## Documentation: Clear Explanation of Approach, Decisions, and Results

### 1. Project Objective

The primary goal of this project was to develop a robust and efficient **Intent Classifier** for handling user utterances in a conversational AI context. The model's task is to categorize a raw text utterance into one of five predefined intents: **email\_send**, **calendar\_schedule**, **web\_search**, **knowledge\_query**, and **general\_chat**. The process of the dataset creation involved systematically generating 200 distinct, human-like utterances for each of the five target intents: email\_send, calendar\_schedule, web\_search, knowledge\_query, and general\_chat. For each intent, I focused on creating diverse examples that cover the full range of user requests, phrasing, and complexity associated with that specific function. The resulting dataset comprises 1000 labeled examples, with indices ranging from 1 to 1000, ensuring a balanced distribution of 200 examples per intent to train a robust model

### 2. Data and Preprocessing Decisions

| Feature | Decision/Parameter | Justification/Result |
| --- | --- | --- |
| **Dataset Size** | 1,000 Utterances (200 per intent) | The dataset is **perfectly balanced**, ensuring the model is not biased towards any single intent during training. |
| **Data Split** | **80% Training (800)**, **10% Validation (100)**, **10% Test (100)** | A separate validation set (10%) was created for initial checks, and a held-out test set (10%) was reserved for final, unbiased performance reporting. Stratified splitting was used to maintain class balance across all subsets. |
| **Text Cleaning** | Lowercasing, removal of punctuation, and removal of numbers | This standardization process reduces the vocabulary size and treats variations (e.g., "Email." vs "email") as the same feature, improving model generalization. |
| **Label Encoding** | sklearn.preprocessing.LabelEncoder | Converts the categorical intent strings into numerical targets required for machine learning algorithms. |

### 3. Modeling and Training Approach

#### **Model Architecture**

The solution uses a machine learning Pipeline to chain the feature engineering step with the classification algorithm, ensuring consistency and preventing data leakage between the steps:

Raw Utterance->TfidfVectorizer->​Numerical Features->LogisticRegression->​Intent Prediction

#### **Feature Engineering**

* **Method:** **TF-IDF (Term Frequency-Inverse Document Frequency)** Vectorization.
* **Initial Parameters:** min\_df=5, ngram\_range=(1, 2), and stop\_words='english'.

#### **Hyperparameter Tuning**

* **Method:** **GridSearchCV** with **5-fold cross-validation (cv=5)** was executed on the 80% training set.
* **Objective:** Maximize cross-validation accuracy.
* **Tuning Grid:** Tested various combinations of TF-IDF feature types (unigrams, bigrams, trigrams), minimum document frequency (min\_df), and Logistic Regression regularization strength (C) and solver.

#### **Final Model Parameters and Results**

The tuning process identified the optimal model configuration:

| Component | Parameter | Best Value |
| --- | --- | --- |
| **TfidfVectorizer** | ngram\_range | **(1, 1)** (Unigrams only) |
|  | min\_df | **1** |
| **LogisticRegression** | C | **10** |
|  | solver | **liblinear** |
| **Cross-Validation Result** | Best Mean Accuracy | **0.8600** |