| **Data Type**  **Types of Data and Machine Learning Models to Use to process these data** | **Characteristics** | **Typical Models** | **Optimization Options** | **Frameworks** |
| --- | --- | --- | --- | --- |
| **Video Data**  Video file format Icon, Video Icon File, rectangle, video png | PNGEgg | Temporal and spatial information, large volume, high dimensionality | * CNNs * RNNs * LSTMs * CNN-LSTM * Pre-trained Models (e.g., VGG, ResNet, YOLO) | * Transfer Learning * Data Augmentation * Model Pruning * Hyperparameter Tuning * Efficient Architectures | * TensorFlow * PyTorch * OpenCV |
| **Text Data**  File Text Data Report Blue Dotted Line Line Icon 13122062 Vector Art at ... | Sequential, high dimensionality (vocabulary size), contextual dependencies | * RNNs * LSTMs * GRUs * Transformer models (e.g., BERT, GPT-3) | * Embedding Techniques (Word2Vec, GloVe, BERT) * Data Augmentation (synonym replacement, back-translation) * Transfer Learning * Regularization * Hyperparameter Tuning | * TensorFlow * PyTorch * Hugging Face * NLTK * spaCy |
| **Unstructured Data**  Databse, format, raw data, scattered, unformat, unstructured data icon | Lacks predefined structure, can include a mix of text, images, videos | * Feature Extraction * Combination Models * Custom Pipelines | * Hybrid Models * Custom Feature Engineering * Regularization Techniques (L1/L2, dropout) * Hyperparameter Tuning | * Scikit-learn * TensorFlow * PyTorch * Hadoop * Spark |
| **Streaming Data**  Data, stream, binary, flow icon - Download on Iconfinder | Continuous, real-time, often requires low-latency processing | * Online Learning Algorithms * Models that can handle data in chunks | * Incremental Learning * Windowing Techniques * Latency Optimization * Scalable Architectures | * Apache Kafka * Apache Flink * Spark Streaming * TFX |
| **Time Series Data**  Time, series, data, forecasting, online, audit icon - Download on ... | Sequential data with temporal dependencies, often trends, seasonality | * Statistical Models (ARIMA, SARIMA, Holt-Winters) * RNNs * LSTMs * GRUs | * Seasonal Decomposition * Hyperparameter Tuning * Feature Engineering (lag variables, rolling statistics) * Regularization * Ensemble Methods | * Tsfresh * Scikit-learn * TensorFlow * PyTorch |
| **Tabular Data**  Data Table Vector Icon 16502424 Vector Art at Vecteezy | Structured data with rows and columns, often numerical and categorical features | * Decision Trees * Random Forests * Gradient Boosting Machines (XGBoost, LightGBM, CatBoost) * SVM * Logistic Regression * Linear Regression * Feedforward Neural Networks | * Feature Engineering (PCA, feature importance) * Hyperparameter Tuning * Model Ensembling (bagging, boosting, stacking) * Handling Imbalanced Data (SMOTE) * Regularization * Cross-Validation | * Scikit-learn * XGBoost * LightGBM * CatBoost * TensorFlow * PyTorch |
| **Graph Data**  Concept map, diagram, network icon | Nodes and edges representing relationships | * Graph Neural Networks (GNNs) * Graph Convolutional Networks (GCNs) | * Graph Embeddings * Node2Vec * Optimization of graph structure * Regularization techniques * Hyperparameter tuning | * PyTorch   Geometric   * DGL * NetworkX |
| **Audio Data**  Audio file - Free files and folders icons | Time series data with frequency properties | * CNNs * RNNs * LSTMs * Transformer models * Convolutional Recurrent Neural Networks (CRNNs) | * Spectrogram generation * Data augmentation (noise addition, time shifting) * Transfer learning * Regularization * Hyperparameter tuning | * LibROSA * TensorFlow * PyTorch * Kaldi |
| **Geospatial Data**  Gis - Free ui icons | Spatial coordinates, spatial relationships | * Geospatial models * CNNs * Graph-based models | * Spatial feature extraction * Data augmentation (cropping, rotating) * Transfer learning * Regularization * Hyperparameter tuning | * QGIS * TensorFlow * PyTorch * GeoPandas |
| **Sensor Data**  Sensor Icon in Two Color Design. Red and Black Style Elements from ... | Time series data, often multiple sensors | * Statistical models * RNNs * LSTMs * GRUs * Autoencoders | * Feature extraction (e.g., principal components) * Data normalization * Outlier detection * Regularization * Hyperparameter tuning | * TensorFlow * PyTorch * Scikit-learn |

**Data Types and Machine Learning Models to Process These Data**

General Criteria for ML Model Selection:

* **Computational Resources**: Be very mindful of the resource intensity of different models and the available computational resources. Evaluate GPU and TPU options and their cost tradeoffs.
* **Data Type and Structure**: Choose a model that closely aligns with the data's structure and characteristics.
* **Accuracy vs. Interpretability**: Always weigh the trade-off between the model's accuracy and its interpretability (e.g., decision trees vs. deep learning models).
* **Size and Dimensionality**: Consider whether models can handle high-dimensional data or require dimensionality reduction.
* **Scalability and Deployment**: Ensure the model can be scaled and deployed effectively in the production environment.
* **Domain-Specific Requirements**: Consider any specific requirements and constraints of the application domain.