Notes by Izam Mohammed

Feature Engineering

It is the process of creating, selecting, or transforming the input variables (features) used in an ML model to improve its performance.

Aspects of Feature Engineering

- *Creating Features*
- *Selecting features*
- *Transforming features*
- *Handling missing data*
- *Dealing with Outliers*
- *Encoding categorical data*
- *Feature Scaling*
- *Time and date feature engineering*

Techniques used in Feature Engineering

- *Feature selection*
  - *Univariate feature selection*
  - *Recursive feature elimination*
  - *Feature importance from tree-based models*
  - *L1 regularization*
- *Feature Transformation*
  - *Scaling and normalization*
  - *One-hot and label encoding*
  - *Binning and discretization*
  - *Logarithmic, square root, or Box-Cox transformation*
  - *Handling text data (TF-IDF, Word Embeddings)*
- *Feature Creation*
  - *Polynomial features*
  - *Interaction terms*
  - *Aggregation and grouping of data*
- *Handling missing data*
  - *Imputation*
  - *Creating indicator variables for missing data*
- *Outliers*
  - *Identifying outliers through statistical methods*
  - *Winsorizing or clipping outliers*
  - *Transform data to get outliers*
- *Feature Scaling*
  - *Standardization*
  - *Min-max*
  - *Robust*
  - *Scaling with maximum absolute value*
  - *Log transformation*

- *Encoding*
    - *One-Hot*
    - *Label*
    - *Mean Encoding(Target Encoding)*
    - *Binary Encoding*
    - *Frequency Encoding*
- *Text data*
    - *Tokenization and text preprocessing*
    - *TF-IDF (Term Frequency-Inverse Document Frequency)*
    - *Word embeddings (Word2Vec, GloVe, FastText)*
    - *Text sentiment analysis*
    - *N-grams and bag of word representations*
- *Dimensionality Reduction*
    - *PCA*
    - *ICA*
    - *LDA*
    - *t-SNE*
    - *Non-Negative Matrix Factorization (NMF)*
- *AutoML libraries for feature engineering*
    - *Featuretools*
    - *TSFresh*
    - *Featurewiz*
    - *Pycaret*

Feature Selection

It is the process of choosing a subset of the most relevant features (variables or columns) from the dataset

Filter Methods

- *Correlation-based feature selection: Select features that have the strongest correlation with the target*
- *Variance Threshold: Remove features with low variance, as they might not provide much information*
- *Information Gain: Calculates the reduction in entropy from the transformation of a dataset*
- *Chi-square Test: Used for categorical features in a dataset. Calculate between each feature and the target and decide the features*
- *Fisher's Score: Fisher's score is one of the most widely used supervised feature selection methods.*
- *Mean Absolute difference (MAD): computes the absolute difference between the mean value. MAD is like a variance.*

- *Dispersion Ration: Another measure of dispersion applies the arithmetic mean and geometric mean.*

$$AM_i = \overline{X_i} = \frac{1}{n}\sum_{j=1}^{n} X_{ij} \; , \qquad GM_i = \left(\prod_{j=1}^{n} X_{ij}\right)^{\frac{1}{n}} \; ,$$

  RM = AM / GM. This is called the dispersion measure and the higher the value, the higher the relevance to a feature

Wrapper Methods

- *Forward Selection: Start with an empty set of features and iteratively add the most informative feature until a stopping criterion is met.*
- *Backward Elimination: Start with all features and iteratively remove the least informative feature until a stopping criterion is met.*
- *Exhaustive Feature Selection: This is a brute-force evaluation of each feature subset. It tries every combination of the variables and returns the best performance*
- *Recursive Feature Elimination (RFE): Repeatedly fit the model and remove the least important feature in each iteration until the desired number of features is reached.*

Embedded Methods

- *L1 Regularization (Lasso): Features with coefficients that become zero are removed from the list*
- *Tree-based Feature Importance: Decision trees and tree-based models can provide feature importances that help identify the most relevant features.*

Sklearn Things

- *SelectKBest: Can select the top k best features.*
- *SelectPercentile: Can select the features by the score above a certain percentage*

Univariate vs Multivariate selection

- *Univariate considers each feature independently of the others. Statistical tests like ANOVA, chi-square test, and correlation coefficients are used in here.*

- *Multivariate feature selection methods take into account interactions or dependencies between multiple features simultaneously. These methods typically involve more complex techniques, such as recursive feature elimination, and wrapper methods.*

Mutual Information

It is a way to measure how much one random variable tells you about another. If the mutual information between 2 variables is high, it means changes in one variable give a lot of information about the changes in the other.

$$I(X;Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \log \frac{p(x,y)}{p(x)\,p(y)}$$

- ★ *I (X; Y): Mutual information between random variables X and Y*
- ★ *∑∑ : Represents a summation of all possible values of X and Y*
- ★ *p(x, y): The joint probability of X and Y occurring together.*
- ★ *p(x): The probability of X occurring on its own*
- ★ *p(y): probability of Y occurring on its own*
- ★ *log 2: This is the base 2 logarithm*

Feature Transformation

Scaling

Feature scaling is a method used to normalize the range of independent variables or features of data.

Types of Scaling and Transformation:

- ➢ *Min-max scaler:*
  - *Scale data into typically [0, 1]*
  - *For each data point, (X - X_min)/ (X_max - X_min)*
- ➢ *Standard Scaler (Z-score scaling):*
  - *Also known as standardization*
  - *Transform data to have a mean of 0 and std of 1*
  - *For each data point, (X- X_mean) / X_std*
- ➢ *Robust Scaler:*
  - *Robust to outliers*
  - *For each data, (X - X_meadian) / IQR*
- ➢ *Max Absolute Transformation*
  - *Scale the data in such a way that the maximum absolute value becomes 1*
  - *X_scaled = X / X_max_abs*
- ➢ *Power Transformation (Box-Cox)*
  - *Applies to make the data more Gaussian-like*
  - *If lambda == 0, log(data)*
  - *else, (data^lambda - 1) / lambda*
- ➢ *Quantile Transformation*
  - *Maps the data to a uniform or normal distribution using quantiles*
- ➢ *Log transformation*
  - *Useful for data with exponential growth or skewed distributions.*
  - *X_transformed = log(X)*

Encoding

- *Label Encoding: Assign a unique integer to each category*
- *One-Hot encoding: Creates a binary column for each category*
- *Binary Encoding: Coverts categories into binary code*
- *Ordinal Encoding: Assign numerical values to categories based on the predefined order*
- *Frequency Encoding: Replace categories with thor frequency in the dataset*
- *Target encoding: Replace categories with the mean of the target variables in that category*

Binning

- *Equal Width Binning: divides the range of the data into equally sized bins*
- *Equal Frequency Binning (Quantile binning): Divide the data such that each bin contains the same number*
- *K-means clustering binning: Applies the K-means algorithm to group data points into bins*
- *Entropy-based binning: uses entropy to measure the impurity of bins and split the data to minimize entropy within each bin*
- *Custom Binning: Define specific boundaries according to the requirement*

Feature Creation

- *Polynomial features: Take the powers of the existing features, such as squaring or cubing them*
- *Interaction Features: Combine 2 or more existing features. Eg;- divide 2 features*
- *Cross-product features: Take dot product of 2 features*
- *Frequency Features: Count the frequency of occurrence of certain values in the dataset*

Dimensionality reduction

- *PCA (Principle component analysis)*

  Steps

  - *Standardize the data*
  - *Calculate the covariance matrix. Each element (i, j) represents the covariance between feature i and feature j*
  - *Calculate the Eigen vectors and Eigen values of the covariance matrix. These Eigenvalues represent the principal components, and the corresponding eigenvalues represent the variance explained by each component*
  - *Sort the eigenvalues in descending order to identify the principal ones.*
  - *Select the number of principal components.*
  - *Build the projection matrix by selecting the top-k eigenvalues*
  - *Get the dot product of the projected data and the standardized X*
- *LDA (Linear Decrement Analyse)*

  Steps

  - *Calculate the mean vector of each class]*
  - *Calculate 2 scatter matrices:*
    - *within the class: Sum of the covariance matrix of each class*

        - *between class: Sum of outer products of the difference between class mean and the overall mean*
    - *Find the eigenvectors and eigenvalues of the matrix S_w(-1) * S_b, where S_w(-1) is inverse of within-class scatter matrix and S_b is between class scatter matrix*
    - *Sort eigenvalues in descending order*
    - *Select the number of components*
    - *Build the projection matrix by selecting the top-k eigenvalues*
    - *Get the dot product of the projected data and the X*
- *ICA (Independent Component Analysis)*

    Used to separate independent sources from a mixed signal

    Steps
    - *Subtract the mean from each feature to center the data*
    - *Whiten the data using PCA*
    - *Choose the number of Independent Components*
    - *Initialize random mixing matrices (W and A)*
    - *Update the Unmixing Matrix (W): Use any optimization such as gradient descent*
        - *Gradient = (1 - 2 / (1 + exp(-1 * X . B).T)) . X*
        - *W -= learning * Gradient*
    - *Get the dot product of data and Unmixing Matrix (W)*

- *t-SNE (t- Distributed Stochastic Neighbor Embedding)*

    Steps
    - *Get the pairwise squared distances between the data points. Shape will (n x n) where n is the number of data points*
    - *Calculate pairwise similarities*
    - *Calculate conditional probabilities of distances and perplexity*
    - *Initialize the low-dimensional embedding*
    - *Choose a perplexity value, a higher perplexity value considers more global relationships, with a lower focus on local relations.*
    - *Optimize to minimize the divergence between the conditional probabilities of high dimensional space and low dimensional space*
    - *Return the low-dimensional embedding*

- *NMF (Non-Negative Matrix Factorization)*

    Steps
    - *Choose the number of components*
    - *Initialize the matrix W with shape (n, k) and H with shape (k, m) where n is the number of data points, m is the number of features, k is the desired dimension*
    - *Define the number of iterations and learning rate*
    - *Update the W and H iteratively with some optimizers*
    - *Reconstruct data by taking the dot product of W and H*