# Machine Learning Cheatsheet

## 1. Core Concepts

- \*\*Supervised Learning\*\*: Model is trained on labeled data. Examples: regression, classification.  
- \*\*Unsupervised Learning\*\*: Model finds patterns in unlabeled data. Examples: clustering, dimensionality reduction.  
- \*\*Reinforcement Learning\*\*: Model learns by interacting with an environment and receiving rewards.  
- \*\*Overfitting\*\*: Model performs well on training data but poorly on test data. Solution: regularization, more data.  
- \*\*Underfitting\*\*: Model is too simple and performs poorly on both training and test data.  
- \*\*Bias-Variance Tradeoff\*\*: Balance between underfitting (high bias) and overfitting (high variance).

## 2. Evaluation Metrics

- \*\*Accuracy\*\*: (TP + TN) / (TP + TN + FP + FN). Good for balanced classes.  
- \*\*Precision\*\*: TP / (TP + FP). Focuses on false positives.  
- \*\*Recall\*\*: TP / (TP + FN). Focuses on false negatives.  
- \*\*F1 Score\*\*: 2 \* (Precision \* Recall) / (Precision + Recall). Harmonic mean of precision and recall.  
- \*\*AUC-ROC\*\*: Area under ROC curve, shows model's ability to distinguish between classes.

## 3. Data Preprocessing

- \*\*Normalization\*\*: Scales data to [0, 1] range.  
- \*\*Standardization\*\*: Centers data to mean 0 and std 1.  
- \*\*Missing Value Imputation\*\*: Fill missing data using mean, median, mode, or model-based methods.  
- \*\*Encoding\*\*: One-hot encoding or label encoding for categorical data.  
- \*\*Outlier Handling\*\*: Use IQR or z-score to detect and treat outliers.

## 4. Feature Engineering

- \*\*Feature Selection\*\*: Choose the most relevant features (e.g., with mutual information or correlation).  
- \*\*Feature Extraction\*\*: Create new features from existing data (e.g., PCA, text vectorization).  
- \*\*Polynomial Features\*\*: Create interaction terms and non-linear features.  
- \*\*Binning\*\*: Group numeric features into intervals (useful in decision trees).

## 5. Algorithms Overview

- \*\*Linear Regression\*\*: For predicting continuous values. Assumes linear relationship.  
- \*\*Logistic Regression\*\*: For binary classification. Outputs probabilities.  
- \*\*Decision Trees\*\*: Non-linear models that split data into decision nodes.  
- \*\*Random Forests\*\*: Ensemble of decision trees to improve accuracy and reduce overfitting.  
- \*\*Support Vector Machines (SVM)\*\*: Finds optimal separating hyperplane between classes.  
- \*\*KNN\*\*: Classifies based on majority vote of k nearest neighbors.  
- \*\*Naive Bayes\*\*: Probabilistic classifier based on Bayes' Theorem.  
- \*\*Neural Networks\*\*: Composed of layers of neurons; capable of learning complex patterns.

## 6. Model Validation

- \*\*Train/Test Split\*\*: Divide dataset into train and test parts.  
- \*\*Cross-Validation\*\*: Divide dataset into k-folds; train/test on different splits to improve reliability.  
- \*\*Grid Search\*\*: Try different hyperparameters and select the best model.  
- \*\*Random Search\*\*: Randomly sample hyperparameters for efficiency.

## 7. Imbalanced Data Techniques

- \*\*SMOTE\*\*: Synthetic Minority Over-sampling Technique.  
- \*\*Undersampling\*\*: Reduce size of the majority class.  
- \*\*Class Weights\*\*: Adjust model to penalize misclassification of minority class.

## 8. Time Series Concepts

- \*\*Rolling Statistics\*\*: Moving average or standard deviation over a window.  
- \*\*Stationarity\*\*: Constant mean and variance over time. Use ADF test.  
- \*\*ARIMA\*\*: Model for stationary time series.  
- \*\*LSTM\*\*: Neural network suited for sequential data.

## 9. Deployment & Monitoring

- \*\*Model Serialization\*\*: Save model using pickle, joblib, or ONNX.  
- \*\*API Deployment\*\*: Serve model through REST API (e.g., Flask, FastAPI).  
- \*\*Monitoring\*\*: Track model drift, latency, and performance in production.