Stage 1. Machine Learning - when and why?

Explain the background (real-life scenario) of your ML application.

Project “Personalized Recipe Recommender”

With the huge amount of recipe websites and cooking apps, people often find it difficult to find recipes that suit their dietary preferences, restrictions, and ingredient availability. This leads to inefficient meal planning and dissatisfaction with the culinary experience. This machine learning application aims to develop a personalized recipe recommendation system to solve this problem.

Justify the need for ML approach for your problem.

Justification for ML Approach: Machine learning is crucial in creating a personalized recipe recommender system. Classical keyword-based search approaches do not consider individual preferences, resulting in bad recipe suggestions. ML enables the system to learn and adapt over time, providing users with increasingly relevant and tailored recipe options.

Explain who are stakeholders* (company owners, investors, ML engineers, developers, users) of your project/company/application and what are their goals.

The primary stakeholders are the users of the application or platform. Their goal is to use the application to meet their specific needs and enhance their cooking experience.

Stage 2. ML problem formulation – DATA

Explain the source of the dataset. How did you collect (sample) data (you can make up some possible scenario for your type of data)? Is your sampling biased (e.g. due to non-probability sampling)? Discuss class (im)balance if applicable.

The dataset is collected from a recipe platform or app where users can search and save recipes. The data is obtained through user interactions with the platform, examples being: search queries, saved recipes, ratings, and feedback. The platform also gathers demographic information and dietary preferences. There may be some sampling bias due to non-probability sampling, as the dataset is collected from users who voluntarily interact with the recipe platform. Users who actively engage with the platform may have different preferences or behavior compared to those who do not use the platform. In this case, class imbalance may not be applicable as the problem is focused on recommending recipes rather than predicting a binary or multi-class outcome.

Clearly explain the data points, features and labels of this ML problem. Indicate type of data (continuous variable, categorical or ordinal values etc.) and units of measurement when applicable.

Each data point represents a recipe. It consists of various features that describe the recipe and user preferences. Features include attributes like ingredients, cooking methods, preparation time, difficulty level, cuisine type, dietary restrictions, user ratings, user feedback, and user demographic information. These features can be both categorical (cuisine type) and continuous (preparation time). The label in this ML problem is the recommendation or relevance score assigned to each recipe for a particular user. The recommendation score can be a continuous variable or a categorical value indicating the level of relevance or match between the user's preferences and the recipe.

Explain your feature selection process (no theoretical justification needed)

Feature Selection Process: The feature selection process can involve various techniques like domain knowledge, correlation analysis, and feature importance ranking. The goal is to select features that are most relevant to the recipe recommendation task. For example, features related to user preferences, dietary restrictions, and user ratings may carry more weight in determining the relevance of a recipe for a specific user.

Do you need to continuously collect (update) data, or do you use static dataset?

The data collection process can be a combination of continuous data collection and using a static dataset. Initially, a static dataset can be used to train the ML model. However, as users interact with the application, their actions and feedback can be continuously collected and used to update the dataset. This allows the ML model to adapt and improve its recommendations over time based on the evolving user preferences and feedback.

Stage 3. ML problem formulation – Model and Loss

State the number of datapoints, briefly describe the dataset and/or any data preprocessing or cleaning needed.

The dataset contains information about 10,000 recipes, including features such as ingredients, cooking methods, preparation time, difficulty level, cuisine type, dietary restrictions, user ratings, user feedback, and user demographic information. Preprocessing steps may include handling missing values, encoding categorical variables (e.g., one-hot encoding cuisine types), normalizing numerical features (e.g., scaling preparation time), and removing outliers if applicable.

If using categorical or ordinal variables, explain how you encode them.

Categorical variables such as cuisine type can be encoded using one-hot encoding, where each cuisine category becomes a binary feature. For example, if there are five cuisine types (for example Italian, Mexican, Chinese, Indian, and French), each recipe will have five binary variables representing the presence or absence of each cuisine type.

Describe and explain (why?) your choice of ML model(s)/hypothesis space(s)*, e.g., linear predictors, etc.**

A suitable choice for this recipe recommendation task could be a collaborative filtering approach, leveraging user-item interactions to make personalized recommendations. Collaborative filtering models, such as matrix factorization or deep learning-based models (neural networks), can capture the underlying patterns in user preferences and suggest recipes accordingly.

Another hypothesis space that can be used is content-based filtering. Content-based models focus on the inherent characteristics of the recipes themselves, such as ingredients, cooking methods, cuisine types, and dietary restrictions. These models analyze the content of the recipes and compare it to user preferences to make recommendations.

Describe and explain (why?) your choice of loss function(s), e.g., logistic loss.

The choice of loss function depends on the specific ML model used. For collaborative filtering models, common choices for loss functions include mean squared error (MSE) or binary cross-entropy loss, depending on the type of recommendation task (for ex rating prediction or binary relevance prediction).

Explain the process of model validation - how did you split the data into the training, validation

and test sets. What are the sizes of each set and why did you make such design choice.

The dataset can be split into training, validation, and test sets. A common split could be 70% for training (7,000 recipes), 15% for validation (1,500 recipes), and 15% for testing (1,500 recipes). The training set is used to train the model, the validation set is used for hyperparameter tuning and model selection, and the test set is used to evaluate the final model's performance on unseen data. The random splitting ensures a representative distribution of recipes and user interactions across the sets.

If applicable, describe and explain your use of metrics (e.g. accuracy) in addition to loss function (e.g. logistic loss)

In addition to the loss function, evaluation metrics such as accuracy, precision, recall, or mean average precision (MAP) can be used to measure the performance of the recommendation model. These metrics provide insights into how well the model is able to recommend relevant recipes to users. The specific choice of metrics depends on the project's goals and the nature of the recommendation task (e.g., top-k recommendation or rating prediction).