Applied Data Science with Python

1. Introduction to Data Science

- **Data Science Fundamentals**
  - Definition and Importance of Data Science
  - Lifecycle of Data Science Projects
  - Key Roles in Data Science
- **Data Science vs. Data Analytics**
  - Differences in Scope and Focus
  - Tools and Techniques Comparison
- **Data Science Tools**
  - Overview of Python Libraries: pandas, NumPy, Matplotlib, Seaborn, Scikit-Learn

2. NumPy for Data Science

- **Introduction to NumPy Arrays**
  - Overview of NumPy and its Role in Efficient Data Processing
- **Array Creation**
  - Basic Creation: `np.array()`, `np.zeros()`, `np.ones()`, `np.arange()`, `np.linspace()`
  - Random Arrays: `np.random.rand()`, `np.random.randint()`
- **Basic Operations**
  - Arithmetic Operations: Addition, subtraction, multiplication, division
  - Comparison Operations: Element-wise comparisons
- **Indexing and Slicing**
  - Accessing Elements, Slicing, and Indexing Arrays
  - Boolean Indexing and Fancy Indexing
- **Reshaping and Broadcasting**
  - Reshaping Arrays: `reshape()`, `ravel()`, and `flatten()`
  - Broadcasting Rules and Examples
- **Linear Algebra Operations**
  - Matrix Operations: `dot()`, `matmul()`, `inner()`, `outer()`
  - Determinants, Inverses, and Solving Systems of Equations
- **Advanced Operations**
  - Advanced Indexing Techniques
  - Aggregation Functions: `sum()`, `mean()`, `std()`, `min()`, `max()`
  - Element-wise Operations: Trigonometric, logarithmic, and exponential functions
- **Performance Optimization**
  - Vectorization: Advantages and Use Cases
  - Avoiding Loops with NumPy's Vectorized Operations

3. Linear Algebra for Data Science

- **Vectors and Matrices**
    - Definition and Importance
    - Basic Operations: Addition, subtraction, scalar multiplication
- **Matrix Operations**
    - Transpose, Inverse, and Determinant
    - Eigenvalues and Eigenvectors
- **Vector Spaces**
    - Basis and Dimension, Orthogonality, and Projections
- **Linear Transformations**
    - Application in Data Science
- **Calculus in Linear Algebra**
    - Basics of Derivatives and Optimization Applications

4. Statistics Fundamentals

- **Descriptive Statistics**
    - Measures of Central Tendency: Mean, Median, Mode
    - Measures of Dispersion: Variance, Standard Deviation, Range, Interquartile Range
- **Data Distribution**
    - Normal Distribution and Empirical Rule
    - Skewness and Kurtosis
- **Covariance and Correlation**
    - Calculation and Interpretation of Covariance and Correlation Coefficients
- **Exploratory Data Analysis (EDA)**
    - Visualization Techniques: Histograms, box plots, scatter plots

5. Probability Distributions and Advanced Statistics

- **Probability Theory Basics**
    - Definitions and Conditional Probability
- **Probability Distributions**
    - Normal, Binomial, and Poisson Distributions
- **Advanced Statistical Concepts**
    - Hypothesis Testing, Types of Errors, Z-tests, T-tests, ANOVA
    - Central Limit Theorem (CLT) and Law of Large Numbers

6. Working with Pandas for Data Science

- **Data Structures in Pandas**
    - Series: Creating, indexing, manipulating Series
    - DataFrames: Creating, indexing, manipulating DataFrames
- **Indexing, Selecting, and Filtering**
    - Selecting Data: `loc`, `iloc`, conditional filtering
    - Setting and resetting index

- **Handling Missing Data**
  - Filling Missing Values: `fillna()`, `dropna()`, interpolation
  - Advanced Missing Data Handling Techniques
- **Data Aggregation and Grouping**
  - Grouping Data: `groupby()` and aggregating functions
  - Advanced Aggregation: `agg()`, multi-level aggregation
- **Merging, Joining, and Concatenating**
  - Merging Techniques: `merge()`, `concat()`, `join()`
  - Concatenating along Different Axes
- **Input/Output Operations**
  - Reading/Writing Files: CSV, Excel, SQL, JSON
- **Time Series Analysis**
  - Handling Date/Time Data: Converting to datetime, extracting components
  - Resampling and Aggregation: Resampling techniques for time series
  - Time Zones: Localizing and converting between time zones
- **Advanced Data Aggregation**
  - Pivot Tables and Cross-tabulations
- **Data Transformation**
  - Applying Functions to DataFrames: `apply()`, `transform()`
  - Aggregations, Normalization, and Standardization

7. Data Visualization with Matplotlib

- **Basic Plotting**
  - Essential Plot Types: Line plots, scatter plots, bar charts, histograms
- **Customizing Plots**
  - Adding Titles, Labels, and Legends
  - Customizing Colors, Line Styles, and Markers
- **Subplots**
  - Creating Multiple Plots in a Figure: `subplots()` function
  - Adjusting Layouts and Axes
- **Plot Styling**
  - Applying Themes and Styles
- **Saving and Exporting Plots**
  - Exporting in Various Formats: PNG, PDF, SVG
- **Interactive Plots**
  - Adding Interactivity with `mpl_toolkits`
  - Widgets for Interactive Plotting
- **Custom Plot Types**
  - Creating Custom Plots
  - Combining Multiple Plot Types for Advanced Visualizations

8. Data Visualization with Seaborn

- **Statistical Plots**
  - Boxplot, Violinplot, Pairplot: Use cases and customization
- **Categorical Data Visualization**
  - Bar plots, count plots, and categorical scatter plots
- **Heatmaps and Correlation Plots**
  - Heatmaps for Visualizing Correlations and Data Distributions
- **Customizing Seaborn Plots**
  - Theme Management and Custom Color Palettes
- **Integration with Matplotlib**
  - Using Seaborn with Matplotlib to Enhance Plots
- **Advanced Statistical Plots**
  - Regression Plots, Distribution Plots, and Joint Plots

9. End-to-End Statistics Application in Python

- **Exploratory Data Analysis (EDA)**
  - Applying Descriptive and Inferential Statistics on Real Datasets
  - Statistical Modeling for Hypothesis Testing
- **Data Storytelling**
  - Visualizing Findings and Creating Reports

Machine Learning

1. Machine Learning Fundamentals

- **Introduction to Machine Learning**
  - What is Machine Learning? Definitions and History
  - Types of Machine Learning: Supervised, Unsupervised, Semi-supervised, and Reinforcement Learning
  - Applications of Machine Learning in Real-World Scenarios
- **Machine Learning Pipeline**
  - Steps in the ML Pipeline: Data Collection, Preprocessing, Model Training, Model Evaluation, and Deployment
  - Feature Engineering and Selection
  - Model Evaluation Metrics: Accuracy, Precision, Recall, F1 Score, and AUC-ROC
  - Model Deployment and Monitoring
- **Introduction to MLOps**
  - Overview of MLOps and the Need for Operationalizing Machine Learning
  - MLOps Tools and Frameworks (e.g., MLflow, Kubeflow)
  - Continuous Integration, Deployment, and Model Monitoring in ML

---

2. Supervised Learning

- **Overview of Supervised Learning**
  - Concept and Workflow of Supervised Learning
  - Types of Supervised Learning: Regression and Classification
  - Applications of Supervised Learning in Different Domains
- **Concepts of Overfitting and Underfitting**
  - Definition and Causes of Overfitting and Underfitting
  - Techniques to Detect and Address Overfitting and Underfitting (Cross-validation, Regularization)
  - Bias-Variance Tradeoff and its Importance
- **Data Splitting Techniques**
  - Train-Test Split, K-Fold Cross Validation, and Leave-One-Out Cross Validation

---

3. Regression Models and Applications

- **Introduction to Regression Analysis**
  - Purpose and Use Cases of Regression in Machine Learning

- Types of Regression Models and When to Use Them
- **Simple Linear Regression**
  - Assumptions, Mathematical Formulation, and Use Cases
  - Interpreting Regression Coefficients
- **Multiple Linear Regression**
  - Extending Linear Regression to Multiple Variables
  - Feature Selection for Multivariate Models
- **Polynomial Regression**
  - Dealing with Non-Linear Relationships with Polynomial Terms
- **Ridge and Lasso Regression**
  - Introduction to Regularization Techniques to Handle Multicollinearity
  - Comparison of Ridge and Lasso and Choosing the Appropriate Model
- **Logistic Regression**
  - Regression for Binary and Multi-class Classification Problems
  - Sigmoid Function and Probability Interpretation
- **Evaluation Metrics for Regression**
  - Mean Absolute Error, Mean Squared Error, R-Squared, Adjusted R-Squared

---

4. Classification Models and Applications

- **Overview of Classification Models**
  - Understanding Classification vs. Regression
  - Types of Classification Models and Their Real-world Applications
- **Decision Trees**
  - Structure and Working of Decision Trees
  - Pruning Techniques to Avoid Overfitting
- **Random Forest Classifier**
  - Ensemble of Decision Trees for Improved Accuracy
  - Tuning Hyperparameters of Random Forests
- **k-Nearest Neighbors (k-NN)**
  - Distance Measures and Choosing the Value of K
  - Pros and Cons of Using k-NN for Classification
- **Support Vector Machine (SVM)**
  - Concept of Hyperplanes and Margins in SVM
  - Kernel Trick for Non-linear Data
- **Naive Bayes Classifier**
  - Bayes' Theorem and Assumptions in Naive Bayes
  - Types of Naive Bayes: Gaussian, Multinomial, and Bernoulli
- **Evaluation Metrics for Classification**
  - Precision, Recall, F1 Score, ROC Curve, AUC
  - Confusion Matrix and Interpreting Results

---

5. Unsupervised Learning

- **Overview of Unsupervised Learning**
    - Understanding When and Why to Use Unsupervised Learning
    - Types of Unsupervised Learning Techniques: Clustering, Dimensionality Reduction
- **Clustering Methods**
    - **k-Means Clustering**
        - Choosing the Right Number of Clusters and the Elbow Method
        - Applications and Limitations of k-Means
    - **Hierarchical Clustering**
        - Agglomerative vs. Divisive Clustering
        - Dendrograms and Determining Optimal Clusters
    - **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**
        - Handling Arbitrarily Shaped Clusters and Outliers
        - Comparison of DBSCAN with k-Means and Hierarchical Clustering
- **Dimensionality Reduction Techniques**
    - **Principal Component Analysis (PCA)**
        - Concept of Eigenvalues and Eigenvectors in PCA
        - Choosing the Number of Components for PCA
    - **t-SNE (t-Distributed Stochastic Neighbor Embedding)**
        - Visualizing High-dimensional Data with t-SNE
        - Comparison of t-SNE with PCA for Visualization

---

6. Ensemble Learning

- **Introduction to Ensemble Methods**
    - Purpose of Ensemble Learning and Common Techniques
    - Types of Ensembles: Bagging, Boosting, Stacking
- **Bagging (Bootstrap Aggregating)**
    - Bagging Process and Reducing Variance
    - Example of Bagging Algorithms: Random Forest
- **Boosting**
    - Concept of Boosting and Reducing Bias
    - Popular Boosting Algorithms: AdaBoost, Gradient Boosting, XGBoost, LightGBM, CatBoost
- **Stacking**
    - Stacking Layers of Models to Improve Performance
    - Combining Multiple Models with Meta-learner
- **Evaluation of Ensemble Methods**
    - Comparing Ensembles with Individual Models
    - Pros and Cons of Using Ensemble Learning in Production

---

7. Recommendation Systems

- **Introduction to Recommendation Systems**
  - Types of Recommendation Systems: Content-Based, Collaborative Filtering, Hybrid
  - Key Components and Use Cases in Various Industries
- **Collaborative Filtering**
  - User-based and Item-based Collaborative Filtering
  - Matrix Factorization Techniques: SVD, SVD++
- **Content-Based Filtering**
  - Item Profiling and User Profiling
  - Similarity Measures for Recommendations (Cosine Similarity, Jaccard Similarity)
- **Hybrid Recommendation Systems**
  - Combining Collaborative and Content-Based Filtering
  - Case Studies and Applications of Hybrid Models
- **Advanced Recommendation Techniques**
  - Neural Collaborative Filtering and Deep Learning Approaches
  - Introduction to Reinforcement Learning for Recommendations
- **Building a Recommendation Engine with PyTorch**
  - Overview of PyTorch Framework
  - Implementation Steps: Data Preprocessing, Model Creation, and Evaluation
  - Fine-tuning and Hyperparameter Optimization for Improved Recommendations

---

8. Machine Learning Frameworks

- **Overview of Machine Learning Frameworks**
  - Comparison of Popular ML Frameworks (TensorFlow, Keras, PyTorch)
  - When to Use Each Framework Based on Project Requirements
- **TensorFlow**
  - Introduction to TensorFlow: Tensors, Computation Graphs
  - Building and Training Models with TensorFlow
  - TensorFlow Extended (TFX) for MLOps
- **Keras**
  - Keras API: Layers, Models, and Preprocessing
  - Model Customization and Hyperparameter Tuning
  - Integrating Keras with TensorFlow
- **PyTorch**
  - Tensors and Autograd in PyTorch
  - Building Deep Learning Models with PyTorch
  - Implementing Transfer Learning with PyTorch

- **Comparing Model Performance Across Frameworks**
    - Performance, Ease of Use, Community Support, Deployment

Deep Learning

1. Introduction to Artificial Intelligence and Deep Learning

- **Foundational Concepts of AI and ML**: Differences between AI, ML, and DL; AI types; history and evolution
- **Deep Learning vs. Traditional Machine Learning**: When and why to use deep learning, applications, limitations
- **Key Deep Learning Terminology**: Epochs, learning rate, backpropagation, and gradient descent
- **Types of Deep Learning Architectures**: Feedforward networks, CNNs, RNNs, GANs, autoencoders
- **Overview of Popular Deep Learning Frameworks**: TensorFlow, Keras, PyTorch

---

2. Artificial Neural Networks (ANNs)

- **Structure of an Artificial Neuron**: Activation functions, weights, and biases
- **Activation Functions**: Sigmoid, ReLU, Tanh, Softmax; selecting activation functions for different scenarios
- **Layers in Neural Networks**: Input, hidden, and output layers; dense and fully connected layers
- **Backpropagation and Gradient Descent**: Calculating gradients, updating weights, convergence
- **Implementing a Basic Neural Network in TensorFlow/Keras**: Simple multi-layer perceptron (MLP) for classification

---

3. Deep Neural Network (DNN) and Tools

- **Understanding Deep Neural Networks**: Depth and complexity in DNNs, vanishing/exploding gradient problem
- **TensorFlow and Keras**: Overview of key modules, building blocks (layers, models), and best practices
- **Building Deep Networks**: Defining model architecture, sequential and functional API in Keras
- **Hyperparameter Tuning**: Tuning parameters like learning rate, batch size, optimizer choice
- **Saving and Loading Models**: Checkpoints, saving weights and model configurations
- **Evaluation and Metrics**: Accuracy, loss, confusion matrix, precision-recall, F1-score

---

4. Optimization, Tuning, and Interpretability of Deep Neural Networks

- **Loss Functions**: Mean squared error, cross-entropy, and other common loss functions
- **Optimizers**: SGD, Adam, RMSprop; tuning optimizers for different scenarios
- **Regularization Techniques**: L1, L2 regularization, weight decay
- **Dropout and Batch Normalization**: Avoiding overfitting, improving model generalization
- **Hyperparameter Tuning with Grid and Random Search**: Practical tuning techniques for neural networks
- **Model Interpretability**: Visualizing activations, attention mechanisms, SHAP, LIME

---

5. Convolutional Neural Networks (CNNs)

- **CNN Architecture Basics**: Convolutions, pooling, fully connected layers
- **Convolutional Layers**: Filters, kernel size, strides, and padding
- **Pooling Techniques**: Max pooling, average pooling; understanding spatial dimensionality reduction
- **Building and Training CNNs in TensorFlow**: Constructing a CNN, training with image data
- **Transfer Learning with CNNs**: Using pretrained models (VGG, ResNet) for new tasks
- **Image Data Augmentation**: Rotation, cropping, flipping, and brightness adjustment to avoid overfitting
- **Practical Applications of CNNs**: Image classification, object detection, facial recognition

---

6. Recurrent Neural Networks (RNNs)

- **Introduction to Sequential Data**: Handling time-series data, natural language processing
- **RNN Structure and Workflow**: Recurrence mechanism, vanishing gradient problem in RNNs
- **Types of RNNs**: Vanilla RNNs, LSTMs, GRUs
- **Implementing RNNs in TensorFlow/Keras**: Creating an RNN model for sequence data (e.g., text or time series)
- **Applications of RNNs**: Text generation, machine translation, stock price prediction
- **Bidirectional RNNs and Attention Mechanisms**: Enhancing model accuracy for complex sequences

---

7. NLP with Deep Learning

- **Introduction to NLP and Deep Learning: Overview of NLP applications in deep learning (e.g., text classification, language translation, sentiment analysis)**
- **Text Preprocessing: Tokenization, stopword removal, stemming, lemmatization, embedding representations (Word2Vec, GloVe)**
- **Recurrent Neural Networks (RNNs) for NLP: Sequence handling, challenges with vanilla RNNs for NLP tasks**
- **Long Short-Term Memory Networks (LSTM): Understanding LSTM cells, memory gates, handling long-term dependencies**
- **Gated Recurrent Units (GRU): Comparison with LSTM, fewer parameters, efficient in training**
- **Transformers: Self-attention mechanism, positional encoding, and architecture of Transformers**
- **Applications of Transformers: Machine translation, language generation, named entity recognition**
- **BERT and GPT Models: Transfer learning in NLP, pre-trained models, fine-tuning for specific NLP tasks**
- **Building NLP Models in TensorFlow/Keras: Implementing LSTMs, GRUs, and Transformers for text processing**
- **Practical Applications: Sentiment analysis, text summarization, machine translation**

---

8. Autoencoders

- **Concept of Autoencoders**: Encoding-decoding framework, applications in data compression and noise removal
- **Structure of Autoencoders**: Encoder, latent space, decoder
- **Types of Autoencoders**: Variational Autoencoders (VAEs), Denoising Autoencoders, Sparse Autoencoders
- **Implementing Autoencoders in TensorFlow/Keras**: Basic autoencoder model, reconstructing input data
- **Anomaly Detection with Autoencoders**: Applications in fraud detection, network security, healthcare
- **Dimensionality Reduction and Feature Extraction**: Using autoencoders for compact representations of data

Deep Learning Specialization

Introduction to Deep Learning (Advanced Concepts)

- **Review: Deep Learning vs. Machine Learning**
- **Challenges in Deep Learning**: Overfitting, underfitting, interpretability, and resource-intensive training
- **Overview of Advanced Deep Learning Applications**: Medical imaging, autonomous vehicles, recommendation systems, etc.
- **Ethics and Bias in Deep Learning**

---

2. Advanced Neural Network Architectures

- **Modular and Complex Networks**: ResNet, InceptionNet, DenseNet
- **Attention Mechanisms**: Self-attention and multi-head attention concepts
- **Graph Neural Networks (GNNs)**: Introduction to GNNs and applications in social networks, recommendation engines
- **Capsule Networks**: Understanding capsules and routing-by-agreement mechanisms

---

3. TensorFlow 2 and Keras (Advanced Functionalities)

- **Custom Model Training Loops**: Creating custom training steps and loops in TensorFlow
- **Distributed Training**: Techniques for scaling model training across multiple GPUs/TPUs
- **TensorFlow Extended (TFX)**: Introduction to productionalizing models with TFX
- **Model Deployment with TensorFlow Serving**

---

4. Model Optimization and Performance Improvement

- **Hyperparameter Tuning**: Techniques (Grid Search, Random Search, Bayesian Optimization) and tools (TensorBoard, Keras Tuner)
- **Batch Normalization and Layer Normalization**: Differences, benefits, and implementation in Keras and PyTorch
- **Dropout and Early Stopping**: Concepts, benefits, and application in avoiding overfitting
- **Weight Initialization Techniques**: Xavier, He initialization, and their effect on model performance
- **Advanced Optimizers**: Exploring Adam, Nadam, and RMSprop in-depth
- **Learning Rate Schedulers**: Step decay, exponential decay, and adaptive learning rates
- **Gradient Clipping**: Preventing exploding gradients in deep networks

---

5. Convolutional Neural Networks (CNNs) - Advanced Topics

- **Advanced CNN Architectures**: Understanding ResNet, EfficientNet, and MobileNet architectures
- **Dilated and Depthwise Separable Convolutions**: How they differ from standard convolutions and use cases
- **Object Detection Algorithms**: YOLO, Faster R-CNN, SSD, and their architectures
- **Semantic and Instance Segmentation**: U-Net, Mask R-CNN for detailed pixel-wise predictions
- **3D Convolutions**: Applications in video analysis and medical imaging

---

6. Transfer Learning

- **Transfer Learning Basics**: Benefits, and common approaches (feature extraction, fine-tuning)
- **Pre-trained Models**: Using VGG, ResNet, Inception, BERT, and other architectures in Keras and PyTorch
- **Domain Adaptation**: Adapting models to new domains (e.g., from natural to synthetic data)
- **Using Transfer Learning for Small Datasets**: Techniques and best practices for limited data scenarios

---

7. Object Detection and Image Segmentation

- **Introduction to Object Detection**: Key concepts, challenges, and evaluation metrics (IoU, mAP)
- **Region Proposal Networks (RPN)**: Foundations of Faster R-CNN
- **YOLO (You Only Look Once)**: Understanding the architecture and single-shot detection
- **Instance and Semantic Segmentation**: Overview of segmentation, applications, and architectures like U-Net and Mask R-CNN
- **Applying Object Detection in Real-Time**: Use cases in autonomous driving, security, and AR/VR

---

8. Recurrent Neural Networks (RNNs) - Advanced Applications

- **RNN Architecture Deep Dive**: Revisiting LSTM and GRU internals
- **Attention in RNNs**: Adding attention to RNNs for improved sequence modeling

- **Sequence-to-Sequence Models**: Using encoder-decoder architectures in Keras and PyTorch
- **Advanced Use Cases of RNNs**: Applications in time series forecasting, speech synthesis, and language modeling

---

9. Transformer Models for NLP

- **Attention Mechanism**: Deep understanding of self-attention and cross-attention
- **Transformer Architecture**: Layers, embeddings, multi-head attention, and position encoding
- **BERT, GPT, and T5**: Differences, pre-training, and fine-tuning for NLP tasks
- **Transfer Learning in NLP**: Using pre-trained language models and transfer learning for NLP-specific tasks
- **Implementing Transformers in TensorFlow and PyTorch**: Building and fine-tuning transformer models for NLP tasks

---

10. Advanced Autoencoders

- **Variational Autoencoders (VAE)**: Understanding probabilistic modeling with VAEs
- **Denoising Autoencoders**: Application in noise reduction and feature extraction
- **Sparse and Contractive Autoencoders**: Regularization and constraints for sparse feature learning
- **Applications of Autoencoders in Anomaly Detection and Dimensionality Reduction**

---

11. Introduction to PyTorch

- **PyTorch vs. TensorFlow**: Key differences and use cases for each
- **Basics of PyTorch Tensors and Autograd**: Tensor operations, gradient tracking
- **Building Neural Networks in PyTorch**: Using `torch.nn.Module` and `torch.optim`
- **Custom Training Loops in PyTorch**: Implementing training and validation loops
- **Debugging and Profiling in PyTorch**: Tools for troubleshooting and optimization

---

12. Model Interpretability and Explainability

- **Model Interpretability Concepts**: Importance of interpretability in deep learning
- **Techniques for Model Interpretation**: LIME, SHAP, and Grad-CAM

- **Explainable AI (XAI) Tools**: Using TensorFlow’s `tf-explain` and PyTorch's interpretability libraries
- **Real-World Applications**: Deploying interpretable models in healthcare, finance, and autonomous systems

---