Problem Statement:

The problem statement is to develop a multi-label text classification model that categorizes user comments into categories like toxic, obscene, insult, severe toxic, identity hate, and threat based on their content.

Architecture:

1. **Data Loading and Exploration:**
   - Import libraries like Pandas, NumPy, and Matplotlib for data analysis and visualization.
   - Load the training and testing data from CSV files into Pandas DataFrames.
2. **Data Preprocessing:**
   - Data Cleaning:
     - Convert comment text to lowercase to ensure uniformity.
     - Use regular expressions to remove special characters.
     - Handle contractions like "what's" to "what is" for better text normalization.
     - Remove common English stopwords to eliminate noise.
   - Calculate the character length for each comment in the training data and create a histogram to analyze the distribution of text length.
3. **Label Analysis:**
   - Check the distribution of labels (toxic, severe toxic, obscene, threat, insult, identity hate) in the training data.
   - Determine the number of comments with no labels (considered "clean").
4. **Text Vectorization (TF-IDF):**

   - Use the TfidfVectorizer from scikit-learn to convert the cleaned text data into a TF-IDF representation.
   - Fit and transform the training data to create a document-term matrix (X_dtm).
   - Transform the test data using the same vocabulary to create a document-term matrix (test_X_dtm).
5. **Binary Relevance Model:**
   - For each label (toxic, severe toxic, obscene, threat, insult, identity hate):

- Create a Logistic Regression classifier.
- Train the model on the TF-IDF representation of the training data.
- Calculate the training accuracy for the model.
- Predict the probabilities of labels for the test data.
- Store the probabilities in the submission file.

6. **Classifier Chains Model:**
   - For each label:
     - Create a Logistic Regression classifier.
     - Train the model on the TF-IDF representation of the training data, including the previous labels in the chain.
     - Calculate the training accuracy for the model.
     - Predict the probabilities of labels for the test data, considering the previous labels in the chain.
     - Store the probabilities in the submission file.
     - Update the TF-IDF representations for both training and test data to include the predicted labels.
7. **Submission File Generation:**
   - Create submission files for both the binary relevance and classifier chains models.
   - Each submission file contains the predicted probabilities for each label for the test data.

Result:

For the "obscene" class, the training accuracy is approximately 98.32%.

For the "insult" class, the training accuracy is approximately 98.18%.

For the "toxic" class, the training accuracy is approximately 96.76%.

For the "severe_toxic" class, the training accuracy is approximately 99.31%.

For the "identity_hate" class, the training accuracy is approximately 99.56%.

For the "threat" class, the training accuracy is approximately 99.86%.

Conclusion:

In this report, we addressed the task of multi-label text classification to categorize user comments into various categories such as toxic, obscene, insult, severe toxic, identity hate, and threat. Two models were developed and evaluated: the Binary Relevance Model and the Classifier Chains Model.

Data preprocessing involved cleaning the text data, including lowercasing, removing special characters, handling contractions, and eliminating stopwords. We analyzed label distributions, revealing the presence of "clean" comments without any labels.

Using TF-IDF vectorization, the models were trained on the cleaned text data, achieving high training accuracies for each label. The Binary Relevance Model achieved accuracies ranging from approximately 96.76% to 99.86%, while the Classifier Chains Model incorporated label dependencies.

The report highlights the need to further evaluate these models using validation data or cross-validation techniques to assess their generalization capabilities. Additionally, hyperparameter tuning and model optimization could enhance their performance. Overall, the report provides a solid foundation for multi-label text classification, with the understanding that fine-tuning and rigorous evaluation will be crucial for real-world deployment.