeks

Topics to Cover:

- End-to-end MLOps pipeline
- Portfolio optimization
- Interview preparation
- System design for ML
- Career strategy

Estimated Time: 14 hours

Tasks:

- System design practice (2 hrs)
- Portfolio review and polish (2 hrs)
- Mock interviews (2 hrs)
- Documentation and presentation (2 hrs)
- Complete final capstone (6 hrs)

Resources (Free Only):

- ML System Design Interview Guide
- Awesome Production ML
- ML Interview Book
- Tech Interview Handbook

My Resources:

- (Leave blank if no specific resources from your list)

Final Capstone Project:

- **Dataset:** Choose a Kaggle Competition or Real Dataset
- **Task:** Build complete ML product with full MLOps pipeline
- **Deliverables:**
  - Production-ready GitHub repo
  - Deployed model with API
  - Monitoring dashboard
  - CI/CD pipeline
  - Documentation site
  - Video presentation
- **Acceptance Criteria:**
  - End-to-end automation
  - Professional documentation
  - Performance benchmarks
  - Security considerations
  - Scalability design

Ethics & Best Practices:

- [ ] Complete ethical review

- [ ] Include bias analysis
- [ ] Document all decisions
- [ ] Add security measures
- [ ] Consider sustainability

Weekly Outcomes:

- Has complete MLOps pipeline
- Portfolio is job-ready
- Prepared for interviews
- Can design ML systems

Self-Assessment Checklist:

- [ ] 16+ projects on GitHub
- [ ] Can build end-to-end ML systems
- [ ] Understands MLOps practices
- [ ] Can deploy and monitor models
- [ ] Ready for ML engineering roles

Interview Prep Questions:

1. Design Netflix recommendation system
2. How do you handle model versioning?
3. Explain your MLOps pipeline
4. Debug a failing production model
5. Design fraud detection system
6. Optimize model for edge deployment
7. Implement gradient descent
8. Explain transformer architecture
9. Design A/B test for model rollout
10. Handle data drift in production

📊 Portfolio Projects Summary

1. **Python Data Analyzer** (Week 1)
2. **COVID-19 Data Pipeline** (Week 2)
3. **World Happiness Dashboard** (Week 3)
4. **Titanic Survival Analysis** (Week 4 - Capstone)

5. **California Housing Predictor** (Week 5)
6. **Credit Card Fraud Detector** (Week 6)
7. **House Price Feature Engineering** (Week 7)
8. **Store Sales Forecasting** (Week 8 - Capstone)
9. **MNIST Neural Network** (Week 9)
10. **CIFAR-10 CNN Classifier** (Week 10)
11. **IMDB Sentiment Analyzer** (Week 11)
12. **Kaggle Competition Solution** (Week 12 - Capstone)
13. **MLOps Pipeline Implementation** (Week 13)
14. **Production Model API** (Week 14)
15. **Model Monitoring System** (Week 15)
16. **Complete ML Product** (Week 16 - Final Capstone)

🎯 Career Readiness Checklist

Technical Skills

- [ ] Python programming mastery
- [ ] Data manipulation (NumPy, Pandas)
- [ ] Machine learning (Scikit-learn)
- [ ] Deep learning (PyTorch/TensorFlow)
- [ ] MLOps tools (Docker, MLflow, DVC)
- [ ] Cloud deployment experience
- [ ] API development
- [ ] Monitoring and maintenance

Portfolio Requirements

- [ ] 16+ GitHub projects with documentation
- [ ] 4 end-to-end capstone projects
- [ ] Deployed models with APIs
- [ ] MLOps pipeline examples
- [ ] Contribution to open source
- [ ] Technical blog posts (optional)

Interview Preparation

- [ ] Data structures and algorithms basics
- [ ] ML theory and mathematics
- [ ] System design for ML

- [ ] Coding challenges practice
- [ ] Behavioral questions preparation
- [ ] Mock interviews completed

Job Search Materials

- [ ] Updated resume with projects
- [ ] LinkedIn profile optimization
- [ ] GitHub profile README
- [ ] Portfolio website (optional)
- [ ] Cover letter template
- [ ] References prepared

🚀 Next Steps After Week 16

1. **Specialize**: Choose a domain (Computer Vision, NLP, RL, etc.)
2. **Contribute**: Join open source ML projects
3. **Compete**: Participate in Kaggle competitions
4. **Network**: Join ML communities and attend meetups
5. **Learn**: Stay updated with latest papers and techniques
6. **Build**: Create your own ML product or startup
7. **Teach**: Write blogs or create tutorials
8. **Certify**: Consider cloud ML certifications

📋 Quick Reference: Tools & Platforms

Development

- **IDE**: VSCode, Jupyter Lab, Google Colab
- **Version Control**: Git, GitHub
- **Environments**: Conda, venv, Docker

ML/DL Frameworks

- **Classical ML**: Scikit-learn, XGBoost, LightGBM
- **Deep Learning**: PyTorch, TensorFlow, Keras
- **NLP**: Hugging Face, spaCy, NLTK

MLOps Tools

- **Tracking**: MLflow, Weights & Biases, TensorBoard
- **Versioning**: DVC, Git-LFS
- **Serving**: FastAPI, TorchServe, TF Serving
- **Monitoring**: Evidently, Prometheus, Grafana

Cloud Platforms (Free Tiers)

- **Google Cloud**: Colab, AI Platform
- **AWS**: SageMaker, EC2 Free Tier
- **Azure**: ML Studio
- **Others**: Paperspace, Kaggle Kernels

JSON Summary

json

```
{
  "roadmap": {
    "duration": "16 weeks",
    "hours_per_week": "12-14",
    "total_projects": 16,
    "capstone_projects": 4,
    "weeks": [
      {
        "week": 1,
        "title": "Python Fundamentals & Development Environment",
        "topics": ["Python basics", "Git/GitHub", "Jupyter notebooks"],
        "project": "Python data analyzer for Iris dataset"
      },
      {
        "week": 2,
        "title": "Data Manipulation with NumPy & Pandas",
        "topics": ["NumPy arrays", "Pandas DataFrames", "Data cleaning"],
        "project": "COVID-19 data analysis pipeline"
      },
      {
        "week": 3,
        "title": "Data Visualization & Statistical Foundations",
        "topics": ["Matplotlib/Seaborn", "Statistics", "Hypothesis testing"],
        "project": "World Happiness interactive dashboard"
      },
      {
        "week": 4,
        "title": "Mathematics for ML & First Capstone",
        "topics": ["Linear algebra", "Calculus", "Probability"],
        "project": "Titanic survival complete EDA pipeline"
      },
      {
        "week": 5,
        "title": "Introduction to Machine Learning",
        "topics": ["ML fundamentals", "Scikit-learn", "Model evaluation"],
        "project": "California housing regression models"
      },
      {
        "week": 6,
        "title": "Classification Algorithms & Evaluation",
        "topics": ["Classification algorithms", "Metrics", "Imbalanced data"],
        "project": "Credit card fraud detection system"
      },
      {
        "week": 7,
        "title": "Feature Engineering & Model Selection",
```

```
  "topics": ["Feature creation", "Selection", "Hyperparameter tuning"],
  "project": "House prices feature engineering competition"
},
{
  "week": 8,
  "title": "Advanced ML & Second Capstone",
  "topics": ["Ensemble methods", "Time series", "System design"],
  "project": "Store sales forecasting with AutoML"
},
{
  "week": 9,
  "title": "Deep Learning Fundamentals",
  "topics": ["Neural networks", "Backpropagation", "PyTorch basics"],
  "project": "MNIST digit classifier from scratch"
},
{
  "week": 10,
  "title": "Convolutional Neural Networks",
  "topics": ["CNN architecture", "Transfer learning", "Data augmentation"],
  "project": "CIFAR-10 image classification"
},
{
  "week": 11,
  "title": "Recurrent Networks & NLP",
  "topics": ["RNN/LSTM", "NLP basics", "Word embeddings"],
  "project": "IMDB sentiment analysis system"
},
{
  "week": 12,
  "title": "Advanced Deep Learning & Third Capstone",
  "topics": ["Generative models", "RL basics", "Model optimization"],
  "project": "Kaggle competition end-to-end solution"
},
{
  "week": 13,
  "title": "MLOps Fundamentals",
  "topics": ["Experiment tracking", "Model versioning", "Docker"],
  "project": "Add MLOps to existing ML project"
},
{
  "week": 14,
  "title": "Model Deployment & Serving",
  "topics": ["FastAPI", "Model serving", "Cloud deployment"],
  "project": "Deploy sentiment model as production API"
},
{
  "week": 15,
```

```
      "title": "Monitoring & Maintenance",
      "topics": ["Model monitoring", "Drift detection", "A/B testing"],
      "project": "Build complete monitoring system"
    },
    {
      "week": 16,
      "title": "Final Capstone & Career Preparation",
      "topics": ["MLOps pipeline", "Portfolio", "Interview prep"],
      "project": "Complete ML product with full MLOps"
    }
  ]
 }
}
```