Machine Learning: Data Science and ML Refresher

Shubhankar Agrawal

Abstract

This document serves as a quick refresher for Data Science and Machine Learning interviews. It covers mathematical and technical concepts across a range of algorithms. This requires the reader to have a foundational level knowledge with tertiary education in the field. This PDF contains material for revision over key concepts that are tested in interviews.

Contents

1. Machine Learning

1.1. Key Concepts

Data Splits:
Train: The model learns from it.
Valid: Tune hyper-parameters, prevent over-fitting.

Table 1. Data Types

Data	Type	Description
Nominal	Categorical	No order
Ordinal	Categorical	Ordered
Interval	Numerical	Can be negative
Ratio	Numerical	Has a defined 0

Test: Unseen data, evaluate performance.

Table 2. Bias Variance Trade-Off

	Bias	Variance
What?	Error	Prediction Variability
Complexity	Too Simple	Too Complex
Fitting	Under	Over
Train Error	High	Low
Test Error	High	High
Formula	$\text{Bias}(\hat{\theta}) = \mathbb{E}[\hat{\theta}] - \theta$	$\text{Var}(\hat{\theta}) = \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2]$

Types of Learning:
Supervised: Labelled data **Unsupervised**: Unlabelled data **Reinforcement**: Learn with feedback from environment
Other terms:
Parameters: Weights the model learns
Hyper-parameters: Weights to adjust performance
Cross Validation: Expose all data (K Fold, LOOCV, Temporal)

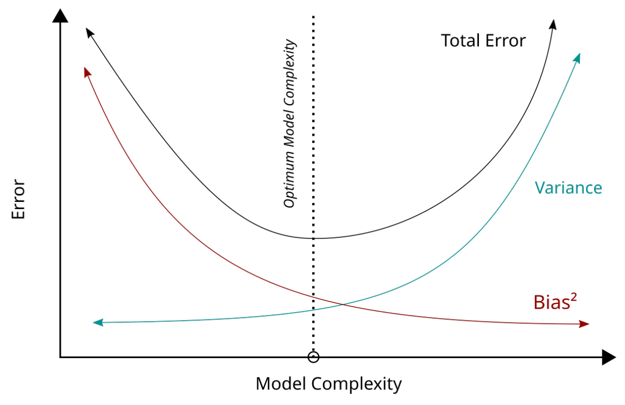

Figure 1. Bias Variance Trade-off
[2]

2. Exploratory Data Analysis

2.1. General Statistics

Pandas operations to analyse dataframes:
info: Information on null values, data types, and memory
describe: Descriptive statistics of mean, median and IQR
value_counts: Counts for categorical columns

2.2. Univariate Analyses

2.2.1. Target Variable Distribution

:

Continuous: Plot distribution, identify outliers
Discrete: Value counts to check for imbalanced data

2.2.2. Feature Distributions

:

Continuous: Box plots, Histograms
Discrete: Bar charts of value counts

2.3. Bi-variate Analyses

Table 3. Bi-variate Analyses

Comparing	Compared	Plots
Continuous	Continuous	Scatter
Continuous	Discrete	Violin, Bar, Line
Discrete	Discrete	Grouped Bars

Pair plots from Seaborn can plot different types of features against each other in a single graphic.

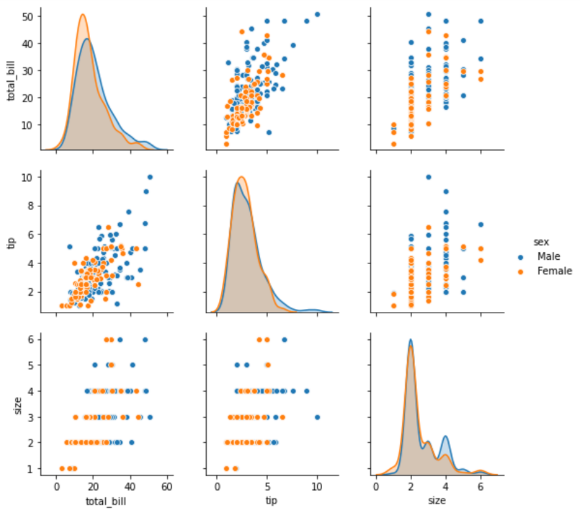

Figure 2. Pair Plots
[4]

2.4. Multi-variate Analyses

- Heat maps
- Violin Plots (Binary categories)
- Scatter Plots (with Sizes)
- Correlation matrices

3. Data Pre-Processing

3.1. Data Cleaning

NOTE: Use median instead of mean when the data is skewed.

3.1.1. Missing Values

Remove if not too many (especially in Target)
Impute: Fill values
Interpolate: Estimate the line
Ways to fill values:

- Fill with Mean / Median by groups (or) nearest neighbours
- Analyse temporal patterns to fill by dates
- Fit a regression line to fill missing values

3.1.2. Outliers

Identify by Box Plots, Anomaly Detection

- Remove if not too many.
- Winsorize (clip by IQR)

3.1.3. Erroneous Values

Some values might be errors and can be fixed

- Amounts scaled by 10 / 100
- Negative values in a non-negative space
- Invalid values (such as co-ordinates)

3.1.4. Others

- Drop duplicates
- Fix entities (NYC vs New York City)
- Date formats

3.2. Feature Scaling

3.2.1. Standardization

- Calculate Z Scores
- Useful to bring features to similar distributions
- Robust to outliers
- Better for SVM, Linear Regression

Can also use IQR to scale more robustly.

3.2.2. Normalization

- MinMax Scaling
- Good for data without outliers
- Better for KNN, Neural Networks

3.2.3. Box Cox Transform

Log (or) Power transform maximizing normality

$$\max_{\lambda}\ \ell(\lambda) = -\frac{n}{2}\log(\sigma^2) + (\lambda - 1)\sum_{i=1}^{n}\log(y_i) \tag{1}$$

Table 4. Feature Scaling

Method	Formula
Standardization	$x_{\text{std}} = \dfrac{x-\mu}{\sigma}$
Normalization	$x_{\text{norm}} = \dfrac{x-\min(x)}{\max(x)-\min(x)}$
Box Cox Transform	$y(\lambda) = \begin{cases} \dfrac{(x^{\lambda}-1)}{\lambda} & \text{if } \lambda \neq 0 \\ \log(x) & \text{if } \lambda = 0 \end{cases}$

3.3. Target Scaling

Log Scale: When extremely skewed, but reduces interpretability (With co-efficients)

4. Feature Engineering

4.1. Extraction

4.1.1. Text

- TF-IDF scores
- Sentence Embeddings

4.1.2. Pixels

- Intensity
- Hue
- Brightness

4.1.3. Temporal

- Year, Month, Day
- Part of day, Weekday
- Fourier Transforms (sin, cos) for periodicity

4.2. Transformation

4.2.1. Temporal

Add temporal changes and history

$$\begin{aligned}\text{Lags @ } x_t &= x_{t-7}\\ \text{Rolling Mean @ } x_t &= (x_{t-2}+x_{t-3}+x_{t-4})/3\\ \text{Difference @ } x_t &= x_t/x_{t-1}\end{aligned} \tag{2}$$

Differencing also brings stationarity

4.2.2. Complexity

Introduce non-linearity

$$\begin{aligned}\text{Polynomial @ } x_i &= x_i^2 + x_i^3\\ \text{Interaction @ } x_t &= x_i x_j + x_i x_j^2\end{aligned} \tag{3}$$

4.3. Encoding

Converting categorical to numerical variables. Some models can handle categorical data, so not necessary.

Table 5. Encoding

Type	How	Cardinality	Columns
One Hot	0/1 Dummies	Low	# distinct values
Ordinal	Order + Scale	Any	-
Target	Group Mean	High	-

NOTE: Encoding should only be done using train data

- Identify data types
- Generate lags, averages, aggregate features
- Feature Pre-processing

4.4. Sampling

Usually performed when data is imbalanced.
Downsample: Reduce instances of majority class
Upsample: Increase instances of minority class
Upsampling involves creating instances:

- SMOTE - Interpolate points in space
- Variational Auto Encoder - Learn Distribution

4.5. Selection

4.5.1. Iterative

Sequentially add or remove features one by one to optimize performance

Table 6. Iterative Feature Selection

Step	Forward	Backward
Start	0 features	All features
Step	Feature to add	Feature to remove

4.5.2. Model Based

Use model capabilities to identify important features

Table 7. Model based Feature Selection

Model	Identification
Lasso (L1)	0 co-efficients - Remove
Random Forest	Feature Importances - Descending

4.5.3. Statistical Tests

Statistical tests can provide comparison of significance by looking at:

- Test Statistic (direction of influence)
- P-value (Acceptance)

Table 8. Statistical Test

Feature	Target	Test
Continuous	Continuous	T-Test / Z-Test
Continuous	Discrete	ANOVA (F-Test)
Discrete	Continuous	ANOVA (F-Test)
Discrete	Discrete	Chi Square

Multicollinearity can also be removed

- Correlation matrix (Identify highly correlated features)
- Variance Inflation Factor (VIF > 5 or 10)

VIF is calculated by regressing each feature on the other features

$$\text{VIF}_i = \frac{1}{1-R_i^2} \tag{4}$$

4.6. Dimensionality Reduction

Principal Components Analysis:

- Standardize data (Z Score)
- Compute covariance matrix
- Eigenvalue Decomposition

Linear Discriminant Analysis:

- Group by classes
- SSW (Sum of Squares **Within** classes)
- SSB (Sum of Squares **Between** classes) [Multiply # class samples]
- SSB / SSW
- Eigenvalue Decomposition

Table 9. Dimensionality Reduction

	PCA	LDA	t-SNE	UMAP
Labels?	No	Yes	No	Both
Linear	Yes	Yes	No	No
Preserves	Global	Classes	Local	Global
Best For	Linear Patterns	Classify	Visualize	Clustering Embeddings
Issues	Non-linear	Labels	Local	Tuning

5. Algorithms

Examples of different types:
Supervised: Most models with labelled data
Unsupervised: Clustering, Variational Auto-Encoders (VAE)
Parametric: Most models with weights to learn
Non-Parametric: K Nearest Neighbours, Decision Trees
Discriminative: Most models that predict
Generative: Naive Bayes, Latent Dirichlet Analysis, VAEs

5.1. Regression

Predicting a continuous variable.

$$y = X\beta + \varepsilon \tag{5}$$

Interpretation: A change in X by 1 unit, increases / decreases y by β units.

5.1.1. OLS

Ordinary Least Squares, closed Form

$$\begin{aligned}\hat{\beta} &= \arg\min_{\beta}(y - X\beta)'(y - X\beta)\\ \hat{\beta} &= (X'X)^{-1}X'y\end{aligned} \tag{6}$$

5.1.2. Gradient Descent

Iterative Solution

Table 10. Gradient Descent Terminology

Variable	Symbol	Description
Loss/Cost Function	J	Penalizes predictions
Learning Rate	α	Learning step size

$$\begin{aligned}\text{MSE } J(\beta) &= \frac{1}{2n}\sum_{i=1}^{n}(y_i - X_i\beta)^2\\ \beta^{(t+1)} &= \beta^{(t)} - \alpha\nabla J(\beta^{(t)})\\ \nabla J(\beta) &= -\frac{1}{n}X'(y - X\beta)\\ \beta^{(t+1)} &= \beta^{(t)} + \eta\cdot\frac{1}{n}X'(y - X\beta^{(t)})\end{aligned} \tag{7}$$

Standard Error

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-k-1} \tag{8}$$

Confidence Interval (Co-efficients)

$$\begin{aligned}\text{Conf Int } \beta &= \beta_j \pm t_{\alpha/2,n-k-1}\cdot \text{SE}(\beta_j)\\ \text{SE}(\beta_j) &= \sqrt{\frac{\hat{\sigma}^2}{\mathbf{X}^\top\mathbf{X}_{jj}}}\end{aligned} \tag{9}$$

Confidence Interval

Range of mean predicted value

$$\begin{aligned}\textbf{Conf Int} &= \hat{y}_* \pm t_{\alpha/2,n-k-1}\cdot \text{SE}(\hat{y}_*)\\ \text{SE}(\hat{y}_*) &= \sqrt{\hat{\sigma}^2\cdot\left(\mathbf{x}_*^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{x}_*\right)}\end{aligned} \tag{10}$$

Prediction Interval

Range of newly predicted value, includes observation noise.

$$\begin{aligned}\text{Pred Int} &= \hat{y}_* \pm t_{\alpha/2,n-k-1}\cdot \text{SE}(\hat{y}_*)\\ \text{SE}(\hat{y}_*) &= \sqrt{\hat{\sigma}^2\cdot\left(1 + \mathbf{x}_*^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{x}_*\right)}\end{aligned} \tag{11}$$

5.1.3. Metrics

$$\begin{aligned}\text{MAE} &= \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|\\ \text{MAPE} &= \frac{1}{n}\sum_{i=1}^{n}|\frac{y_i - \hat{y}_i}{y_i}|\cdot 100\\ \text{MSE} &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\\ \text{RMSE} &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}\\ R^2 &= 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}\\ R^2_{\text{adj}} &= 1 - \left(\frac{(1-R^2)(n-1)}{n-p-1}\right)\end{aligned} \tag{12}$$

Notes:

- R^2 increases with variables; use adjusted R^2
- Use RMSE instead of MSE to stay in the same scale

5.1.4. Regularization

Control co-efficients preventing them from getting too large

Table 11. Regression Regularization

Type	Term	Gradient
Lasso (L1)	$\lambda\sum\lvert\beta_j\rvert$	$\lambda\cdot\text{sign}(\beta_j)$
Ridge (L2)	$\frac{\lambda}{2}\sum\beta_j^2$	$2\lambda\beta_j$
Elastic Net	$\frac{\lambda}{2}\sum\beta_j^2 + \lambda_1\sum\lvert\beta_j\rvert$	$2\lambda\beta_j + \lambda_1\cdot\text{sign}(\beta_j)$

5.1.5. Assumptions

- Data follows linear relationship
- Errors are normally distributed
- Errors are homo-skedastic (Constant variance)
- No multicollinearity of features
- No auto-correlation of errors

Fixes: Log transforms, outlier removals

5.2. Classification

Predicting a discrete variable, a binary or a multi-class.

5.2.1. Logistic Regression

Fits a Linear Regression to the **log odds**
Terminology

$$\begin{aligned}\text{Odds} &= \frac{p}{1-p}\\ \text{Log-Odds} &= \ln\left(\frac{p}{1-p}\right) = \beta_0 + \cdots + \beta_k x_k\\ \text{Odds Ratio (OR)} &= e^{\beta_j}\end{aligned} \tag{13}$$

Interpretation: A change in X by 1 unit, increases / decreases the Log Odds by β units.

$$\begin{aligned}z &= \beta_0 + \beta_1x_1 + \beta_2x_2 + \cdots + \beta_kx_k\\ p &= \sigma(z) = \frac{1}{1+e^{-z}}[\text{Sigmoid}]\end{aligned} \tag{14}$$

Table 12. Logistic Regression Co-efficients

Beta	Effect
1	No effect ($p = 1 - p$)
< 1	Decreases odds
> 1	Increases odds

$$\text{Log Loss} = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)]$$
$$\nabla_{\beta_j}\text{Log Loss} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)\,x_{ij} \tag{15}$$
$$\beta_j = \beta_j - \eta\cdot\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)\,x_{ij}$$

Multi-class Classification

$$p_{ik} = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}$$
$$\text{Cross Entropy Loss} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik}\log(p_{ik}) \tag{16}$$
$$\nabla_{z_k}\text{Loss} = p_{ik} - y_{ik}$$
$$\beta_j^{(k)} = \beta_j^{(k)} - \eta\cdot\frac{1}{n}\sum_{i=1}^{n}(p_{ik} - y_{ik})x_{ij}$$

Table 13. Multi Class Classification

	Binary	Multiclass
Loss	Binary Log Loss	Cross Entropy
Activation	Sigmoid	Softmax

5.2.2. Naive Bayes

Key Assumptions:

- Features are independent
- Continuous features follow Gaussian distribution

$$P(C_k \mid X_1, X_2, \ldots, X_n) = \frac{P(C_k)\prod_{i=1}^{n} P(X_i \mid C_k)}{P(X_1, X_2, \ldots, X_n)}$$
$$\hat{C} = \arg\max_{C_k} P(C_k)\prod_{i=1}^{n} P(X_i \mid C_k) \tag{17}$$

5.2.3. Metrics

Table 14. Confusion Matrix

Real / Pred	True	False
True	True Positive (TP)	False Negative (FN)
False	False Positive (FP)	True Negative (TN)

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$
$$\text{Precision} = \frac{TP}{TP + FP}$$
$$\text{Recall / Sensitivity (TPR)} = \frac{TP}{TP + FN}$$
$$\text{Specificity (TNR)} = \frac{TN}{TN + FP}$$
$$\text{F1 Score} = 2\cdot\frac{\text{Precision}\cdot\text{Recall}}{\text{Precision} + \text{Recall}}$$
$$\text{F}_\beta = (1+\beta^2)\cdot\frac{\text{Precision}\cdot\text{Recall}}{(\beta^2\cdot\text{Precision}) + \text{Recall}} \tag{18}$$

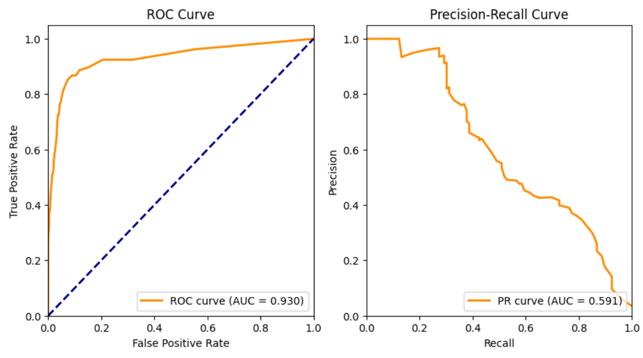

Figure 3. AUC ROC (and) AUC PR [1]

5.2.4. Assumptions

Similar assumptions as linear regression on fit

- Logits follow linear relationship
- Data is linearly separable
- Categories are mutually exclusive

5.3. More Methods

Methods that can be used for both regression and classification

5.3.1. K Nearest Neighbours

- Non-parametric method
- Does not have any training
- Evaluation done by aggregating nearest points
- Cross-validate to get best K value

5.3.2. Support Vector Machines

Fits a hyperplane
Support Vectors: Points on margin

$$f(x) = w^T x + b$$
$$\text{Minimize: } \frac{1}{2}\|w\|^2 \tag{19}$$

Classification
Constraints

$$y_i(w^T x_i + b) \geq 1, \quad \forall i \tag{20}$$

Hinge Loss

$$\text{L} = \frac{1}{n}\sum_{i=1}^{n}\max(0, 1 - y_i\hat{y}_i) \tag{21}$$

Multi-class classification

One vs One (OvO): Fit n^2 models
One vs Rest (OvR: Fit n models
Regression
Constraints

$$\begin{aligned} y_i - (w^T x_i + b) &\le \epsilon + \xi_i^+, \\ (w^T x_i + b) - y_i &\le \epsilon + \xi_i^- \end{aligned} \tag{22}$$

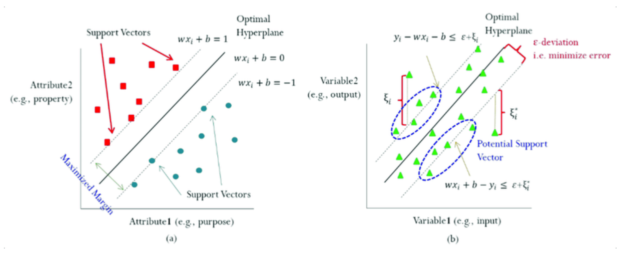

Figure 4. Support Vectors
[5]

Kernels

- Linear
- Polynomial
- Radial Basis Function (RBF): Exponential equation

NOTE: RBF Non-parametric (dimensions $\propto$ samples)

Table 15. Regularization with C in SVM

C	E.g.	Margin	Slack / Errors
Low	0.1	Wide	High Slack
Moderate	1.0	Balanced	Moderate Slack
High	10	Narrow	Low Slack
Very High	0^6	Very Narrow	Very Low Slack

5.3.3. *Decision Trees*

Tree-based structure to perform splits.
Can be considered a non-parametric model for not making assumptions on data distribution.

- Identify best feature to split on
- Recursively continue splits
- Calculate predictions by average of node values

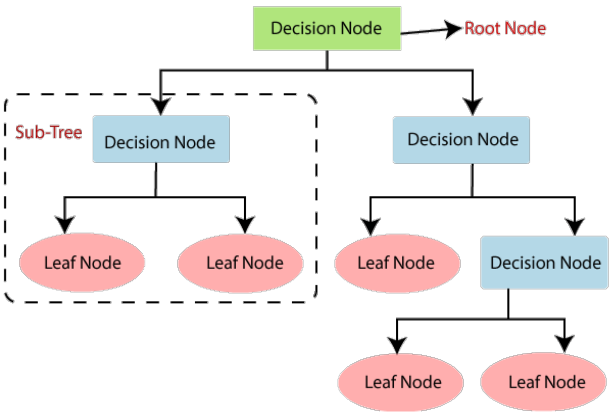

Figure 5. Decision Tree
[3]

Split Calculations

- Maximize Information Gain
- Minimize Entropy
- Minimize Gini
- Minimize MSE (Regression)

Discrete: Evaluate each categorical value vs others
Continuous: Evaluate boundaries where predictions change

$$\begin{aligned} \text{Gini}(t) &= 1 - \sum_{i=1}^{k} p_i^2 \\ \text{Entropy}(t) &= -\sum_{i=1}^{k} p_i \log_2(p_i) \\ \text{Information Gain}(t, A) &= \text{Entropy}(t) - \sum_{v \in \text{Values}(A)} \frac{|t_v|}{|t|} \text{Entropy}(t_v) \\ \text{Intrinsic Information}(A) &= -\sum_{v \in \text{Values}(A)} \frac{|t_v|}{|t|} \log_2\left(\frac{|t_v|}{|t|}\right) \\ \text{Gain Ratio}(A) &= \frac{\text{Information Gain}(A)}{\text{Intrinsic Information}(A)} \end{aligned} \tag{23}$$

C 5.0 Algorithm

- Use Gain Ratio for optimized splits
- Winnowing (Remove features least used)
- Prune with Cost Complexity (# Leaf Nodes, Entropy)

Hyper-parameters
Can be used for Regularization (with Pruning)

Table 16. Hyper-parameters

Hyper-parameter	Use
max_depth	Maximum depth of tree
min_samples_split	Minimum samples to split
min_samples_leaf	Minimum samples at leaf

5.4. Ensemble Methods

5.4.1. *Bagging*

Bootstrap Aggregating models

- Run parallel
- Minimize variance

$$\text{Var}(\hat{y}) = \text{Var}\left(\frac{1}{n}\sum_{i=1}^{n} \hat{y}_i\right) = \frac{1}{n^2}\sum_{i=1}^{n} \text{Var}(\hat{y}_i) = \frac{1}{n^2} \cdot n \cdot \text{Var}(\hat{y}_i) \tag{24}$$

Random Forest

- Randomly subset samples at each node
- Randomly subset features to test at each node

Out of Bag: Remaining samples not used for validation score

5.4.2. *Boosting*

Sequentially built models

- Run iteratively
- Minimize bias
- Gradient Descent
- Sum predictions from weighted models

Adaptive Boosting (AdaBoost)
Weight samples higher for misclassification

$$F(x) = \sum_{m=1}^{M} \alpha_m h_m(x)$$
$$\alpha_m = \frac{1}{2}\ln\left(\frac{1-\epsilon_m}{\epsilon_m}\right)$$
$$D_m(x) = D_{m-1}(x)\cdot\exp\left(-\alpha_m y_m h_m(x)\right)$$
$$\begin{aligned}&\text{where}\\ F(x) &= \text{Final Prediction}\\ h_m(x) &= \text{Prediction from the } m^{\text{th}} \text{ model}\\ \alpha_m &= \text{Weight for the } m^{\text{th}} \text{ model, based on its accuracy}\\ \epsilon_m &= \text{Weighted error rate of the } m^{\text{th}} \text{ model}\\ D_m(x) &= \text{Weight of sample } x \text{ after the } m^{\text{th}} \text{ model update}\end{aligned} \tag{25}$$

Gradient Boosting
Add learners on residuals from previous models

$$\begin{aligned} F_M(x) &= F_{M-1}(x) + \eta h_M(x)\\ r_i &= -\nabla L(F_{M-1}(x_i))\\ F(x) &= \sum_{m=1}^{M}\eta h_m(x)\\ &\text{where}\\ F_M(x) &= \text{Prediction from the } M^{\text{th}} \text{ iteration of the model}\\ F_{M-1}(x) &= \text{Prediction from the } (M-1)^{\text{th}} \text{ iteration of the model}\\ h_m(x) &= \text{Model at the } m^{\text{th}} \text{ iteration (fit on residuals)}\\ r_i &= \text{Residual (negative gradient of the loss function)}\\ L &= \text{Loss function, used to compute the residuals}\end{aligned} \tag{26}$$

Table 17. Gradient Boosting

	LightGBM	XGBoost
Growth	Leaf Wise	Depth Wise
Categorical	Direct	Encoding
Memory	Efficient	Not so much

LightGBM:

- Histogram based approach to optimize splits
- Gradient Based One Sided Sampling (GOSS)
- Exclusive Feature Bundling (EFB)

$$\text{Leaf Value } w_j = -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda} \tag{27}$$

5.4.3. Stacking

Combine advantages of many models

- Train several base models
- Train meta model - cross validated predictions of base models

Hard Voting: Majority prediction
Soft Voting: Weighted average (accuracies)

5.5. Clustering

5.5.1. K-Means

Unsupervised approach to group data

- Assumes spherical clusters
- Results depend on initialization
- Follows Expectation (Assign cluster) Maximization (recalculate centroid)

$$J(K) = \sum_{k=1}^{K}\sum_{x_i\in C_k} ||x_i - \mu_k||^2 \tag{28}$$

NOTE: KMeans++ can be used to distance centroid initialization

5.5.2. DB Scan

Density based clustering

- Needs # Points and Minimum Distance
- Does not need number of clusters
- Identifies outliers

$$\text{core point: Number of points within distance } \epsilon \geq \text{minPts} \tag{29}$$

5.5.3. GMM

Gaussian Mixture Models: Soft clustering

- Assume several Gaussian distributions
- Model latent parameters

$$P(x) = \sum_{k=1}^{K}\pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k) \tag{30}$$

5.5.4. Metrics

Silhouette Score: Intra-cluster vs inter-cluster distance.
Range: -1 to 1

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{31}$$

Where:

- $s(i)$: Silhouette score for point i,
- $a(i)$: Mean intra-cluster distance (average distance of i to all other points in the same cluster),
- $b(i)$: Mean nearest-cluster distance (average distance of i to points in the nearest cluster).

Adjusted Rand Index: Similarities by pairwise points
Range: 0 to 1

$$ARI = \frac{\text{Index} - \text{Expected Index}}{\text{Max Index} - \text{Expected Index}} \tag{32}$$

Where:

- Index: Number of point pairs assigned to the same or different clusters in both ground truth and predicted clusters
- Expected Index: The expected value of the Index if clusters were randomly assigned
- Max Index: The maximum possible value of the Index

6. More Techniques

6.1. Anomaly Detection

- Isolation Forest (Random Forest - Average depth)
- DB Scan

6.2. Reinforcement Learning

Receptive environment-based algorithm with feedback

6.2.1. Policy Learning

Bellman Equation: Model-based recursive equation for state value updates.

$$
\begin{aligned}
V(s) &\leftarrow \max_a \sum_{s'} P(s'|s,a)\,[R(s,a,s') + \gamma V^*(s')] \\
V^\pi(s) &= \sum_a \pi(a|s) \sum_{s'} P(s'|s,a)\,[R(s,a,s') + \gamma V^\pi(s')]
\end{aligned} \tag{33}
$$

NOTE: Requires known probabilities and rewards

6.2.2. Q-Learning

Model-free algorithm to decide best actions from trial and error.

$$
\begin{aligned}
Q(s,a) &\leftarrow Q(s,a) + \alpha \left[R(s,a,s') + \gamma \max_{a'} Q(s',a') - Q(s,a)\right] \\
Q^\pi(s,a) &= \sum_{s'} P(s'|s,a) \left[R(s,a,s') + \gamma \sum_{a'} \pi(a'|s') Q^\pi(s',a')\right] \\
\pi(s) &= \arg\max_a Q(s,a)
\end{aligned} \tag{34}
$$

6.2.3. Exploration Exploitation

Balance exploration (to search new paths) and exploitation (capitalize on high rewards)

ϵ-Greedy

$$
a = \begin{cases} \text{random action} & \text{with probability } \epsilon, \\ \arg\max_a Q(s,a) & \text{with probability } 1-\epsilon. \end{cases} \tag{35}
$$

7. Nuances

7.1. Imbalanced Data

- Data Sampling
- Weighted Loss Functions
- Tree Based Methods (Robust)
- Precision-Recall instead of ROC (False Positives)

7.2. Biases

- Identify difference in distributions for features
- Up-sample data across biased attributes
- Normalize with respect to groups
- Mask / Group together data
- Embed to lower dimension with VAE

References

[1] *Area Under Curves*. [Online]. Available: https://juandelacalle.medium.com/how-and-why-i-switched-from-the-roc-curve-to-the-precision-recall-curve-to-analyze-my-imbalanced-6171da91c6b8.

[2] *Bias Variance Tradeoff*. [Online]. Available: https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff.

[3] *Decision Trees*. [Online]. Available: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm.

[4] *Pair Plots*. [Online]. Available: https://www.geeksforgeeks.org/python-seaborn-pairplot-method/.

[5] *Suport Vectors*. [Online]. Available: https://www.researchgate.net/figure/Overview-of-SVM-algorithm-a-SVM-for-classification-b-SVM-for-regression_fig1_347831458.