Introduction to Machine Learning Analysis and Modeling

Evan Misshula

June 7, 2025

g

End to end process

Recall ML workflow is a sequence of steps to build and deploy a model that solves a problem using data.

The pipeline

:BEAMER$_{\text{env}}$: block :END:

Ingestion & Preprocessing	Analysis	Modeling	Deployment
Definition	EDA	Selection	Tuning
Data Collection	Feature Engineering	Training	Deployment
Cleaning		Evaluation	Monitoring

ML Workflow Graph

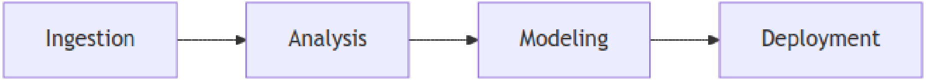
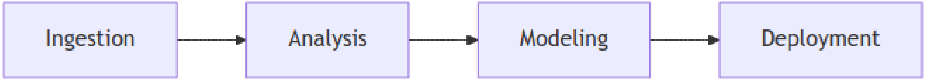


Figure: ML workflow steps rendered as a flowchart



Exploratory Data Analysis

Exploratory Data Analysis (EDA) in the context of machine learning (ML) refers to the systematic process of examining and visualizing the structure, patterns, anomalies, and relationships within a dataset before applying machine learning algorithms. The goal is to gain intuition and insight about the data to inform: Understand the distribution of each feature (e.g., normality, skewness, outliers).
Assess relationships between input features and the target variable (e.g., correlation, mutual information).

Exploratory Data Analysis in Pandas

Pandas tools:

'.info()' Structure Overview

```
df.info()
```

- Displays a concise summary of the DataFrame:
  - Number of non-null values per column
  - Data types of each column
  - Memory usage
  - Total number of rows and columns

Pandas example

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 5 columns):
```

Pandas ‘.describe()‘

‘.describe()‘ Summary Statistics

```
df.describe()
```

- Returns summary statistics for numeric columns:
  - ‘count‘, ‘mean‘, ‘std‘, ‘min‘, ‘25%‘, ‘50%‘, ‘75%‘, ‘max‘

```
              age        income
count   950.000000  1000.000000
mean     35.5       60000.0
std      10.0       15000.0
min      18.0       20000.0
25%      28.0       50000.0
50%      35.0       60000.0
75%      42.0       70000.0
max      65.0       120000.0
```

Non-numeric results

For non-numeric columns:

```
df.describe(include='object')
```

- Shows: ‘count‘, ‘unique‘, ‘top‘, ‘freq‘

Pandas Tool Summary

Method	Purpose	Applies To
‘df.info()‘	Structure & metadata	All columns
‘df.describe()‘	Descriptive stats (summary)	Numeric by default

Univariate analysis Visualize

Look at your data

- Histogram + KDE $\rightarrow$ quick skew/kurtosis check.
- Q-Q Plot $\rightarrow$ best for tail behavior.
- Boxplot $\rightarrow$ highlights symmetry and outliers.

[Live Code 2]

Univariate Analysis Tests

Tests for Skewness and Kurtosis

- D'Agostino's K^2 Test: Combines measures of skewness and kurtosis.
  - Based on transformations of the sample skewness and kurtosis.
  - Null Hypothesis: The data is normally distributed.
  - Available in 'scipy.stats.normaltest'.
  - JarqueBera Test:
    - Specifically evaluates skewness and excess kurtosis against a normal distribution.
    - Null Hypothesis: Data is normally distributed.

Summary Univariate Analysis

Interpretation

- Low p-value (< 0.05): Reject null $\rightarrow$ evidence of non-normal skew/kurtosis.
- High p-value (0.05): Fail to reject null $\rightarrow$ no evidence of non-normality.

Motivation for Multivariate EDA

- Univariate EDA is insufficient for understanding dependencies and structure in multivariate data.
- Multivariate EDA focuses on relationships, redundancy, and conditional structure across features.
- Goal: Identify informative, redundant, or interacting features.

Joint and Marginal Distributions

- Let $X = (X_1, X_2, \ldots, X_d) \in \mathbb{R}^d$ be a random vector.
- The joint distribution P_X describes full probabilistic structure.
- The marginal distribution of a feature X_i is obtained by integrating out all other variables.
- Understanding joint vs. marginal behavior is central to multivariate EDA.

Statistical Dependence

- Two variables X and Y are independent if:

$$P_{X,Y}(x,y) = P_X(x)P_Y(y)$$

- EDA seeks to discover dependencies between variables.
- Classical tools: covariance, correlation but these are limited to linear dependence.

Mutual Information

- Mutual Information (MI) is a nonparametric measure of dependence:

$$I(X;Y) = \int\int p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right) dxdy$$

- $I(X;Y) = 0$ iff $X \perp Y$.
- Captures all kinds of dependence not just linear.

Connection to KL Divergence

- Mutual Information is a special case of the Kullback-Leibler divergence:

$$I(X;Y) = D_{\mathrm{KL}}(P_{X,Y} \| P_X \otimes P_Y)$$

- It measures how far the joint distribution is from the product of the marginals.
- Interpreted as: How surprising is the joint distribution, compared to independence?

Why It Matters in EDA

- Helps detect feature redundancy or relevance.
- Basis for feature selection and structure learning.
- Multivariate visualizations (pair plots, heatmaps, etc.) are motivated by mathematical notions of dependence.

[Live Code]

What is KL Divergence?

KL Divergence is a measure of how one probability distribution Q differs from a reference distribution P.

- It is not symmetric: $D_{KL}(P \parallel Q) \neq D_{KL}(Q \parallel P)$
- KL divergence is always non-negative: $D_{KL}(P \parallel Q) \geq 0$
- $D_{KL}(P \parallel Q) = 0$ if and only if $P = Q$ almost everywhere

Mathematical Definition (Discrete)

Let P and Q be probability mass functions over a finite or countable set $\mathcal{X}$.

$$D_{KL}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)} \tag{1}$$

- The log is typically taken to base 2 (bits) or base e (nats)
- Requires $Q(x) > 0$ wherever $P(x) > 0$

Interpretation

- Measures the expected number of extra bits needed to code samples from P using a code optimized for Q
- It is the relative entropy of P with respect to Q

Mathematical Definition (Continuous)

Let $p(x)$ and $q(x)$ be probability density functions over a domain $\mathcal{X} \subseteq \mathbb{R}^n$:

$$D_{KL}(P \parallel Q) = \int_{\mathcal{X}} p(x) \log \frac{p(x)}{q(x)} \, dx \tag{2}$$

- Again, the divergence is zero iff $p(x) = q(x)$ almost everywhere

Practical Calculation

Given empirical data samples $x_1, \ldots, x_n \sim P$, estimate KL divergence:

- Use histograms or kernel density estimators (KDE) to estimate $p(x)$, $q(x)$
- Approximate:

$$\hat{D}_{KL}(P \parallel Q) = \frac{1}{n} \sum_{i=1}^{n} \log \frac{p(x_i)}{q(x_i)} \tag{3}$$

- Common in variational inference and mutual information estimation

KL Divergence vs. Other Measures

Measure	Symmetric	Interpretable
KL Divergence		Code inefficiency
Jensen-Shannon		Interpolated KL
Mutual Information		Redundancy

Summary of KL Divergence

- KL divergence quantifies divergence from a reference distribution
- Central to many ML methods: variational inference, GANs, language modeling
- Not symmetric, not a true metric
- Requires careful estimation for continuous variables

What is Feature Engineering?

- Feature engineering is the process of transforming raw data into meaningful input features for machine learning models.
- It involves:
  - Creating new features
  - Modifying existing ones
  - Selecting the most relevant subset
- The goal is to enhance model performance by exposing the most useful signal in the data.

Why is Feature Engineering Important?

- Quality of features often outweighs choice of algorithm.
- Poor features = poor model performance, regardless of the model used.
- Good features can:
  - Improve accuracy
  - Speed up training
  - Reduce overfitting
  - Make models interpretable

Common Types of Feature Engineering

- Normalization/Scaling: StandardScaler, MinMaxScaler
- Encoding: One-hot, Label encoding
- Discretization/Binning
- Polynomial Features: Capture interactions
- Date/Time decomposition: Day, month, weekday, etc.
- Log transformations: For skewed distributions

Feature Selection and Extraction

- Feature Selection: Identify and keep the most relevant variables.
  - RFE (Recursive Feature Elimination):
    - Iteratively builds a model and removes the least important feature.
    - Works with any estimator that exposes ‘coef_‘ or ‘feature$_{importances}$_‘.
- Feature Extraction: Derive new features from raw data.
  - t-SNE (t-distributed Stochastic Neighbor Embedding):
    - A nonlinear dimensionality reduction technique.
    - Preserves local structure; useful for visualizing high-dimensional data.
  - UMAP (Uniform Manifold Approximation and Projection):
    - Similar to t-SNE but faster and better preserves global structure.
    - Based on topological and geometric foundations.

Best Practices and Guidelines

- Understand the data context and business goals.
- Visualize feature distributions and relationships.
- Watch out for data leakage.
- Use cross-validation to evaluate engineered features.

Summary of Feature Engineering

- Feature engineering is essential for successful modeling.
- Methods like RFE, t-SNE, and UMAP help in selection and dimensionality reduction.
- Combining domain knowledge with statistics is key.