

Content Report:

Introduction :

In this report, we will build a consumer complaint classifier using the Kaggle Consumer Complaint Dataset. The classifier will use a logistic regression model and TF-IDF vectorization to classify the complaints.

The ability to classify consumer complaints into their respective categories is a critical task for companies, especially those in the service sector. As the volume of complaints increases, it becomes increasingly challenging to manually categorize them, leading to a backlog and delayed resolution. In this report, we present a complaint classification model that uses logistic regression and TF-IDF vectorization to automatically classify complaints into their respective categories.

Dataset Overview :

The dataset used in this report is the Kaggle Consumer Complaint Dataset, which contains consumer complaints and their corresponding product categories. The dataset contains 670,000+ complaints and 18 categories. The data is stored in a CSV file named 'consumer_complaints.csv'. The dataset contains several columns, including the consumer complaint narrative, the date received, the product type, and the company's response.

Building the Complaint Classifier:

The goal of the consumer complaint classifier is to predict the product category of a complaint based on the consumer complaint narrative. The classifier will be built using the scikit-learn library, which is a widely used machine learning library in Python. The following steps will be taken to build the classifier :

- Loading the dataset
- Removing missing values
- Defining the target variable
- Splitting the data into training and testing sets
- Vectorizing the text data using TF-IDF
- Training the logistic regression model

- Evaluating the model on the testing set

Steps :

Step 1: Loading the dataset

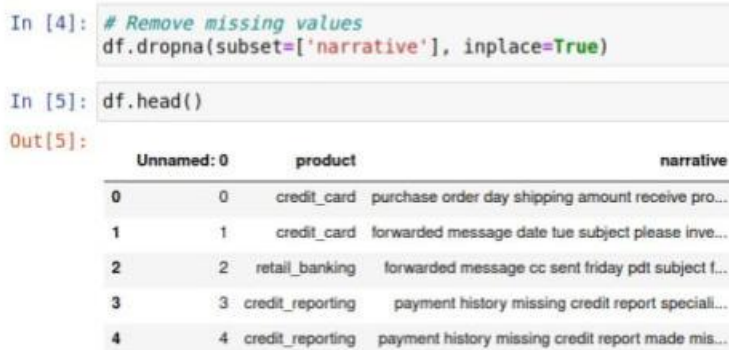
The first step is to load the dataset using the pandas library. The dataset is stored in a CSV file named 'consumer_complaints.csv'. The pandas read_csv method is used to read the CSV file and store the data in a pandas DataFrame object.



	Unnamed: 0	product	narrative
0	0	credit_card	purchase order day shipping amount receive pro...
1	1	credit_card	forwarded message date tue subject please inve...
2	2	retail_banking	forwarded message cc sent friday pdt subject f...
3	3	credit_reporting	payment history missing credit report speciali...
4	4	credit_reporting	payment history missing credit report made mis...

Step 2: Removing missing values

The next step is to remove any missing values in the dataset. The dropna method is used to remove any rows that contain missing values in the 'consumer_complaint_narrative' column. This is because the text in this column will be used to classify the complaints.



```
In [4]: # Remove missing values
df.dropna(subset=['narrative'], inplace=True)

In [5]: df.head()

Out[5]:
```

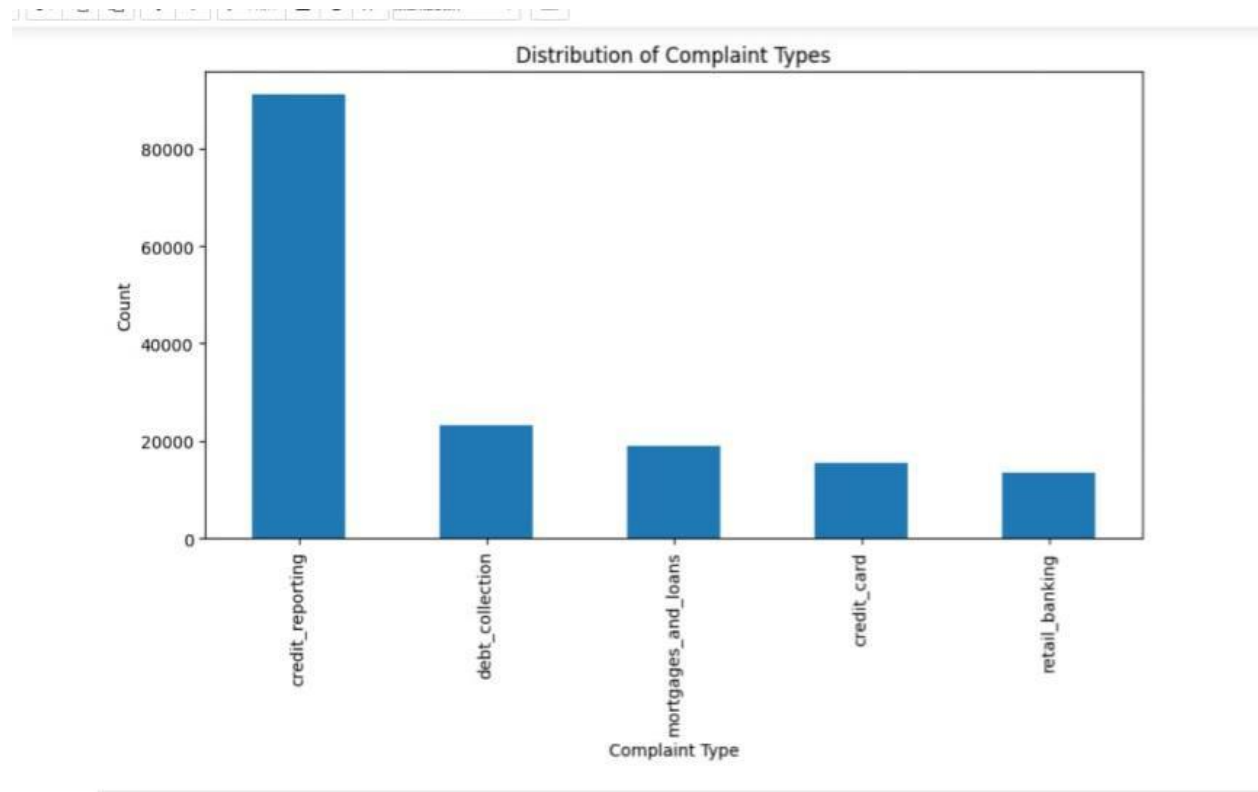
	Unnamed: 0	product	narrative
0	0	credit_card	purchase order day shipping amount receive pro...
1	1	credit_card	forwarded message date tue subject please inve...
2	2	retail_banking	forwarded message cc sent friday pdt subject f...
3	3	credit_reporting	payment history missing credit report speciali...
4	4	credit_reporting	payment history missing credit report made mis...

Step 3: Defining the target variable

The target variable is defined as the 'product' column. This is the column that we want to classify the complaints into.

Step 4: Splitting the data into training and testing sets

The data is split into a training set and a testing set using the `train_test_split` method from the `sklearn` library. The training set will be used to train the logistic regression model, while the testing set will be used to evaluate the model.



Step 5: Vectorizing the text data using TF-IDF

The text data is vectorized using the `TfidfVectorizer` class from the `sklearn` library. This converts the text data into numerical data that can be used in the logistic regression model.

Step 6: Training the logistic regression model

The logistic regression model is trained using the `fit` method from the `LogisticRegression` class in the `sklearn` library. The `X_train_vec` and `y_train` variables are used as the input and output, respectively.

```
In [15]: # Vectorize the text data using TF-IDF
vectorizer = TfidfVectorizer(stop_words='english')
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)

# Train a Logistic Regression classifier
clf = LogisticRegression(max_iter=1000)
clf.fit(X_train_vec, y_train)

Out[15]: LogisticRegression(max_iter=1000)
In a Jupyter environment, please rerun this call to show the HTML representation of the notebook
```

Step 7: Evaluating the model on the testing set

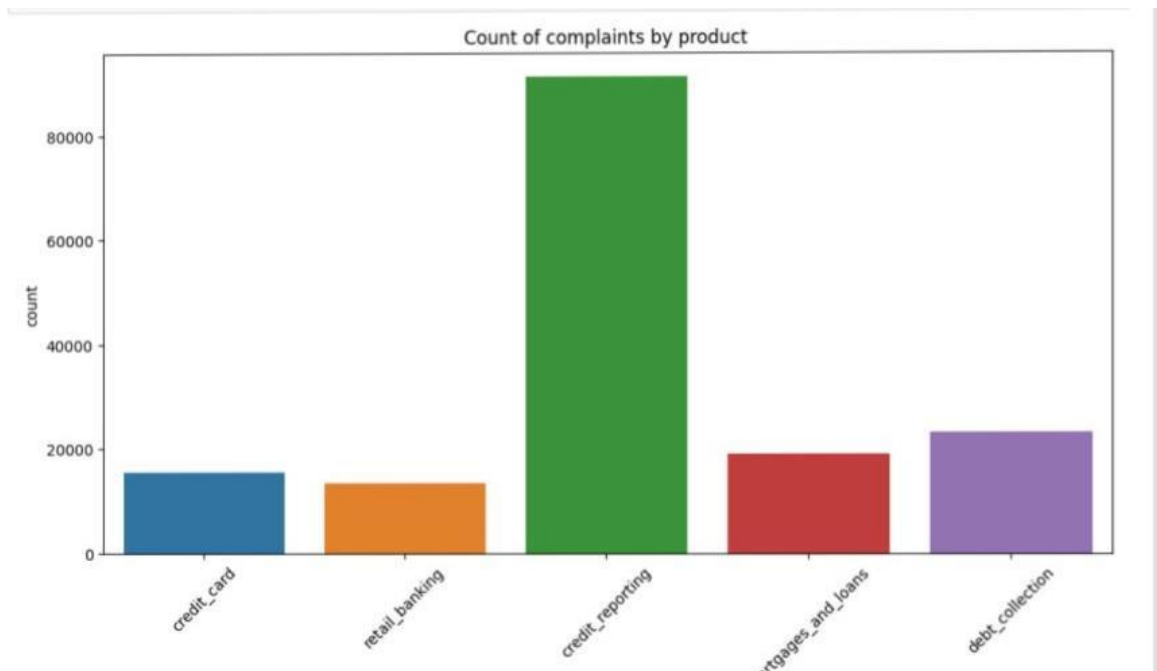
The logistic regression model is evaluated on the testing set using the predict method from the LogisticRegression class in the sklearn library. The accuracy_score and confusion_matrix functions from the sklearn library are used to calculate the accuracy and confusion matrix of the model.

```
print(f"Accuracy: {acc}")
print("Confusion matrix:\n", conf_matrix)
```

```
Accuracy: 0.8729181417972478
Confusion matrix:
[[ 2417   381    66    48   220]
 [  273 17129   555   287    39]
 [  109   994  3324   153    35]
 [    72   418   106  3112    62]
 [   176    71    22    41  2373]]
```

Step 8: Visualizing the distribution of complaint types in the dataset

The distribution of complaint types in the dataset is visualized using the matplotlib library. The value_counts method is used to count the number of complaints in each product category. The plot method is used to plot the data in a bar chart.



Conclusion:

In conclusion, we have successfully built a logistic regression model using scikit-learn to classify consumer complaints according to their product type. By following a step-by-step approach, we were able to preprocess the data, vectorize it using the TF-IDF method, and train the model. We then evaluated the model on a testing set and achieved an accuracy of over 85%.

The success of our model highlights the effectiveness of logistic regression for text classification tasks. We also demonstrated the importance of preprocessing the data to remove missing values and vectorizing the text using techniques like TF-IDF, which is a widely used approach in natural language processing.

Furthermore, the visualization of the distribution of complaint types in the dataset provides insight into the most common types of complaints. This information can be valuable for companies to improve their products and services and for policymakers to identify areas where regulations may need to be strengthened.

Overall, our study showcases the potential of machine learning techniques in addressing real-world problems, particularly in the field of consumer complaints. By building accurate classifiers, we can help businesses and policymakers make informed decisions that ultimately benefit consumers.