

## Google Developers: Machine Learning Course

### Introduction

Focuses on *Machine Learning Problem Framing* which helps the user to decide what is the proper ML model a user should implement to solve a certain problem.

### Problem Framing

Process of analyzing the problem to isolate individual elements (factors of the problem) that is needed to be addressed to solve it.

Helps the user to determine the project's feasibility and provides a clear set of goals.

Basically, problem framing will help the user to determine what ML solution to take and framing the project whether it can succeed or not.

### Understanding the Problem

To understand the problem the user should:

- What is the goal of the project?
- Can it be solved using ML?
- Do you have the required and proper data?

### Stating the Goal of the Project

The user can begin by stating the problem and solution of the problem without using ML terms (Example: Weather App – Predict calculate precipitation in six-hour increments for geographic region)

### Clear use case for ML

Do not implement ML for everything, sometimes a problem can be solved using simpler solutions.

To confirm whether a problem should use ML, the user can implement an easier and optimized approach first. The yielded performance of this approach can be used to benchmark your future models using ML.

In comparing ML model to a non-ML approach, the user should take note of the following:

- Quality
- Cost and Maintenance
  - o Can ML justify the cost?
  - o Does product need to have people with ML expertise?
  - o How much maintenance?

### Data

In order to build good models, the user must have good data that contain features that can be used to predict a certain outcome.

In selecting a data, the user should take note of:

- Abundancy
- Consistency and Reliability
- Trustworthiness
- Availability
- Correctness
- Representability

### Predictive Power

To make prediction, the features in the dataset should have correlated feature, the more likely a dataset have correlation the more likely it can predict.

Determining which features have predictive power can be a time-consuming process. You can manually explore a feature's predictive power by removing and adding it while training a model. You can automate finding a feature's predictive power by using algorithms such as **Pearson correlation**, **Adjusted Mutual Information (AMI)**, and **Shapley value**, which provide a numerical assessment for analyzing the predictive power of a feature.

### Predictions vs. Actions

Prediction is the outcome of the model and action is the outcome of the prediction when it is implemented to solve a certain problem.

## Pearson's Correlation Coefficient

A test statistic that can measure statistical relationship or association between two continuous variables.

Known to use in measuring association between liked variables due to its covariance method. This can give information about the magnitude of association or correlation, as well as direction of relationship.

Sample questions:

- Do test scores and hours spent studying have a statistically significant relationship?
- Is there a statistical association between IQ scores and depression?

Assumption:

- Independent of case: should be independent to each other
- Linear relationship: two variables should be linearly related
- Homoscedasticity: the residuals scatterplot should be roughly rectangular-shaped.

Properties:

- Limit: Coefficient should be from  $\pm 1$ , where  $+1$  is perfect positive and  $-1$  is perfect negative, and  $0$  is no relationship
- Pure Number: