End-To-End
Machine Learning Pipeline

Pipeline Steps:

1. Frame the problem:

Before starting, clearly define what you want to solve. Understanding the goal helps guide all your decisions, from choosing the data to selecting the right model. Knowing what counts as success keeps your work focused.

2. Select a Performance Matrix:

Decide how you will measure your model's success. Pick a metric that makes sense for your problem, like accuracy for classification tasks or RMSE for predicting continuous values. This ensures you are optimizing for the right outcome.

3. Check the Assumption:

Before you start, make sure the assumptions you're making about the problem are correct. For example, if you want to predict house prices, you might assume it's a regression problem. But later, you could realize it makes more sense to classify houses as expensive, average, or cheap. Checking your assumptions early helps avoid choosing the wrong type of model and prevents wasted effort.

4. Get the Data:

Collect the datasets you need to solve the problem. Without relevant and sufficient data, your model cannot learn meaningful patterns or make accurate predictions.

5. Create the Workspace:

Set up your coding environment, folders, and tools. An organized workspace makes it easier to manage data, code, and results efficiently throughout the project.

6. Download the Data:

Bring the dataset into your workspace so you can start exploring and analyzing it. Make sure the files are readable and correctly formatted.

7. Take a Quick look at the data structure:

Inspect the columns, types, and any missing values. This initial check helps you understand the dataset and plan your cleaning and preprocessing steps.

8. Create a Test Set:

Set aside a portion of the data, usually 20%, for testing later. This reserved data acts as unseen examples to evaluate how well your model generalizes.

9. Discover & Visualize the Data to Gain Insights:

Explore patterns and trends using charts, plots, and summaries. Visualization helps you understand relationships, detect anomalies, and generate ideas for features.

10. Looking for Correlations:

Check which features are related to each other and to the target. Strong correlations can guide feature selection and help focus on the most informative variables.

11. Experimenting with Attribute Combinations:

Try combining existing features or creating new ones. These combinations sometimes reveal hidden patterns and improve your model's predictive performance.

12. Prepare the Data for ML Algorithms:

Clean missing or messy data, handle text and categorical features, and scale numerical values. Proper preprocessing ensures that your model can learn effectively and fairly.

13. Select & Train a Model:

Split your data into training and validation sets and use cross-validation to check performance. Train your chosen model on the training data while tuning based on validation results.

14. Fine Tune your Model:

Adjust hyperparameters using grid search or random search. You can also try ensemble methods, which combine multiple models to improve accuracy and stability.

15. Evaluate System on Test Set:

Test the final model on the reserved test set to see how it performs on completely unseen data. This gives a realistic measure of how it will work in practice.

16. Launch and Evaluate:

Deploy the model if needed and monitor its performance. Real-world data may change over time, so continuous evaluation ensures your model remains useful and accurate.

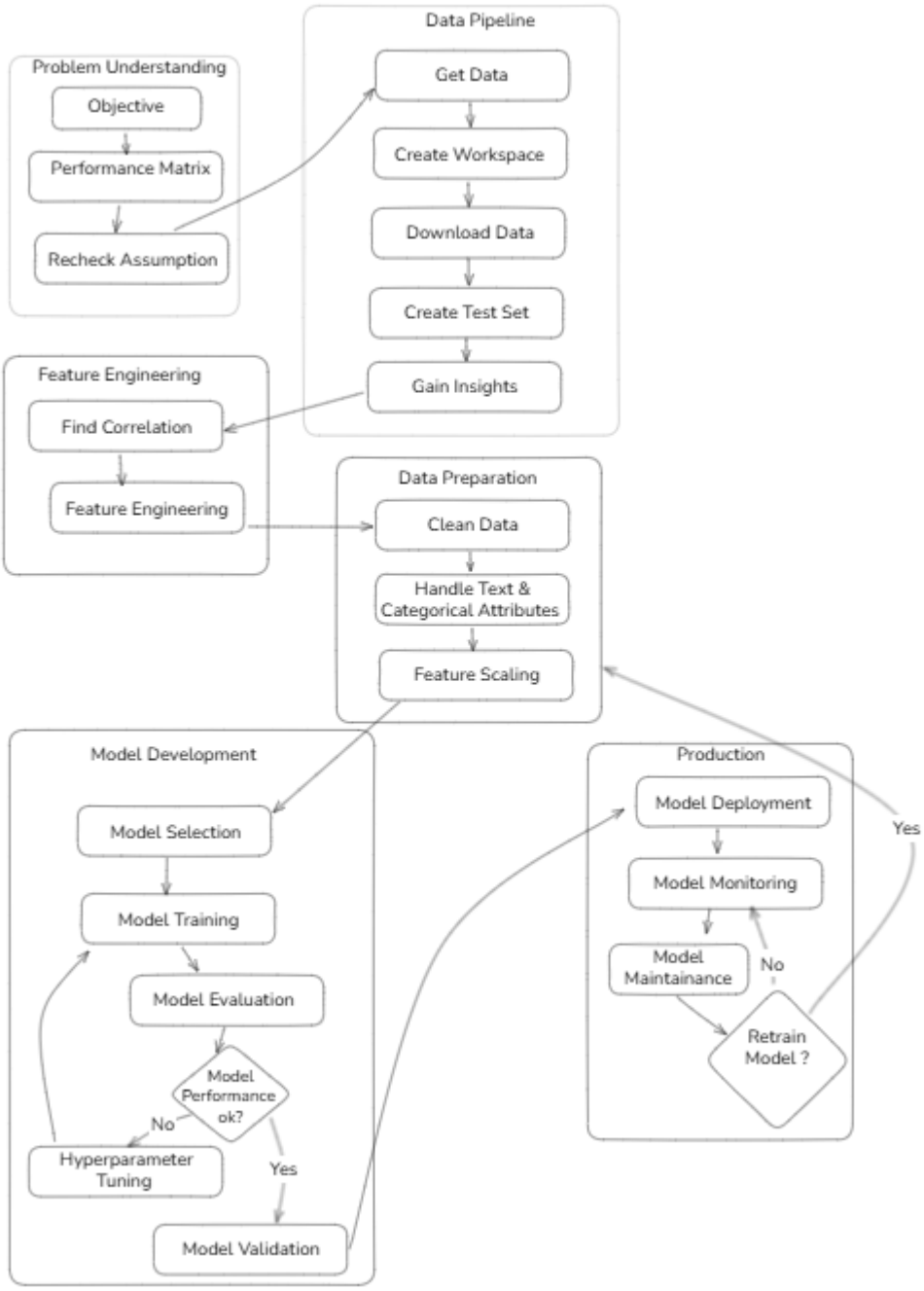

Figure: Mermaid Diagram of End-to-End Machine Learning Pipeline