Introduction

A consumer complaint is an expression of dissatisfaction made by a customer regarding a product or service they have purchased or used. This could be over a wider range of issues such as product defects, poor quality of service or bad consumer experience, which is usually directed toward a business, government bodies etc. with an aim to seek resolution or fix such inconsistencies caused by the product or service provider.

Consumer Complaints are usually submitted through various channels such as forms, emails, telephones, in-person or via written correspondence. With a widespread of online platforms and social media, conumers now have more avenues and channels to voice their complaints and concerns regarding products and services.

Consumer Complaints Classification is an important process taken by companies and bodies which involves categorizing and analyzing consumer complaints using Machine Learning and Artificial Intelligence (AI) to gain valuable insights into recurring issues and identify trends which will help them take measures to improve their products and services.

In this technical documentation, we will look into the various stages of the classification pipeline, including data collection, preprocessing, feature extraction, model training, evaluation, and deployment.

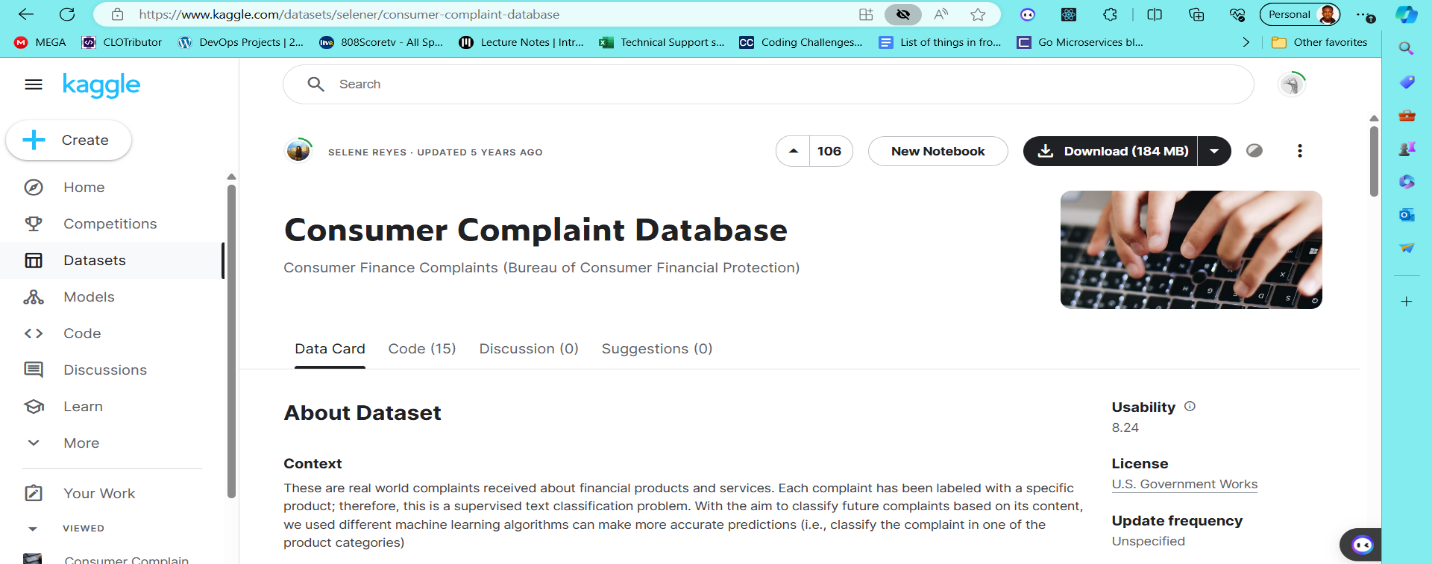
Methodology

The consumer complaint classification requires high-quality, relevant data to effectively train and evaluate the classification models. In this section, we will discuss the process of data collection, including the sources of the dataset and steps taken to preprocess the data.

1. Source of Dataset:

- The dataset used for training and testing the consumer complaint classification system was obtained from Kaggle, a popular platform for data science competitions and datasets.

- Each complaint entry in the dataset consists of textual descriptions of the consumer's grievances, along with metadata such as 'Date received', 'Product', 'Sub-product', 'Issue', 'Sub-issue', 'Consumer complaint narrative'.



2. Data Preprocessing:

- Before training our model on the dataset, preprocessing steps were carried out to clean and prepare the raw complaint data.

- Data preprocessing techniques included:

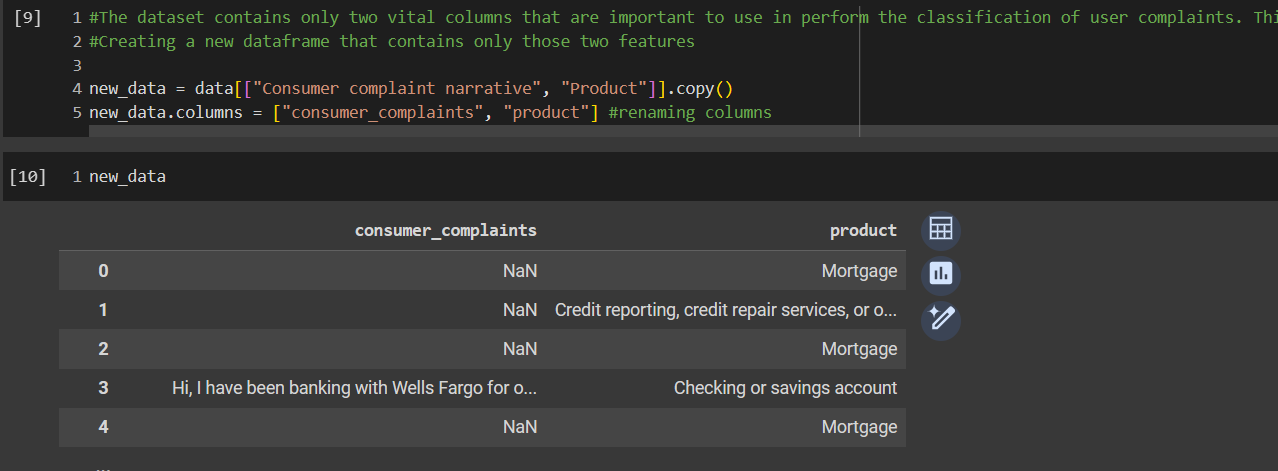
- Removal of empty entries: Complaints entries that are missing were identified and removed to ensure the integrity of the dataset.

A screenshot of a computer

Description automatically generated

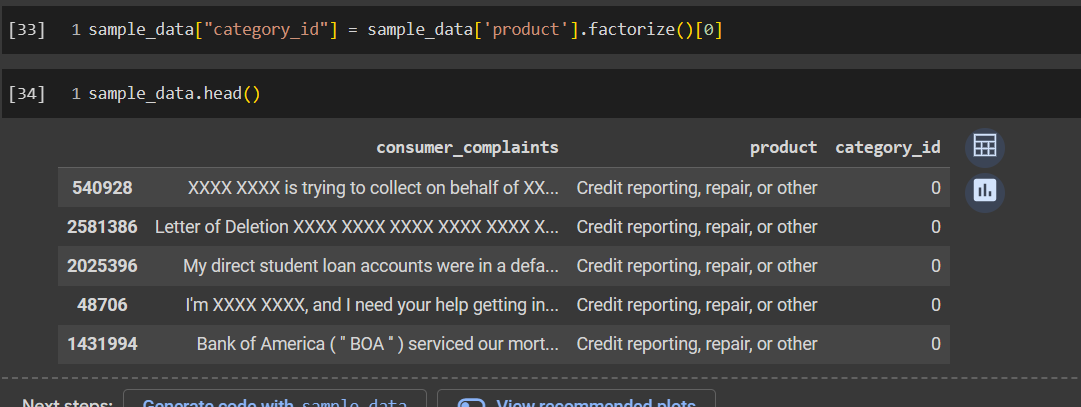
- Feature Extraction:

The dataset contained two vital features to perform classification on, the customer complaint narrative and the product(the classified complaint). By extracting only important features, we reduce the dimensionality of the dataset by transforming raw input data into a more compact and informative representation as this ensures that the complexity is decreased, making the training process more efficient.



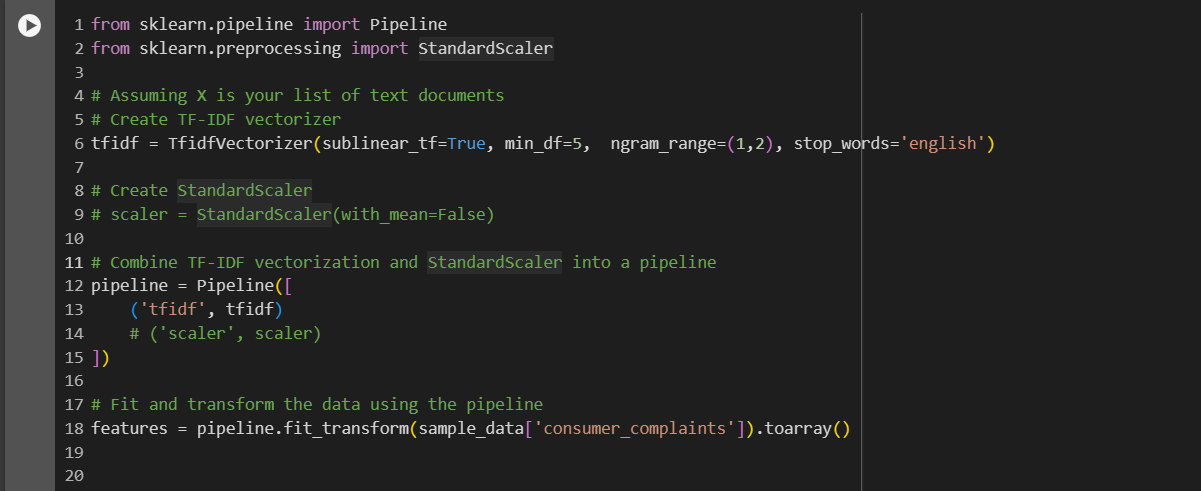
- Factorization:

Carried out the factorization of the product feature which helped to assign numerical values to unique entries of the product feature. This is helpful later to identify which complaint was predicted after the model has made its prediction.



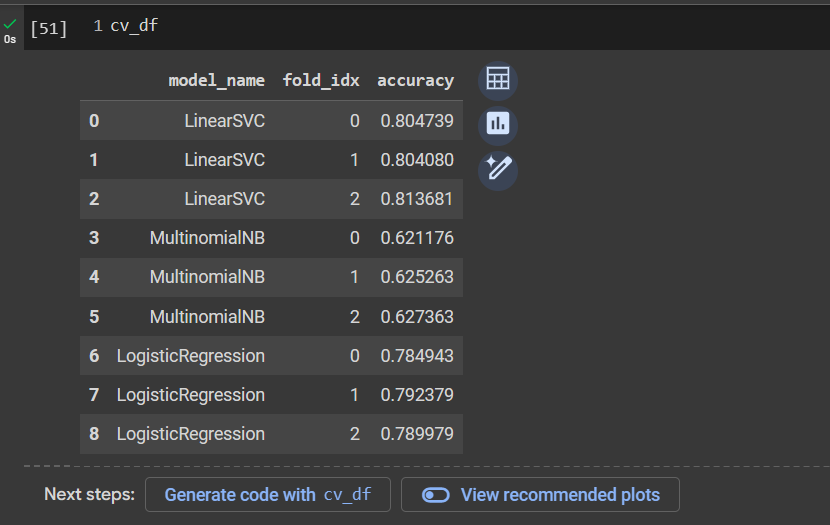
- Text Preprocessing with TF-IDF:

Finally in order to prepare the data for training by the model , performed TF-IDF vectorization which converts the textual data into numerical data. TF-IDF vectorization is a fundamental technique in NLP for converting textual data into numerical features that capture the semantic content and importance of words within documents. It enables machine learning models to process and analyze text effectively, facilitating tasks such as text classification, information retrieval, and document clustering.



3. Model Selection and Training:

- To perform training on the vectorized dataset, I choose 3 classification models and trained the dataset on it looking closely at training and test accuracies. The accuracy of a model is an important factor to consider in a classification model as the higher the value the better the algorithm is.



* The LinearSVC model performed best and so it was used to train the model and dumped into a pickle file so as to used in a deployed application.

4. Model Deployment

* Loaded the trained classification model and any necessary preprocessing components, such as TF-IDF vectorizers, using joblib. The model and vectorizer are loaded into memory when the Flask application starts, ensuring they are readily available for making predictions.
* Created HTML templates for the user interface, including an index.html file for the input form and a result.html file for displaying prediction results. CSS stylesheets were used to enhance the visual presentation of the web pages, providing a user-friendly experience.
* The index.html template contains an input form where users can enter their consumer complaints. When the form is submitted, the complaint text is sent as a POST request to the Flask backend for prediction.
* The Flask application defines a '/predict' endpoint to handle incoming POST requests.
* Upon receiving a complaint text from the input form, the backend performs TF-IDF vectorization using the loaded vectorizer.
* The vectorized complaint is then passed to the trained model for prediction.
* The predicted class label is returned to the frontend for display on the result.html page.

Results and Discussion

Upon training the consumer complaint dataset using various classification models, it was observed that the LinearSVC model generalized better compared to other models. As a result, the LinearSVC model was selected for further deployment for complaint classification in real-world scenarios.

Preprocessing steps, including TF-IDF vectorization and feature extraction, were important to building a good classification model. TF-IDF vectorization transformed raw text data into numerical features, capturing the importance of words within documents.

Performance evaluation metrics such as accuracy, precision, recall, and F1-score were used to assess the models' predictive performance. The LinearSVC model consistently exhibited high accuracy and balanced performance across multiple evaluation metrics.

In the future, the deployed model can be optimized and fine-tuned to accommodate growing consumer complaint data and requirements and advanced techniques such as ensemble learning, deep learning, and transfer learning could be pursued to enhance the model's predictive capabilities and robustness.

.

Conclusion

In this project, we successfully developed and deployed a consumer complaint classification model aimed at automating the categorization of consumer complaints. Through the process of data preprocessing, model selection, and deployment, the below wa achieved:

* After evaluating various classification algorithms, the LinearSVC model emerged as the best, this showed excellent generalization on the complaint dataset. The model was trained using techniques including TF-IDF vectorization and feature engineering, to effectively capture the semantic content and importance of words within documents.
* Using Flask, HTML, and CSS, the trained LinearSVC model was deployed into a user-friendly web application accessible via web browsers. The interface allows users to input their complaints, with real-time predictions generated by the deployed model. HTML templates and CSS stylesheets were used to create visually appealing and responsive web pages, increasing user experience.
* The evaluation metrics confirmed the effectiveness of the deployed model in accurately classifying consumer complaints, with high accuracy and performance across multiple evaluation metrics.

Appendix

Link to Dataset - [Consumer Complaint Database (kaggle.com)](https://www.kaggle.com/datasets/selener/consumer-complaint-database)

Link to Github Repository - https://github.com/Micah-Shallom/customer\_complaint\_classification.git

Link to colab Notebook - https://colab.research.google.com/drive/1QwhRjcfEade-ek2VQB4BLNC1z3GJXV4O?usp=sharing