That's a great question that expands on the evaluation concepts already in the Canvas. Understanding the different types of models is key to selecting the right algorithm for your data.

Here is a breakdown of the main types of models within Supervised and Unsupervised machine learning, along with common examples for each.

### I. Supervised Machine Learning

Supervised learning involves training a model on a labeled dataset, meaning the training data includes both the input features (1X) and the desired output (the target, 2Y).3 The model learns a mapping function from X to Y.

Supervised learning is divided into two primary problem types:4

#### A. Classification Models

These models predict a discrete category or label (e.g., 50 or 61, Red or Blue).7

| Model Type | Best Used For | Python Example (Scikit-learn) |
| --- | --- | --- |
| **Logistic Regression** | Binary classification problems, especially when linearity and interpretability are important. | LogisticRegression() |
| **Support Vector Machines (SVM)** | Complex classification tasks with clear margin of separation; effective in high-dimensional spaces. | SVC() |
| **Decision Trees** | High interpretability; forms a flow-chart-like structure based on feature splitting. | DecisionTreeClassifier() |
| **Random Forest** | An ensemble method that combines many decision trees to improve accuracy and reduce overfitting. | RandomForestClassifier() |
| **K-Nearest Neighbors (KNN)** | Simple, non-parametric model where classification is based on the majority vote of the K closest data points. | KNeighborsClassifier() |

#### B. Regression Models

These models predict a continuous value or quantity (e.g., price, age, temperature).

| Model Type | Best Used For | Python Example (Scikit-learn) |
| --- | --- | --- |
| **Linear Regression** | Modeling the linear relationship between features and a target variable. **(Used in Section B of the Canvas)** | LinearRegression() |
| **Ridge and Lasso Regression** | Regularized versions of linear regression used to prevent overfitting by penalizing large coefficients. | Ridge(), Lasso() |
| **Decision Tree Regressor** | Predicting continuous values using decision tree logic (splits data based on features to find local averages). | DecisionTreeRegressor() |
| **Gradient Boosting (e.g., XGBoost, LightGBM)** | Highly performant ensemble methods that build models sequentially, where each new model corrects errors of the previous ones. | GradientBoostingRegressor() |

### II. Unsupervised Machine Learning

Unsupervised learning involves training a model on an unlabeled dataset. The model must learn the inherent structure, patterns, or relationships within the data without any external guidance.8

Unsupervised learning is generally divided into Clustering and Dimensionality Reduction:9

#### A. Clustering Models

These models group similar data points together into clusters, where data points within a cluster are more similar to each other than to those in other clusters.

| Model Type | Best Used For | Python Example (Scikit-learn) |
| --- | --- | --- |
| **K-Means Clustering** | Simple and efficient for segmenting data into a pre-defined number (K) of clusters. | KMeans() |
| **DBSCAN (Density-Based Spatial Clustering)** | Identifying clusters of varying shapes and densities in data that may contain noise and outliers. | DBSCAN() |
| **Agglomerative Clustering** | A hierarchical method that starts with each data point as its own cluster and merges them based on similarity. | AgglomerativeClustering() |

#### B. Dimensionality Reduction Models

These models reduce the number of features (dimensions) while retaining most of the essential information, which helps speed up model training and visualize high-dimensional data.

| Model Type | Best Used For | Python Example (Scikit-learn) |
| --- | --- | --- |
| **Principal Component Analysis (PCA)** | Linearly transforming the features into a new, lower-dimensional space, capturing maximum variance. | PCA() |
| **t-distributed Stochastic Neighbor Embedding (t-SNE)** | Non-linear technique primarily used for visualizing high-dimensional data in 2 or 3 dimensions. | TSNE() |
| **Linear Discriminant Analysis (LDA)** | Similar to PCA, but supervised (it uses class labels) and aims to maximize the separation between classes. | LinearDiscriminantAnalysis() |