ML Cheatsheet

A comprehensive cheat sheet covering core machine learning algorithms, evaluation metrics, and essential concepts for interview preparation. Includes supervised, unsupervised learning, deep learning and NLP.

Supervised Learning: Regression

Linear Regression

Description: Models the relationship between a dependent variable and one or more independent variables by fitting a linear equation to observed data.

Formula: $y = \beta_0 + \beta_1x_1 + \beta_2x_2 + ... + \beta_nx_n + \epsilon$

Assumptions: Linearity, independence, homoscedasticity, normality of residuals.

Use Cases: Predicting sales, estimating prices, forecasting demand.

Advantages: Simple, easy to interpret, computationally efficient.

Disadvantages: Sensitive to outliers, assumes linearity, can suffer from multicollinearity.

Regularization: Not inherently regularized. Use Ridge or Lasso for regularization.

Ridge Regression

Description: Linear regression with L2 regularization. Adds a penalty term equal to the square of the magnitude of coefficients.

Formula: Minimize $\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} \beta_jx_{ij})^2 + \alpha \sum_{j=1}^{p} \beta_j^2$

Effect of α: Controls the strength of regularization. Higher α shrinks coefficients towards zero, reducing overfitting.

Use Cases: When multicollinearity is present, or to prevent overfitting.

Advantages: Reduces overfitting, handles multicollinearity better than linear regression.

Disadvantages: Requires tuning of the regularization parameter α, less interpretable than linear regression.

Lasso Regression

Description: Linear regression with L1 regularization. Adds a penalty term equal to the absolute value of the magnitude of coefficients.

Formula: Minimize $\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} \beta_jx_{ij})^2 + \alpha \sum_{j=1}^{p} |\beta_j|$

Effect of α: Controls the strength of regularization. Higher α can lead to feature selection (some coefficients become exactly zero).

Use Cases: Feature selection, when many features are irrelevant.

Advantages: Performs feature selection, reduces overfitting, handles multicollinearity.

Disadvantages: Can arbitrarily select one feature among correlated features, requires tuning of the regularization parameter α.

Notes: L1 regularization.

Supervised Learning: Classification

Logistic Regression

Description: Models the probability of a binary outcome using a logistic function.

Formula: $p(y=1|x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1x_1 + ... + \beta_nx_n)}}$

Use Cases: Binary classification problems like spam detection, disease prediction.

Advantages: Simple, interpretable, provides probability estimates.

Disadvantages: Assumes linearity, can suffer from overfitting with high-dimensional data.

Regularization: Can be regularized using L1 or L2 regularization to prevent overfitting.

k-Nearest Neighbors (k-NN)

Description: Classifies data points based on the majority class among its k nearest neighbors.

Algorithm:

1. Choose the number of neighbors k.
2. Calculate the distance between the query point and all other data points.
3. Select the k nearest neighbors.
4. Assign the class label based on majority vote.

Use Cases: Recommendation systems, pattern recognition, image classification.

Advantages: Simple, no training phase, versatile.

Disadvantages: Computationally expensive, sensitive to irrelevant features, requires appropriate choice of k.

Distance Metrics: Euclidean, Manhattan, Minkowski.

Decision Trees

Description: A tree-like model that makes decisions based on features. Each node represents a feature, each branch represents a decision rule, and each leaf represents an outcome.

Splitting Criteria: Gini impurity, entropy, information gain.

Use Cases: Classification and regression tasks, feature selection, interpretable models.

Advantages: Easy to understand and interpret, handles both categorical and numerical data, can capture non-linear relationships.

Disadvantages: Prone to overfitting, can be sensitive to small changes in the data.

Ensemble Methods: Random Forests, Gradient Boosting.

Model Evaluation and Tuning

Evaluation Metrics

Accuracy:
\frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}
Useful when classes are balanced.

Precision:
\frac{\text{True Positives}}{\text{True Positives + False Positives}}
Ability of the classifier not to label as positive a sample that is negative.

Recall:
\frac{\text{True Positives}}{\text{True Positives + False Negatives}}
Ability of the classifier to find all the positive samples.

F1-Score:
2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision + Recall}}
Weighted average of Precision and Recall.

ROC-AUC: Area Under the Receiver Operating Characteristic curve.
Measures the ability of a classifier to distinguish between classes.

Confusion Matrix: A table summarizing the performance of a classification model.

Model Tuning

Cross-validation:
A technique for evaluating model performance by partitioning the data into subsets for training and validation.
Common methods include k-fold cross-validation.

Bias-variance tradeoff:
Finding the right balance between bias (underfitting) and variance (overfitting) is crucial for model generalization.

Overfitting/underfitting:
Overfitting: Model performs well on training data but poorly on unseen data.
Underfitting: Model fails to capture the underlying patterns in the data.

Hyperparameter tuning:
Optimizing model hyperparameters to improve performance.
Techniques include GridSearchCV and RandomizedSearchCV.

GridSearchCV:
Exhaustively searches through a specified subset of the hyperparameters.

RandomizedSearchCV:
Randomly samples a given number of candidates from a parameter space.

Deep Learning Fundamentals

Core Concepts

Neural Networks:
Models composed of interconnected nodes (neurons) organized in layers.
Learn complex patterns through weighted connections and activation functions.

Perceptron:
A single-layer neural network.
Forms the basis for more complex networks.

Activation Functions:
Introduce non-linearity to the network, enabling it to learn complex relationships.
Examples: ReLU, sigmoid, tanh.

Backpropagation:
An algorithm for training neural networks by iteratively adjusting the weights based on the error between predicted and actual outputs.

Loss Functions:
Quantify the error between predicted and actual outputs.
Examples: Mean Squared Error (MSE), Cross-Entropy.

Optimizers:
Algorithms that update the network's weights to minimize the loss function.
Examples: Adam, SGD, RMSprop.

Convolutional Neural Networks (CNNs)

Description:
Specialized for processing structured array of data, such as images.
Use convolutional layers to automatically learn spatial hierarchies of features.

Key Layers:
Convolutional layers, pooling layers, fully connected layers.

Use Cases:
Image classification, object detection, image segmentation.

Advantages:
Efficiently captures spatial dependencies, robust to variations in position and scale.

Disadvantages:
Can be computationally expensive, require large datasets.