ML Questions | Sessions 1-5

* What is ML?

ML stands for Machine Learning, a subset of AI that allows computers to learn from data and improve performance with experience.

* Difference btw ML and traditional programming?

ML differs from traditional programming in that it learns patterns from data, while traditional programming relies on explicitly defined rules and instructions.

* Should we feed ML model with desired output?

Yes, for supervised learning, we provide the ML model with input-output pairs to learn the underlying relationship.

* ML types?

ML types include supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning.

* What can we do with ML models?

With ML models, we can perform tasks like classification, regression, clustering, dimensionality reduction, and recommendation.

* What are the main categories of Supervised Learning?

Supervised Learning main categories: regression (continuous output) and classification (discrete output).

* Main categories of Unsupervised Learning?

Unsupervised Learning main categories: clustering (grouping data) and dimensionality reduction (simplifying data).

* Which data types can be used in ML?

Data types used in ML: numerical, categorical, ordinal, and text data.

* ML Lifecycle steps

ML Lifecycle steps: data collection, data preprocessing, feature engineering, model selection, training, evaluation, hyperparameter tuning, and deployment.

* How should we use data to train and build robust models?

To build robust models, use a representative dataset, preprocess data, handle missing data, encode categorical data, normalize/standardize data, and eliminate irrelevant features.

* What if we use entire data in both training and testing?

If you use the entire data for both training and testing, it leads to overfitting, and the model will not generalize well to new data.

* What is the general rule in data splitting? Rates?

The general rule for data splitting: 60-80% for training, 20-40% for testing. Often, 70% training and 30% testing is used.

* Before building ML models, we should analyze the data set and feature set. What can we do?

Analyze the dataset and feature set by visualizing data, calculating descriptive statistics, and checking for correlations, outliers, and missing data.

* What is multicollinearity?

Multicollinearity occurs when independent variables in a regression model are highly correlated, leading to unreliable estimates of coefficients.

* Which ML models are available in Regression family?

Regression family models: Linear Regression, Ridge Regression, Lasso Regression, Elastic Net, Polynomial Regression, and Support Vector Regression.

* What are the assumptions in Linear Regression?

Linear Regression assumptions: linearity, independence of errors, constant variance of errors (homoscedasticity), normally distributed errors, and no multicollinearity.

* Linear Regression Equation

Linear Regression equation: y = b0 + b1\*x1 + b2\*x2 + ... + bn\*xn

* Formula of error

Error formula: error = actual\_output - predicted\_output

* How can we evaluate ML models in regression?

Regression evaluation methods: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared.

* X y variables in machine learning?

X represents independent variables (features or predictors).Y represents dependent variable (target or response).

* b0, b1, b2, b3 in machine learning

b0 represents the y-intercept or bias term. b1, b2, b3 represent coefficients (slopes) of the independent variables.

* There is an individual y-intercept value for each independent variable. True/False?

False, there is only one y-intercept in a linear regression model.

* Coefficients are hyper-parameters in Linear Regression. We should optimize them. T/F?

False, coefficients are learned parameters. Hyperparameters are set before the learning process begins.

* Error metrics in Regression?

Error metrics in Regression: MSE, RMSE, MAE, and R-squared.

* Loss function vs cost function?

Loss function measures the error for a single observation, while cost function measures the average loss over the entire dataset.

* How to find the best fit line in LR?

To find the best fit line in LR, minimize the cost function (e.g., MSE).

* Gradient Descent?

Gradient Descent is an optimization algorithm to find the minimum of the cost function by iteratively updating the coefficients.

* Learning rate? Hyper-parameter?

Learning rate is a hyperparameter controlling the step size in Gradient Descent.

* Bias-variance tradeoff?

Bias-variance tradeoff: balancing model complexity to avoid underfitting (high bias, low variance) or overfitting (low bias, high variance).

* What are the most common problems in ML?

Common ML problems: overfitting, underfitting, data imbalance, and data leakage.

* How to recognize them?

Recognize problems by visualizing data, checking model performance on training/testing sets, and using cross-validation.

* Solution proposals?

Solution proposals: regularization, data augmentation, resampling, feature selection, and hyperparameter tuning.

* How can we increase the complexity of the regression model with polynomial features?

Increase the complexity of a regression model with polynomial features by adding higher-degree terms of independent variables.

* Degree hyper-parameter? How to find the optimal one?

Degree hyperparameter determines the highest order of polynomial features. Find the optimal one using cross-validation or other model selection techniques.

* What if too complex?

If too complex, the model may overfit the training data and perform poorly on new data.

* Regularization techniques in ML?

Regularization techniques: Ridge Regression (L2), Lasso Regression (L1), and Elastic Net (combination of L1 and L2).

* Common element in Ridge-Lasso-Elastic Net?

Common element in Ridge, Lasso, Elastic Net: a penalty term that discourages overfitting by shrinking coefficients.

* How to determine the size of penalty?

Determine the size of the penalty using cross-validation or other model selection techniques.

* What is the optimal value for … ?

Optimal values depend on the problem and data. Use cross-validation or other model selection techniques to find them.

* When can we prefer Ridge? When Lasso?

Prefer Ridge when there are many small/medium-sized effects, and Lasso when there are a few large effects.

* When does Elastic Net behave like Ridge?

Elastic Net behaves like Ridge when the L1 ratio is close to 0.

* Why should we implement feature scaling? Always?

Feature scaling is important because it can help improve the performance and convergence of many machine learning algorithms. It's not always necessary, but it can be beneficial in many cases, especially when features have different scales or units.

* Most common ways? In scikit-learn?

Two common ways to scale features are normalization (scaling to a [0,1] range) and standardization (scaling to a mean of 0 and standard deviation of 1). Scikit-learn provides various scaling methods, including MinMaxScaler and StandardScaler.

* Cross-validation types?

Common types of cross-validation include k-fold cross-validation, stratified k-fold cross-validation, and leave-one-out cross-validation. Scikit-learn provides functions for these and other types of cross-validation.

* Difference between parameter and hyper-parameter?

Parameters are values that are learned by the machine learning algorithm during training, while hyperparameters are values set prior to training that control the behavior of the algorithm. Hyperparameters are typically set by the user and can affect the algorithm's performance.

* How to optimize the hyper-parameters?

Hyperparameters can be optimized through techniques such as grid search, randomized search, and Bayesian optimization. These methods involve trying different combinations of hyperparameters and evaluating their performance on a validation set.

* How can we implement Grid Search in scikit-learn?

Scikit-learn provides a GridSearchCV class that allows users to perform grid search over a specified hyperparameter space. This class takes a model, a hyperparameter grid, and a scoring metric as inputs and returns the best hyperparameters based on the validation scores.

* What else after finding the best model?

After finding the best model, it's important to evaluate its performance on a test set to ensure that it generalizes well to new, unseen data. It's also important to interpret the model's predictions and understand the factors that are driving its decisions.

* How to improve performance in general?

There are various ways to improve the performance of a machine learning model, including collecting more data, engineering better features, trying different algorithms, optimizing hyperparameters, and ensembling multiple models. It's also important to understand the limitations of the data and the problem being solved.