ML Pipline

DATE
PAGE

1. Feature Engineering

Feature Engineering is the process of using domain knowledge to create new features (variables) from existing data to improve the performance of machine learning models. It is a crucial step as the performance of a model is heavily dependent on the quality of the features provided.

Key aspect include :

Creation : Generating new features, e.g., creating "FamilySize" from "SibSp" and "Parch" in the Titanic dataset.

Transformation : Modifying existing features, such as converting a "Date" string into separate 'Day', 'Month' and 'Year' components, or extracting the title from a 'Name' field (Mr., Mrs., Miss).

Aggregation : Summarizing data, like calculating the average spending per customer.

Handling Missing Values : Imputing missing data using mean, median, mode or more advanced techniques like K-Nearest Neighbors.

The goal is to provide the model with more informative and relevant inputs, which can lead to significantly better predictive accuracy.

DATE
PAGE

2. Feature Selection:

Feature Selection is the process of automatically or manually selecting a subset of the most relevant features from the original dataset from model construction. It is essential to avoid the "curse of dimensionality" and build simpler, faster, and more interpretable models.

Benefits include:

Reduces Overfitting: less redundant data means less opportunity for the model to learn noise.

Improves Accuracy: By removing irrelevant features, the model can focus on the tree signals.

Reduces Training Time: Fewer features means fewer computations.

Enhances Model Interpretability: It's easier to understand a model build with 10 features than one with 1000.

Feature selection methods are broadly classified into three categories: Filter methods, wrapper methods, and Embedded ~~med~~ methods.

DATE
PAGE

3. Filter Based Approach.

The Filter Based approach is a feature selection method that selects features based on their intrinsic statistical properties, independent of any machine learning model. It uses a statistical measure ("a "Filter") to score each feature and select the top-ranked ones.

Characteristics:

Model Agnostic: It is not tied to a specific learning algorithm, making it computationally cheap and fast.

Univariate: Typically, it evaluates each feature in isolation, ignoring feature dependencies.

Common Statistical Measures: Includes correlation co-efficient, Information Gain, Chi-Square Test, and fisher score.

Process: Calculate a score for each feature.
Rank the features based on their scores.
Select the top-K features or all features above a certain threshold. While efficient, its main drawback is that it may fail to select the optimal feature subset if features are correlated, as it does not consider feature interactions.

DATE
PAGE

4. Information Gain

Information Gain is a filter-based feature selection metric commonly used for classification tasks. It measures how much "information" a feature provide about the target class. It is based on the concept of Entropy from information theory.

Explanation:

Entropy: Measures the impurity or uncertainty in a dataset. A high entropy means the data is mixed (50% class A, 50% class B), while low entropy means the data is pure (100% class A).

Information Gain: It is the reduction in entropy achieved by partitioning the data according to a feature. A feature that perfectly separates the classes will have a high Information Gain.

Mathematically, $IG(T, A) = Entropy(T) - Entropy(T/A)$.
where T is the target and A is the features.

Features with high information Gain are considered more important for predicting the target variable, as they bring the most order to the data.

DATE
PAGE

5. Chi - Square Test

The Chi-Square (X^2) test is a statistical filter method used for feature selection in categorical datasets. It assesses the independence between a feature and the target class. The fundamental hypothesis is that the target and the feature are independent.

Process: For each categorical feature, a contingency table is built against the target class.

The Chi-square statistic is calculated, which measures the divergence of the observed frequencies from the frequencies expected if the feature and target were independent.

A high Chi-Square value indicates that the observed and expected counts are significantly different, leading to the rejection of the null hypothesis. This means the feature is dependent on the target and thus is relevant for classification

It is a powerful method for selecting categorial features but is not suitable for continuous data unless they are first discretized (binned).

DATE
PAGE

6. Fisher Score (and the Fisher Bone Method)

The Fisher Score is a filter-based feature selection techniques that selects features which best separate the data points from different classes. It does this by maximising the ratio of the between class variance to the within-class variance for each feature.

Institution:

A Good feature should have:

High between-class variance: The means of different classes should be far apart.

Low within-class variance: The data points within the same class should be close to each others.

Calculation: For a feature, the Fisher score is calculated as: (SB/SW) where SB is b/w-class scatter and SW is within-class scatter.

A Higher Fisher Score indicates a feature with better discriminatory power. The term "Fisher Bone Method" is likely a mishearing or misspelling of the Fisher-Rao method or simply refers to the core ("bone") principle of the Fisher Score.

DATE
PAGE

7. Handling Missing Values

Handling missing values is a critical step in data preprocessing, as most machine learning algorithms cannot work with incomplete data. The strategy depends on the nature and amount of the missing data.

Common Techniques:

Deletion:

Listwise Deletion: Removing entire rows with any missing value. Suitable only if the data is large and the missing values are random.

Imputation (Filling in Values):

For Numerical Data: Use mean, median or mode.
For Categorical Data: Use the mode (most frequent category)

Advanced Methods: Use algorithms like K-Nearest Neighbors (KNN) or regression models to predict and fill missing values based on other features.

Flagging: Creating an additional binary feature to indicate whether the value was missing, which can sometimes be informative for the model.

Choosing the right method is essential to prevent bias and maintain the integrity of the dataset

DATE
PAGE

8. Wrapper Method :

Wrapper methods are a category of feature selection that uses the performance of a specific machine learning model to evaluate the quality of a feature subset. They "wrap" around a predictive model and use its performance as the objective function.

Common Approaches :

Forward Selection : Starts with no features and iteratively adds the features the most improves the models performance.

Backward Selection : Starts with all features and iteratively removes the least significant feature.

Recrusive Feature Elimination (RFE) : A popular form of backward elimination that recrusively fits the model and removes the weakest features.

Advantages : They are very efficient and can capture feature interactions.

Disadvantages : They are computationally very expensive and prone to overfitting, especially with large feature sets, as they require training a model for every candidate feature subset.

DATE
PAGE

9. Exhaustive Feature selection

Exhaustive Feature Selection is a wrapper method that aims to find the best subset of features by evaluating all possible feature combinations. It trains and tests a machine learning model for every single possible subset of features.

Process: For a dataset with n features, it evaluates $2^n - 1$ possible subsets (excluding the empty set). The subset that results in the highest model performance (e.g., accuracy) is selected as the optimal feature set.

Advantage: It guarantees finding the best possible subset of features for a given model and performance metric.

Disadvantage: It is a computationally prohibitive and infeasible for even a moderately large number of features (e.g., 20 features would require (1,04,857 model evaluations), making it impractical for most real-world scenarios.

DATE
PAGE

10. Recrusive Feature Elimination (RFE)

Recrusive Feature Elimination (RFE) is a popular and efficient wrapper-style feature selection method. It works by recrusively removing the least important features and building a model on the remaining features.

Steps:

Train a model (like a linear model or SVM that can provide feature importance) on the entire set of features.

Rank the features based on their importance (e.g., coefficients in linear models).

Remove the least important feature(s) from the current set.

Repeat steps 1-3 with the reduced feature set until the desired number of features is reached.

RFE is computationally more feasible than exhaustive search and effectively finds a high-performing subset of features. It is widely used with models like Support Vector Machines (SVM) and Logistic Regression.

DATE
PAGE

11. Project: Titanic Dataset Feature Engineering.
In the Titanic Dataset, effective feature engineering can significantly boost model performance by creating more meaningful predictors from raw data.

1. Family and Sibling Features:
The raw features 'SibSp' (Siblings/spouse) and 'Parch' (parents/children) can be combined.

Family Size = SibSp + Parch + 1 (including the passengers themselves). This captures the total family size on ~~board~~ broad.

IsAlone: A binary feature (1 if Family Size == 1, else 0). This can be a strong indicator, as solo travelers might have had a different survival rate.

2. Ticket Price and Class Features:
While 'Pclass' (ticket class) is a direct indicator of socio-economic status, combining it with 'Fare' (ticket price) can reveal inconsistencies or more nauced information.

Fare per Person = Fare / Family size: This normalizes the fare, as a single ticket might have been purchased for an entire family. A high fare per person in a lower class might be a significant indicator.

Analyzing Discrepancies: A passanger in Pclass = 1 with a very low fare, or in Pclass = 3 with a

very high fare, could be engineered into a feature flag, potentially indicating data errors or special circumstances.

These engineered features help the model capture complex relationships that the original raw features alone might miss.

DATE
PAGE

12. Encoding Categorical Variables

Most machine learning algorithms require numerical input. Encoding is the process of converting categorical variables (text or categories) into a numerical format.

Common Techniques:

Label Encoding: Assigns a unique integer to each category (e.g. Red=0, Green=1, Blue=2). Suitable for ordinal data (where order matters, like "Low", "Medium", "High").

One-Hot Encoding: Creates new binary columns for each category. A value of 1 indicates the presence of that category. This is preffered for nominal data (where no order exists, like "France", "Germany", "Spain") as it avoids imposing a false ordinal relationship.

Target Encoding: Replaces a category with the mean of the target variable for that category can be powerful but is prone to overfitting.

Choosing the correct encoding methods is vital to prevent the model from learning incorrect relationships from the data.

DATE
PAGE

13. Embedding

Embedding is an advanced technique for representing high-dimensional categorical data (like words or user IDs) as low-dimensional, dense vectors of real numbers. Unlike one-hot encoding which creates sparse, high-dimensional vectors, embeddings are learned by the model and capture semantic meaning.

Key Properties:

Dimensionality Reduction: They represent categories in a much lower-dimensional space.

Semantic Capture: Similar categories have similar vector representations. For example, in word embeddings, the vectors for "king" and "queen" are closer to each other than to the vector for "apple".

Learned Representation: The values of the embedding vector are parameters that are learned during the model training process.

Embeddings are a cornerstone of modern NLP (e.g., word2vec) and recommendation systems, allowing models to understand complex relationships between categorical entities.

DATE
PAGE

14. Dimensionality Reduction

Dimensionality reduction is the process of reducing the number of random variables (features) under consideration by obtaining a set of principle variable. It helps in combating the "curse of dimensionality" which can lead to overfitting and high computational costs.

It serves two main purposes:

Visualization: Projecting high-dimensional data onto 2D or 3D planes for human interpretation.

Feature Extraction: Creating a new, smaller set of features that captures most of the important information from the original data.

The two primary functions are:

Principal Component Analysis (PCA): An unsupervised method that finds directions of maximum variance.

~~Linear Component~~

Linear Discriminant Analysis (LDA): A supervised method that finds directions that maximise the separation between classes.

DATE
PAGE

15. Principal Component Analysis (PCA).

Principal component Analysis (PCA) is an unsupervised, linear dimensionality reduction technique. Its goal is to transform the original features into a new set of uncorrelated features called Principal Components, which are ordered by the amount of variance they capture from the data.

How it works:

Standardize the data (mean = 0, variance = 1). Compute the covariance matrix to understand feature relationships.

Calculate the eigenvectors (principal components) and eigenvalues of this matrix. The eigenvectors define the directions of the new feature space, and the eigenvalues indicate the magnitude of variance along these directions.

Select the top-K eigenvectors with the highest eigenvalues to form the new, lower-dimensional dataset.

PCA is excellent for noise reduction and visualization but does not consider class labels, so it is not always optimal for classification tasks.

DATE
PAGE

16. Linear Discriminant Analysis (LDA).

Linear Discriminant Analysis (LDA) is a supervised dimensionality reduction techniques primarily used for classification tasks. Unlike PCA, which focuses on maximizing variance, LDA finds a feature subspace that maximizes the separation between multiple classes.

Objective:

LDA seeks to project the data onto a lower-dimensional space that achieves:

Maximum between-class separation: The distance b/w the means of different classes should be as large as possible.

Minimum within-class variance: The data points of the same class should be as close together as possible.

This results in a projection where classes are as distinct and compact as possible, which often leads to better classification performance than PCA when the class labels are known. LDA is both a dimensionality reducer and a classifier.

DATE
PAGE

17. Bagging and Boosting:

Bagging and Boosting are two powerful ensemble learning techniques that combine multiple weak learners (e.g., decision trees) to create a single strong learner.

Bagging (Bootstrap Aggregating):

Concept: Trains multiple models in parallel on different random subsets of the training data (sampled with replacement).

Goal: To reduce variance and prevent ~~overfit~~ overfitting.

Combining Prediction: For classification, it uses a majority vote; For regression, it uses an average.

Prime Example: Random Forest, which is an ensemble of decision trees trained with bagging

Boosting:

Concept: Trains models sequentially, where each new model focuses on correcting the errors made by the previous ones.

Goal: To reduce bias and create a strong leader from a sequence of weak ones.

DATE
PAGE

Combining Predictions: Models ~~are~~ are weighted based on their performance; more accurate models have a higher say.

Prime Examples: AdaBoost, Gradiant Boosting Machine (GBM), and XGBoost.