Comprehensive Overview of Machine Learning, Deep Learning, and NLP Models

---

I. Machine Learning (ML) Models and Algorithms

1. Supervised Learning

- **Regression Algorithms**
- Linear Regression
- *Application:* Predicting house prices based on area, location.
- *Code:* from sklearn.linear_model import LinearRegression
- Polynomial Regression
- *Application:* Modeling nonlinear trends in stock prices.
- *Code:* from sklearn.preprocessing import PolynomialFeatures
- Ridge/Lasso Regression
- *Application:* Regularization to avoid overfitting in high-dimensional data.
- *Code:* from sklearn.linear_model import Ridge, Lasso

- **Classification Algorithms**
- Logistic Regression
- *Application:* Email spam detection.
- *Code:* from sklearn.linear_model import LogisticRegression
- Decision Tree
- *Application:* Credit risk assessment.
- *Code:* from sklearn.tree import DecisionTreeClassifier
- Random Forest
- *Application:* Disease prediction from symptoms.
- *Code:* from sklearn.ensemble import RandomForestClassifier
- Support Vector Machine (SVM)
- *Application:* Face detection.
- *Code:* from sklearn.svm import SVC
- K-Nearest Neighbors (KNN)
- *Application:* Handwritten digit recognition.
- *Code:* from sklearn.neighbors import KNeighborsClassifier
- Naive Bayes
- *Application:* Sentiment analysis.
- *Code:* from sklearn.naive_bayes import MultinomialNB

2. Unsupervised Learning

- **Clustering**
- K-Means
- *Application:* Customer segmentation.
- *Code:* from sklearn.cluster import KMeans
- DBSCAN
- *Application:* Anomaly detection in network traffic.
- *Code:* from sklearn.cluster import DBSCAN
- Hierarchical Clustering
- *Application:* Gene expression data analysis.

- *Code:* from scipy.cluster.hierarchy import dendrogram, linkage

- **Dimensionality Reduction**
- PCA (Principal Component Analysis)
- *Application:* Data visualization.
- *Code:* from sklearn.decomposition import PCA
- t-SNE
- *Application:* High-dimensional text data clustering.
- *Code:* from sklearn.manifold import TSNE
- LDA (Linear Discriminant Analysis)
- *Application:* Face recognition.
- *Code:* from sklearn.discriminant_analysis import LinearDiscriminantAnalysis

3. Reinforcement Learning

- Q-Learning
- *Application:* Game AI like Tic-Tac-Toe.
- Deep Q-Network (DQN)
- *Application:* Playing Atari games.
- Policy Gradient Methods
- *Application:* Robotic control.

II. Deep Learning (DL) Models

1. Feedforward Neural Networks (FNN)

- Multilayer Perceptron (MLP)
- *Application:* Predicting numerical outcomes, like stock trends.

2. Convolutional Neural Networks (CNN)

- *Application:* Image classification, object detection, video analysis.

3. Recurrent Neural Networks (RNN)

- LSTM (Long Short-Term Memory)
- GRU (Gated Recurrent Unit)
- *Application:* Time-series forecasting, speech recognition.

4. Generative Models

- GANs (Generative Adversarial Networks)
- *Application:* Deepfake generation, image enhancement.
- VAEs (Variational Autoencoders)
- *Application:* Image compression, data denoising.

5. Transformers

- BERT, GPT, ViT

- *Application:* Language modeling, summarization, translation, image classification.

III. Natural Language Processing (NLP) Models

1. Traditional Models

- BoW, TF-IDF, N-grams

2. Classical Algorithms

- Naive Bayes, SVM, LDA, HMM, CRF

3. Word Embeddings

- Word2Vec, GloVe, FastText

4. Modern NLP Models (Transformer-based)

- BERT, RoBERTa, DistilBERT, GPT, T5, XLNet

IV. Diagrams & Flowcharts

ML Workflow → DL Pipeline → NLP Pipeline → Transformer Architecture

(See diagrams in repo for visuals)