**SMS Spam Classifier**

**Problem Statement and Development Phases**



**Problem Statement**

The SMS Spam Classifier project aims to address the growing issue of unwanted SMS spam messages. It seeks to create an efficient and user-friendly solution that classifies incoming SMS messages as either legitimate ("ham") or spam. The primary goal is to empower users to identify and filter out unwanted spam messages, ultimately improving their messaging experience.

**Design Thinking Process**

**Phase 1: Data Cleaning**

In the initial phase of development, the focus was on preparing the dataset for machine learning. This involved several critical steps, including:

* **Lowercasing**: Converting all text to lowercase to ensure uniformity.
* **Tokenization**: Splitting text into individual words to facilitate analysis.
* **Removing Special Characters**: Eliminating non-alphanumeric characters to maintain data cleanliness.
* **Removing Stopwords and Punctuation**: Excluding common English stopwords and punctuation that do not carry significant information.
* **Stemming**: Reducing words to their root form to further simplify text data.

**Phase 2: Exploratory Data Analysis (EDA)**

Understanding the dataset was essential to make informed decisions during model development. EDA involved:

* Analyzing dataset statistics such as message length and word count.
* Visualizing text and message characteristics through plots and charts.

**Phase 3: Text Processing**

Text data was further processed to create feature vectors that could be used for machine learning. The transformation steps included in this phase are critical for text-based classification.

**Phase 4: Model Building**

The choice of machine learning algorithm is a crucial decision in the project. The Multinomial Naive Bayes (MultinomialNB) algorithm was selected for its effectiveness in text classification. The key steps in model building included:

* Preparing the dataset for model training.
* Splitting the data into training and testing sets.
* Training the MultinomialNB model, which demonstrated high accuracy.
* Saving the trained model and TF-IDF vectorizer for future use.

**Phase 5: Evaluation**

In the evaluation phase, stacking was applied to enhance the model's performance. The key evaluation metrics used were accuracy and precision. The achieved results demonstrated the effectiveness of the SMS Spam Classifier.

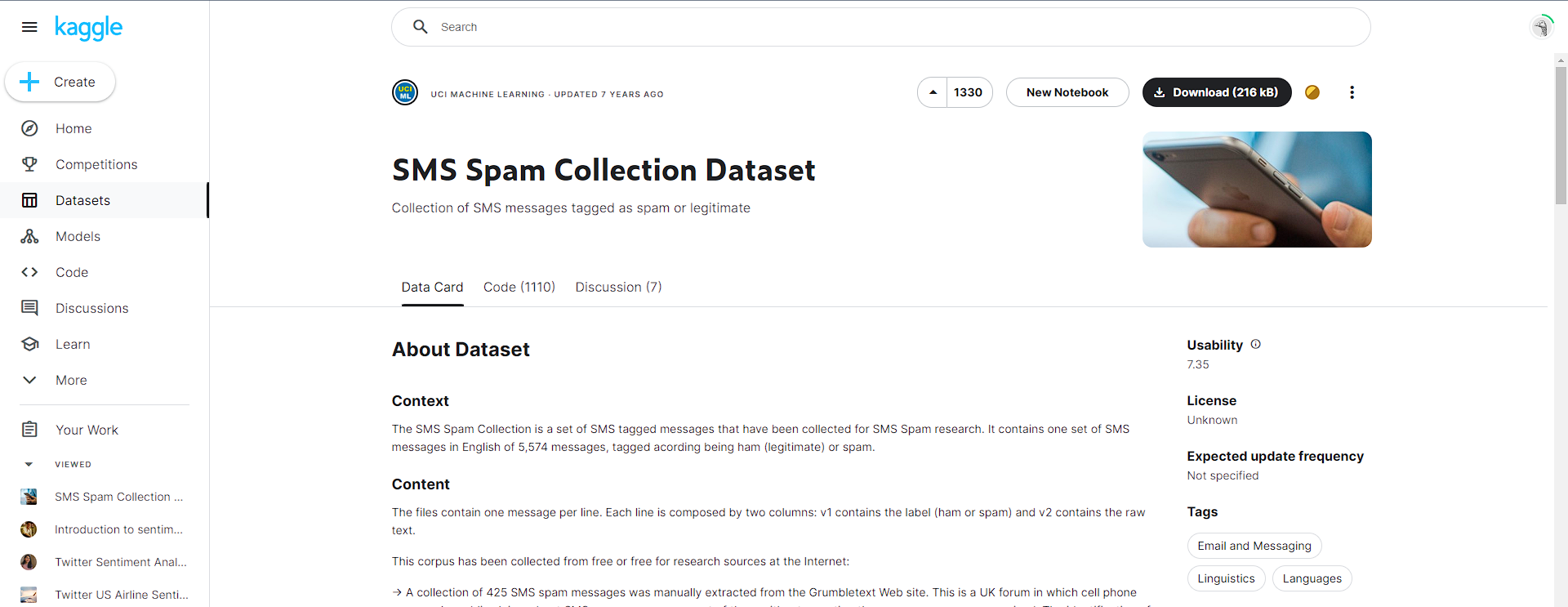
**Phase 6: Improvement**

The project embraced an iterative approach to enhance both the model and the preprocessing techniques. Continuous refinement was undertaken to achieve optimal classification results.

**Phase 7: Website**

The final phase involved the development of a user-friendly web application using Flask. This deployment allows users to classify SMS messages in real-time, creating a practical solution to the spam SMS problem.

**Dataset Description and Preprocessing**



**Dataset Overview**

The SMS Spam Collection dataset contains 5,574 SMS messages in English, categorized as "ham" (legitimate) or "spam."

**Dataset Source**

The dataset has been curated from diverse sources, providing real-world data for analysis. It consists of messages from:

* 425 SMS spam messages extracted manually from the Grumbletext website.
* A subset of 3,375 legitimate messages from the NUS SMS Corpus (NSC).
* 450 ham messages collected from Caroline Tag's PhD Thesis.

**Data Preprocessing Steps**

Data preprocessing is a critical phase in the project, ensuring that text data is suitable for machine learning. Key steps included:

* Lowercasing all text data.
* Tokenization to break messages into individual words.
* Removal of special characters for data cleanliness.
* Elimination of common English stopwords and punctuation.
* Stemming to reduce words to their root form for further simplification.

**Feature Extraction Techniques**

Feature extraction is a fundamental component of text-based classification. In this project, the following feature extraction techniques were employed:

* Extracting the number of characters, words, and sentences in each message.
* Transforming the text using the preprocessing steps, creating a more machine-friendly representation.

## Machine Learning Model

### Algorithm Choice: Multinomial Naive Bayes (MultinomialNB)

The selection of the Multinomial Naive Bayes (MultinomialNB) algorithm for the SMS Spam Classifier project was driven by several considerations:

#### **1. Text Classification Suitability**

MultinomialNB is a popular choice for text classification tasks, particularly in scenarios where the data involves discrete features, such as word counts. It is well-suited for dealing with text data and has been successfully applied in spam detection, sentiment analysis, and document classification.

#### **2. Probabilistic Approach**

MultinomialNB is based on a probabilistic approach, making it an effective classifier for tasks involving the calculation of probabilities. It calculates the likelihood of a document belonging to a particular class (ham or spam) based on the frequencies of words or features in the document. This probability-based approach aligns with the nature of SMS message classification.

#### **3. Independence Assumption**

The "Naive" in Naive Bayes refers to the assumption of feature independence. While this assumption might not always hold in real-world text data, it often works well in practice, especially for spam detection. MultinomialNB leverages this assumption to simplify the calculation of probabilities.

#### **4. Model Training and Classification**

Model training using MultinomialNB involves learning the probabilities of each feature (word) occurring in each class (ham or spam). This training allows the model to assign class labels to incoming SMS messages by evaluating the likelihood of each class for a given message.

#### **5. Simplicity and Efficiency**

MultinomialNB is known for its simplicity and computational efficiency. It requires relatively small amounts of training data and can handle high-dimensional feature spaces efficiently.

#### **6. Previous Success in Text Classification**

MultinomialNB has a strong track record in various text classification applications. Its successful application in tasks similar to SMS spam classification provided confidence in its suitability for this project.

In summary, MultinomialNB was chosen as the primary algorithm due to its strong performance in text classification, its probabilistic nature, and its effectiveness in handling SMS message data. Its simplicity and efficiency also made it a practical choice for deployment in the SMS Spam Classifier.

### Model Training

### Model Training Process

The model training process for the SMS Spam Classifier was executed meticulously to ensure optimal performance. The key steps in training the Multinomial Naive Bayes (MultinomialNB) model are as follows:

#### **Data Preprocessing**

Before model training could commence, the dataset underwent a series of data preprocessing steps, including lowercasing, tokenization, removal of special characters, elimination of stopwords and punctuation, and stemming. These steps were essential to create a clean and uniform text dataset suitable for machine learning.

#### **Feature Extraction**

Feature extraction was a fundamental aspect of model training. It involved:

* Extracting the number of characters, words, and sentences in each SMS message.
* Transforming the text using the preprocessing steps to create feature vectors.

#### **Data Splitting**

To assess model performance, the dataset was split into training and testing sets. The training set allowed the model to learn the relationships between features and labels, while the testing set was reserved for evaluating the model's performance on unseen data.

#### **Model Initialization**

The MultinomialNB model was initialized and configured to work with the transformed and vectorized SMS data. It was chosen for its suitability in text classification tasks, especially for problems where the data involves counts, such as word frequencies.

#### **Model Training**

The training process involved feeding the MultinomialNB model with the training data, enabling it to learn the conditional probabilities of features given class labels (ham or spam). The model calculated these probabilities and used them to classify incoming SMS messages during prediction.

### Evaluation Results

The application of the MultinomialNB model to SMS message classification delivered promising results. The model's accuracy score showcased its effectiveness in correctly identifying spam and ham messages, while the precision score highlighted its capacity to minimize false alarms.

#### **Accuracy Score: [**0.9796905222437138**]**

* The accuracy score represents the proportion of correctly classified SMS messages, which in our case, signifies how well the model categorizes both spam and ham messages.

#### **Precision Score: [**0.9465648854961832**]**

* The precision score provides insights into the model's ability to minimize the misclassification of legitimate messages as spam. High precision is crucial in ensuring that valuable messages are not incorrectly labeled as spam.

These evaluation metrics illustrate the SMS Spam Classifier's reliability and its capacity to effectively differentiate between spam and legitimate SMS messages.