TOXIC COMMENT CLASSIFICATION

*PARITOSH SHARMA*

*17 MAY 2018*

**Contents**

**1 Introduction**

1.1 Problem Statement

1.2 Data

**2 Methodology 4**

2.1 Pre Processing

2.1.1 Outlier Analysis

2.1.2 Feature Selection

2.2 Modeling

2.2.1 Model Selection

**3 Conclusion 15**

3.1 Model Evaluation

**Chapter 1**

**Introduction**

**1.1 Problem Statement**

Discussing things you care about can be difficult. The threat of abuse and harassment online means that many people stop expressing themselves and give up on seeking different opinions. Platforms struggle to effectively facilitate conversations, leading many communities to limit or completely shut down user comments.

The Conversation AI team, a research initiative founded by Jigsaw and Google (both a part of Alphabet) are working on tools to help improve online conversation. One area of focus is the study of negative online behaviors, like toxic comments (i.e. comments that are rude, disrespectful or otherwise likely to make someone leave a discussion). So far they’ve built a range of publicly available models served through the Perspective API, including toxicity. But the current models still make errors, and they don’t allow users to select which types of toxicity they’re interested in finding (e.g. some platforms may be fine with profanity, but not with other types of toxic content).

In this project, I’ve to build a multi-headed model that’s capable of detecting different types of toxicity like threats, obscenity, insults, and identity-based hate better than Perspective’s current models. I’ll be using a dataset of comments from Wikipedia’s talk page edits.

* 1. **Data**

I am provided with a large number of Wikipedia comments which have been labeled by human raters for toxic behavior. The types of toxicity are:

* Toxic
* severe\_toxic
* obscene
* threat
* insult
* identity\_hate

I must create a model which predicts a probability of each type of toxicity for each comment.

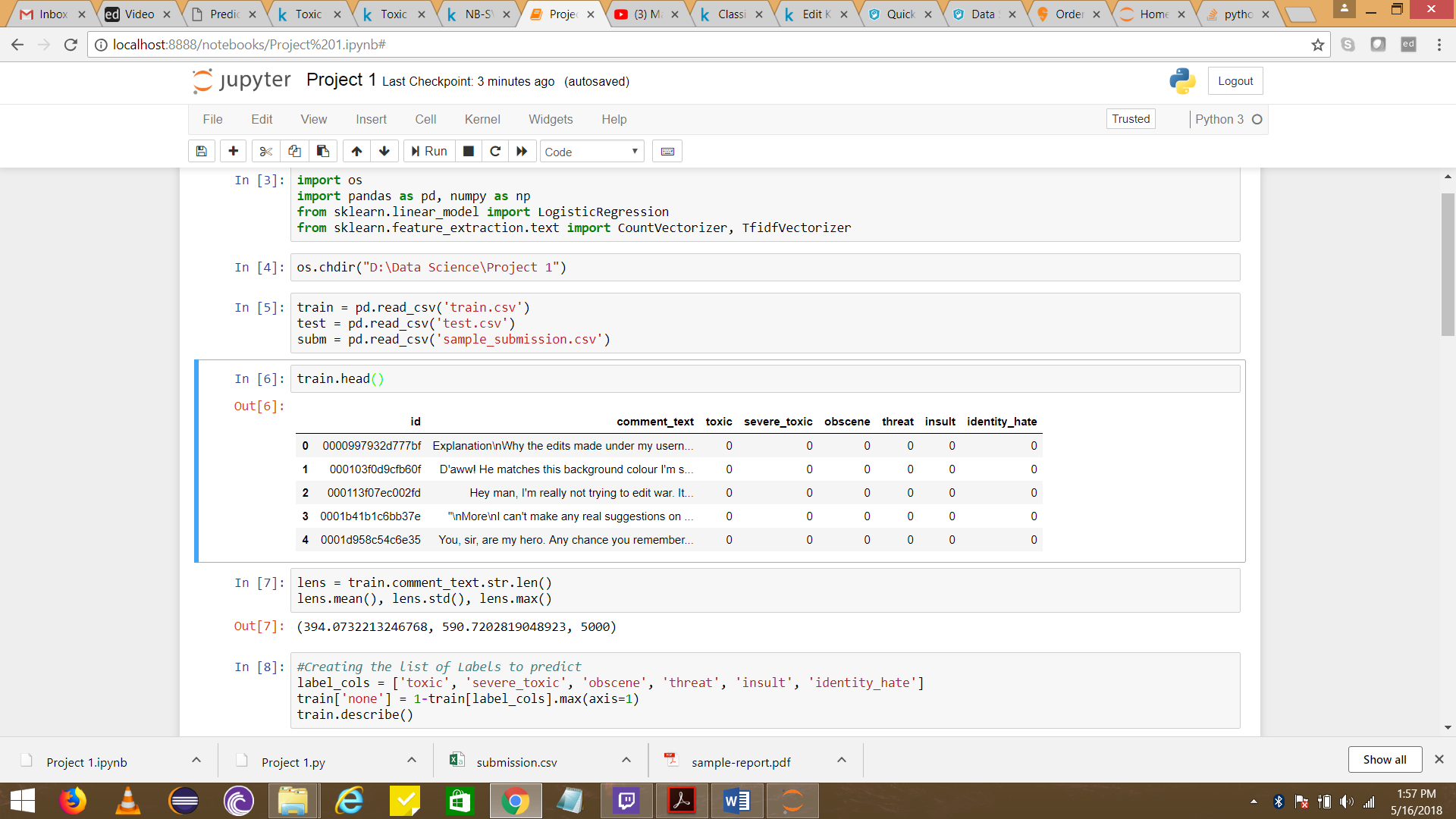


Table 1.2.1 Shows the Training Data

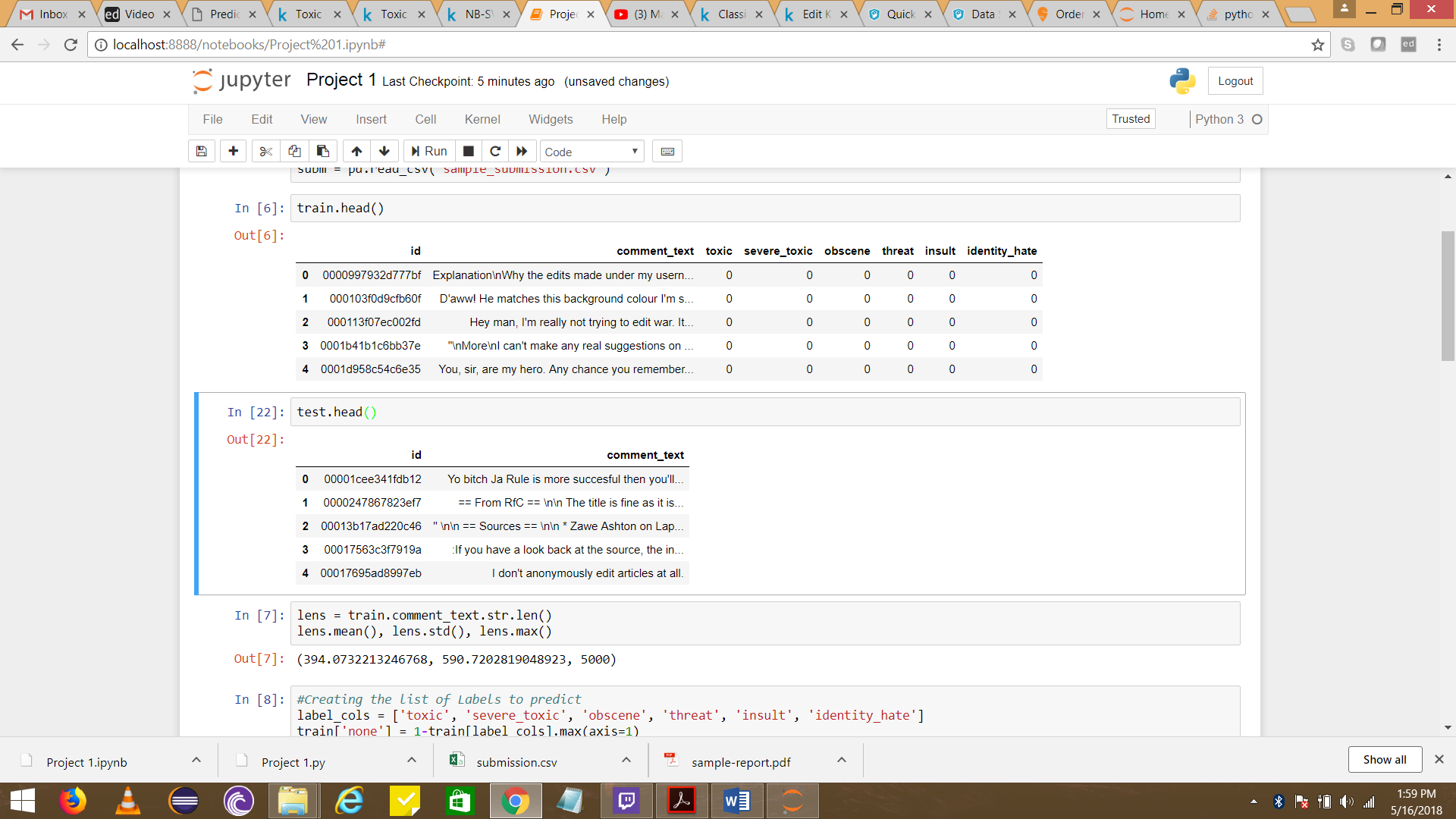


Table 1.2.2: Shows the sample Test Data

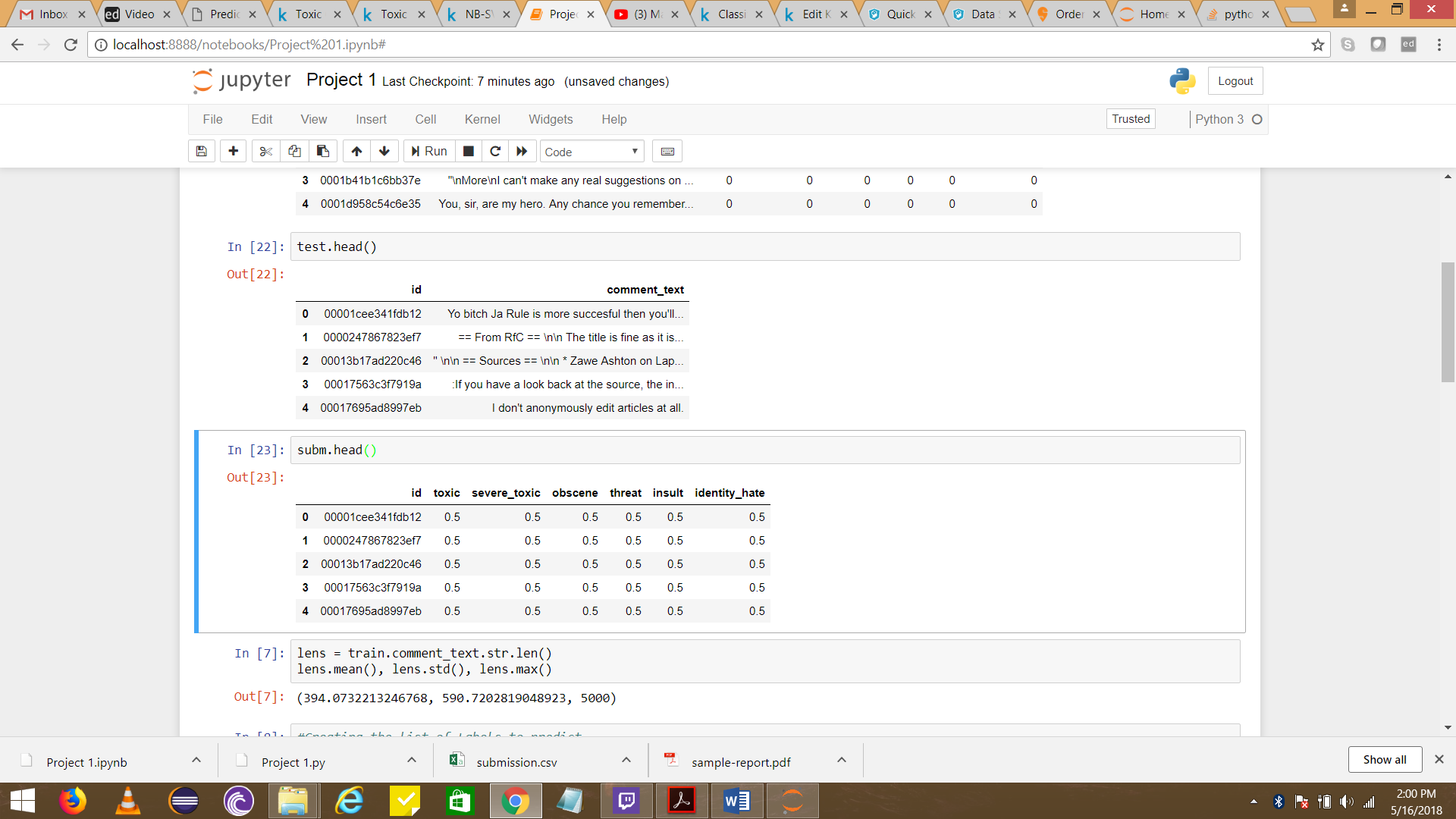


Table 1.2.3: Shows the Sample Submission Data in which the Final Report has to be submitted.

**Chapter 2**

**Methodology**

**2.1 Pre Processing**

As we all know what type of data we get in Text Mining Operations therefore it is a must that the data is pre-processed before we make a model on it. In Text Mining the pre-processing of the data is done by cleaning it and making a matrix at the end of it, which is called Document Term Matrix.

But before jumping to the conclusion we first have to look at the data and analyze it which is called Exploratory Data Analysis.

Since the Training and Testing Datasets are already given, we will compare both the data sets as both the data sets must be equal otherwise the SKLEARN library won’t run the program.

After doing such comparison we will equalize both the data sets by removing null values.

**2.1.1 Outlier Analysis**

After doing the Exploratory Data Analysis we will do the Outlier Analysis and will do the Pre Processing of the Data.

In Text Mining operation, the pre-processing is generally done by getting rid of punctuations, numbers, stop words, white spaces, and convert the upper class letters to lower class letter.

After doing the pre-processing, we’re left with bunch of words on which we have to make the model.

These words are called BAG OF WORDS. From these bag of words we create Term Document Matrix. On that Term Document Matrix we apply the Naïve Bayes Algorithm to make our Desired Model.

**2.1.2 Feature Selection**

As we know that in Naïve Bayes every word is treated as a unique feature therefore we create Term Document Matrix and pass it through the Naïve Bayes Classifier.

**2.2 Modeling**

**2.2.1 Model Selection**

It is already defined that we have to classify the given comments on the level of their toxicity. Therefore we concluded that this is a Classification Problem.

Now, from our early stages of analysis during pre-processing we have come to understand this problem belongs to Multi Label Classification Problem.

Therefor we concluded that we have to use the Naïve Bayes Algorithm for this Problem Statement and classify the variable into 6 different categories.

**2.2.1 Naïve Bayes**

Naïve Bayes is a Classification technique based on the Bayes Theorem where it assumes that every feature in the class is different from any other feature even though it is not.

Naïve Bayes is mostly used in Text classification due to better result in multi class problems and independence rule. It has higher success rate as compared to other algorithms. As a result, it is widely used in Spam Filtering and Sentiment Analysis

**Chapter 3**

**Conclusion**

**3.1 Model Evaluation**

Now that we have created the model and submitted the data set, we have successfully predicted the toxicity of the comments.