Feature Engineering Assignment

1. What is a parameter?

A parameter in machine learning is a model-internal value learned from training data (e.g., weights and biases in neural networks) that determines the mapping from inputs to outputs.

2. What is correlation?

Correlation is a statistical measure that quantifies the strength and direction of a linear relationship between two numeric variables.

What does negative correlation mean?

It indicates an inverse relationship between two variables.; values close to -1 indicate a strong inverse linear relationship.

3. Define Machine Learning. What are the main components in Machine Learning?

Machine Learning is a field where algorithms learn patterns from data to make predictions or decisions. Main components: dataset (inputs & labels), features, model/algorithm, training procedure, evaluation metrics, and deployment.

4. How does loss value help in determining whether the model is good or not?

Loss quantifies prediction error; lower loss during validation usually indicates better model performance. Compare training and validation loss to detect overfitting or underfitting.

5. What are continuous and categorical variables?

Continuous variables take numeric values on a continuum (e.g., height, temperature). Categorical variables take discrete labels or categories (e.g., color, class).

6. How do we handle categorical variables in Machine Learning? What are the common techniques?

Common techniques: One-Hot Encoding for nominal categories, Label/Integer Encoding for ordinal categories, Target/Frequency Encoding, and embedding layers for high-cardinality categories.

7. What do you mean by training and testing a dataset?

Training uses a portion of data to fit the model (learn parameters). Testing (or test set) is held-out data used only to evaluate final model generalization.

8. What is sklearn.preprocessing?

A scikit-learn module offering utilities for data preprocessing: scalers, normalizers, encoders, imputers, and transformers used before modeling.

9. What is a Test set?

A test set is a dataset subset not used during training or validation, reserved to provide an unbiased estimate of final model performance.

10. How do we split data for model fitting (training and testing) in Python?

Use sklearn.model_selection.train_test_split to randomly split data into training and testing sets; optionally use StratifiedShuffleSplit for imbalanced class distributions.

How do you approach a Machine Learning problem?

Typical approach: define the problem and metric, gather and inspect data, perform EDA, preprocess and feature-engineer data, select models, train with cross-validation, tune hyperparameters, evaluate on test set, and deploy.

11. Why do we have to perform EDA before fitting a model to the data?

EDA reveals distributions, missing values, outliers, and relationships between variables; it guides cleaning, feature engineering, and model selection to improve results.

12. What is correlation?

Correlation is a statistical measure that quantifies the strength and direction of a linear relationship between two numeric variables.

13. What does negative correlation mean?

It indicates an inverse relationship between two variables.; values close to -1 indicate a strong inverse linear relationship.

14. How can you find correlation between variables in Python?

Use pandas.DataFrame.corr() for pairwise correlations, numpy.corrcoef(), or visual tools like seaborn.heatmap for correlation matrices.

15. What is causation? Explain the difference between correlation and causation with an example.

Causation means one variable directly influences another. Correlation only indicates an association. Example: ice cream sales and drowning incidents correlate (both rise in summer) but ice cream sales do not cause drownings.

16. What is an Optimizer? What are different types of optimizers? Explain each with an example.

An optimizer updates model parameters to minimize loss. Examples: SGD (stochastic gradient descent) — simple gradient updates; Momentum — accelerates SGD by accumulating gradients; Adam — adaptive learning rates combining momentum and RMSProp; RMSProp — scales updates by running average of squared gradients.

17. What is sklearn.linear_model ?

A scikit-learn module that contains linear model implementations like LinearRegression, Ridge, Lasso, LogisticRegression and others for regression and classification tasks.

18. What does model.fit() do? What arguments must be given?

model.fit(X, y) trains the estimator using feature matrix X and target y, updating internal parameters. Some models accept additional args like sample_weight or epochs for iterative estimators.

19. What does model.predict() do? What arguments must be given?

model.predict(X_new) returns predicted labels or values for new input features X_new. For probabilistic outputs use predict_proba when available.

20. What are continuous and categorical variables?

Continuous variables take numeric values on a continuum (e.g., height, temperature). Categorical variables take discrete labels or categories (e.g., color, class).

21. What is feature scaling? How does it help in Machine Learning?

Feature scaling (normalization or standardization) rescales numeric features to similar ranges; it helps gradient-based optimizers converge faster and prevents distance-based models from being biased by feature scales.

22. How do we perform scaling in Python?

Use sklearn.preprocessing.StandardScaler to standardize (zero mean, unit variance) or MinMaxScaler to scale to a given range; apply fit on training set and transform on train/test.

23. What is sklearn.preprocessing?

A scikit-learn module offering utilities for data preprocessing: scalers, normalizers, encoders, imputers, and transformers used before modeling.

24. How do we split data for model fitting (training and testing) in Python?

Use sklearn.model_selection.train_test_split to randomly split data into training and testing sets; optionally use StratifiedShuffleSplit for imbalanced class distributions.

25. Explain data encoding?

Data encoding converts categorical/text features into numeric representations suitable for models: label encoding, one-hot encoding, binary encoding, target encoding, or learned embeddings for deep models.