Capstone3_Final_Report

Background

Here's the background: When the Conversation AI team first built toxicity models, they found that the models incorrectly learned to associate the names of frequently attacked identities with toxicity. Models predicted a high likelihood of toxicity for comments containing those identities (e.g. "gay"), even when those comments were not actually toxic (such as "I am a gay woman"). This happens because training data was pulled from available sources where unfortunately, certain identities are overwhelmingly referred to in offensive ways. Training a model from data with these imbalances risks simply mirroring those biases back to users.

The Conversation AI team, a research initiative founded by Jigsaw and Google (both part of Alphabet), builds technology to protect voices in conversation. A main area of focus is machine learning models that can identify toxicity in online conversations, where toxicity is defined as anything rude, disrespectful or otherwise likely to make someone leave a discussion.

Problem Statement

The toxic comments are a severe issue on social media, because it leads to people expressing their opinion irrationally and trying to harm each other's feelings. The goal of this project is to use a deep learning model to identify toxicity in text and try to suggest the solution for classifying the toxic comments in various categories using natural language processing concepts.

Data Source

Source: https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/data

Disclaimer: The dataset for this competition contains text that may be considered profane, vulgar, or offensive.There are total 1804874 rows and 44columns in the dataset train.csv - the training set, which includes toxicity labels and subgroups

UPDATE (Nov 18, 2019): The following files have been added post-competition close to facilitate ongoing research. See the File Description section for details.

test_public_expanded.csv

test_private_expanded.csv

toxicity_individual_annotations.csv

identity_individual_annotations.csv

In the data supplied for this competition, the text of the individual comment is found in the comment_text column. Each comment in Train has a toxicity label (target), and models should predict the target toxicity for the Test data. This attribute (and all others) are fractional values which represent the fraction of human raters who believed the attribute applied to the given comment. For evaluation, test set examples with target >= 0.5 will be considered to be in the positive class (toxic).

The data also has several additional toxicity subtype attributes. Models do not need to predict these attributes for the competition, they are included as an additional avenue for research. Subtype attributes are:

severe_toxicity
obscene
threat
insult
identity_attack
sexual_explicit
Additionally, a subset of comments have been labelled with a variety of identity attributes, representing the identities that are mentioned in the comment. The columns corresponding to identity attributes are listed below. Only identities with more than 500 examples in the test set (combined public and private) will be included in the evaluation calculation. These identities are shown in bold.

- male
- female
- transgender
- other_gender
- heterosexual
- homosexual_gay_or_lesbian
- bisexual
- other_sexual_orientation
- christian
- jewish
- muslim

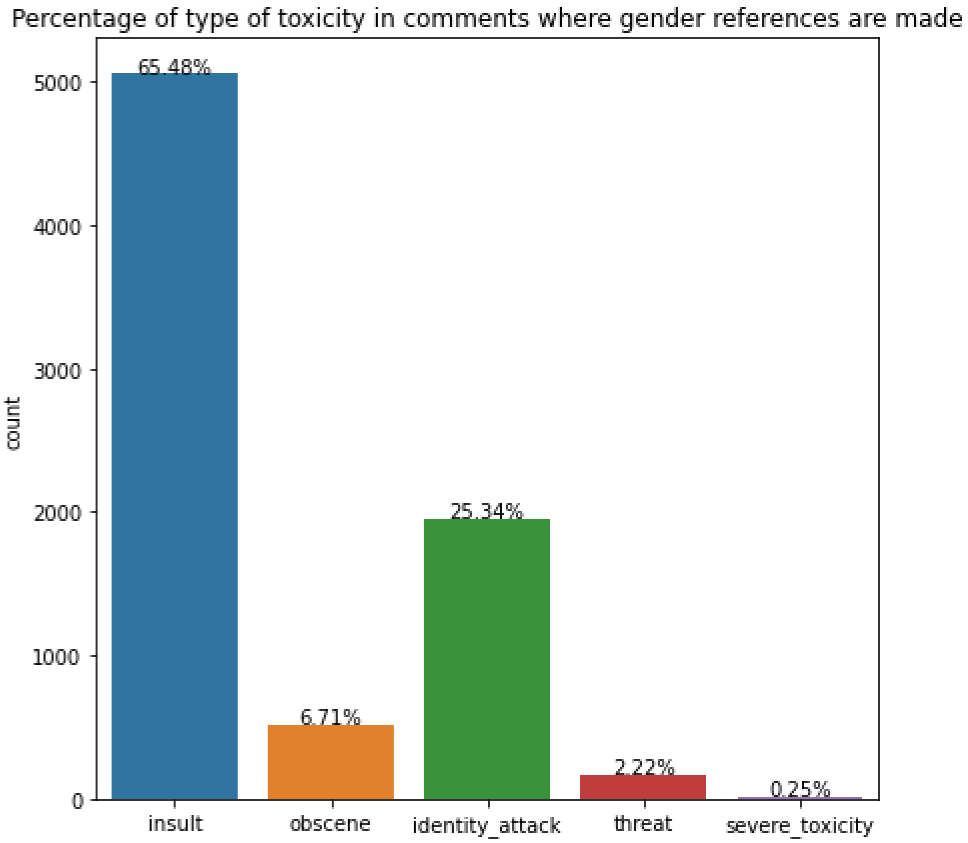
- hindu
- buddhist
- atheist
- other_religion
- black
- white
- asian
- latino
- other_race_or_ethnicity
- physical_disability
- intellectual_or_learning_disability
- psychiatric_or_mental_illness
- other_disability

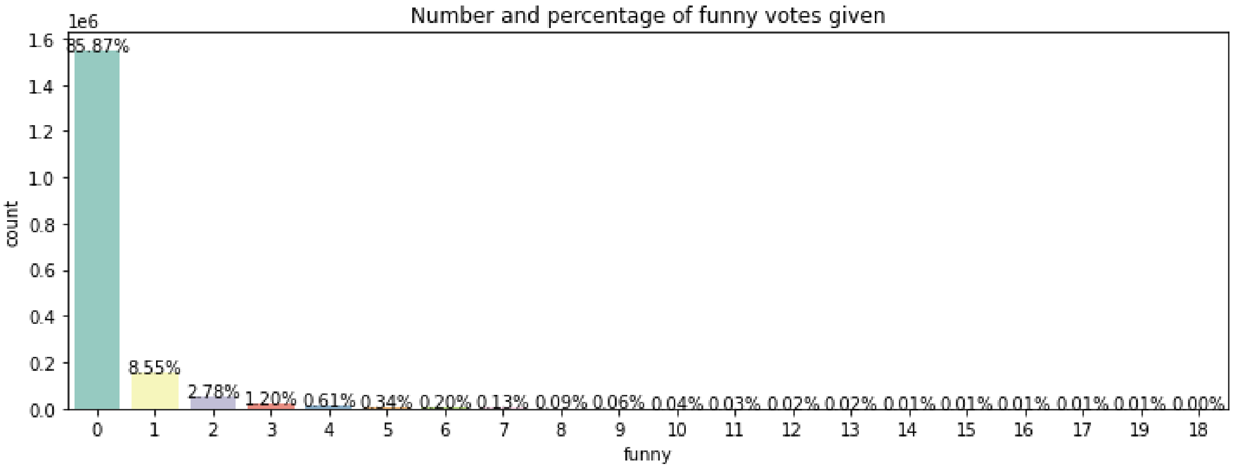
Note that the data contains different comments that can have the exact same text. Different comments that have the same text may have been labeled with different targets or subgroups.

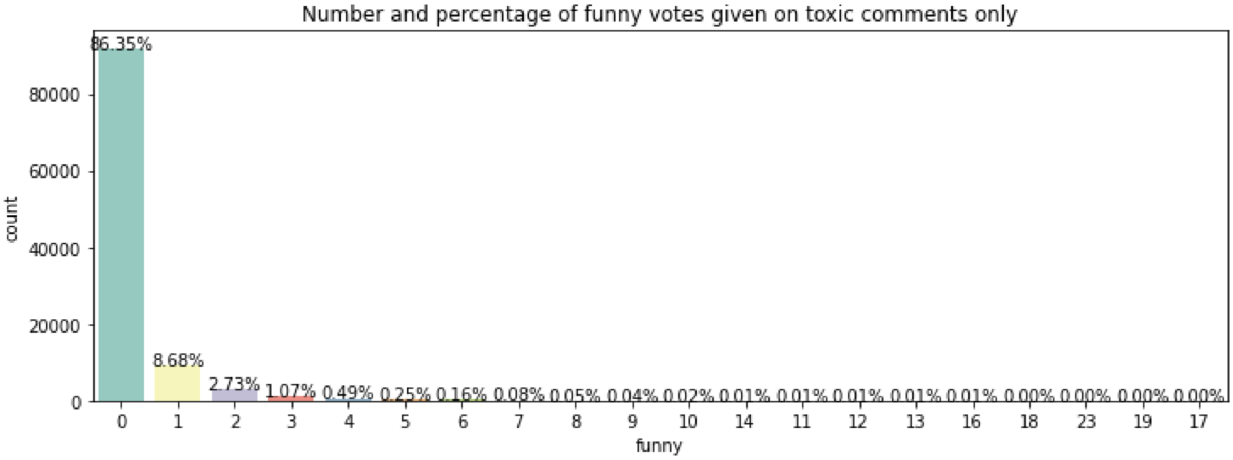
Data cleaning and Exploration

Firstly, we should import the dataset by pandas and conduct some basic clearing tasks to remove some information that is not relevant for purpose. Therefore, this stage focus on collecting the data, organizing it and making sure it is well defined into format.

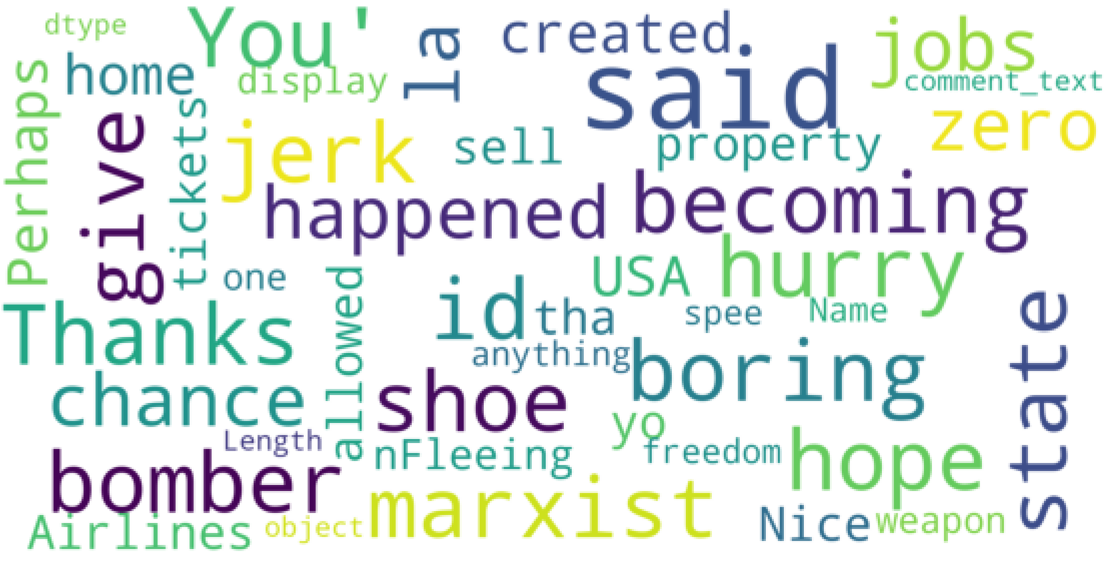
After that, conducting exploratory data analysis and understanding the target variable in the training dataset. Plot the distribution of the target variable and it shows that most of the comments in the dataset are non-toxi(<0.5). The value <0.5 is non_toxic else it is toxic, then encoding them into two categories. There are around 92% non-toxic comments and 8% toxic comments. Compared with other kinds of comments among toxic comments, insulting comments appear most frequently.


Percentage of type of toxicity in comments where gender references are made

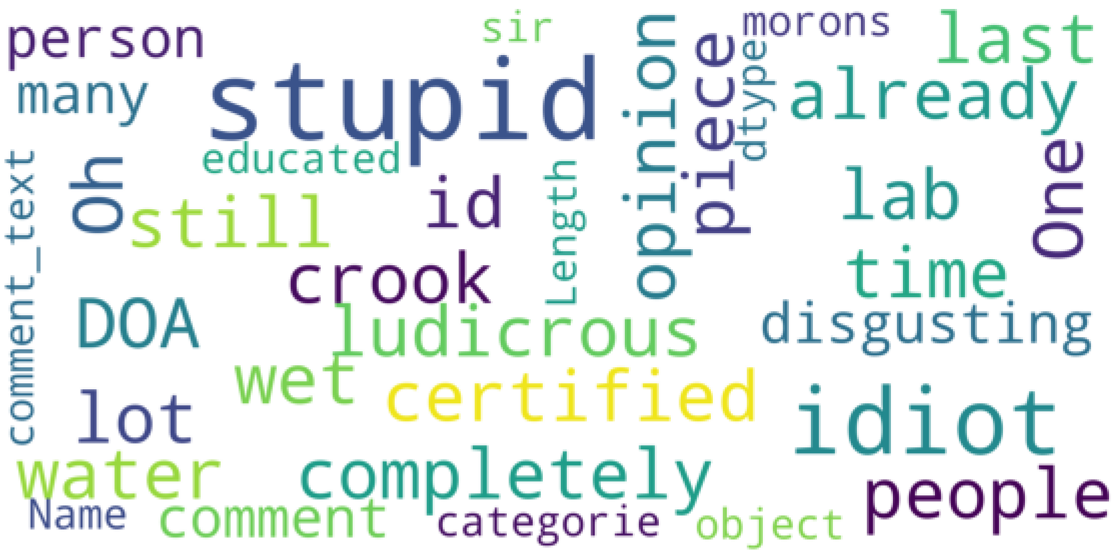

Number and percentage of funny votes given



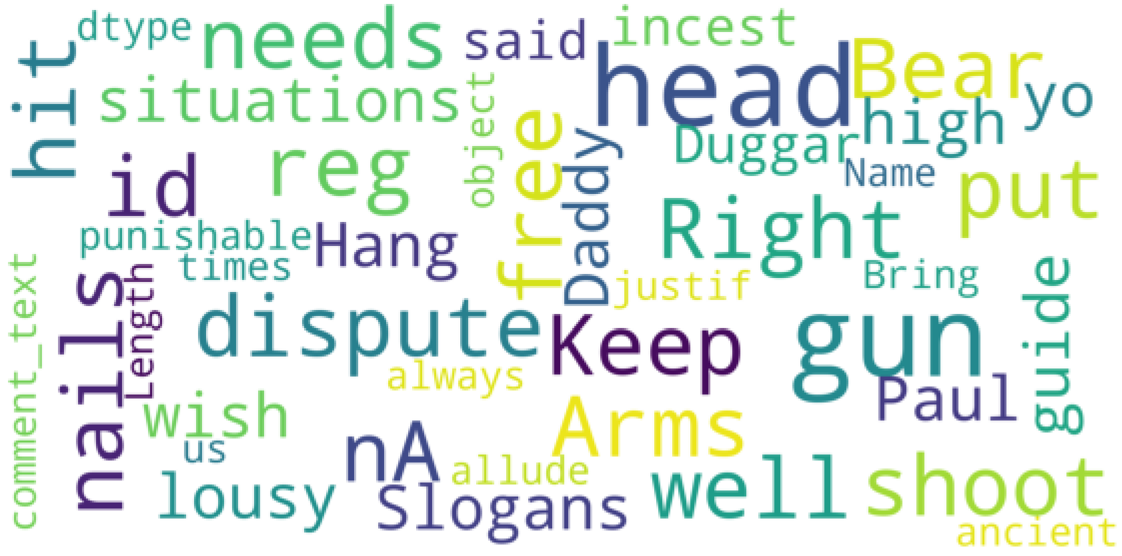
A subset of comments have been labelled with a variety of identity attributes, representing the identities that are mentioned in the comment. The columns corresponding to identity attributes are listed below. Only identities with more than 500 examples in the test set (combined public and private) will be included in the evaluation calculation. These identities are shown in bold.



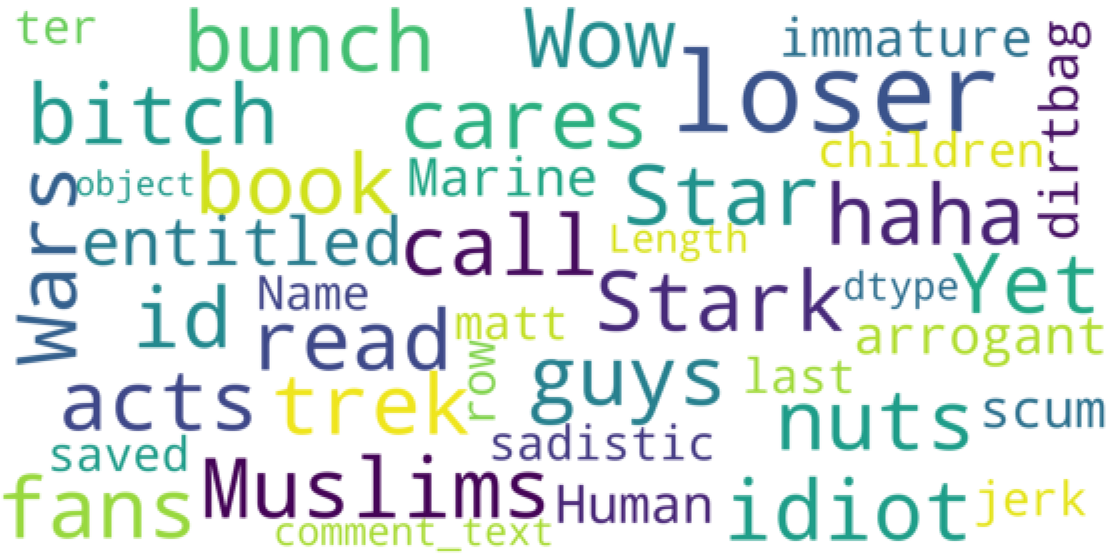
Prevalent words in comments - train data



Prevalent comments with insult score > 0.75



Prevalent words in comments with threat score > 0.75



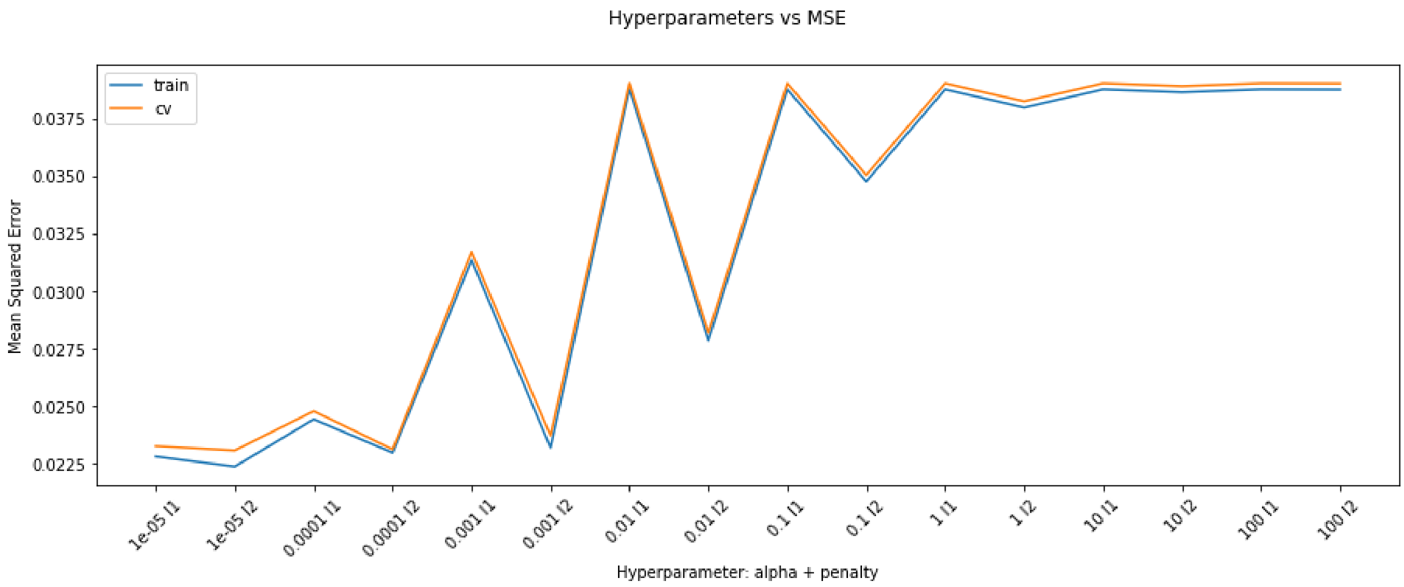
Prevalent words in comments with target score > 0.75

Machine Learning Model Apply for predicting Toxic Comments

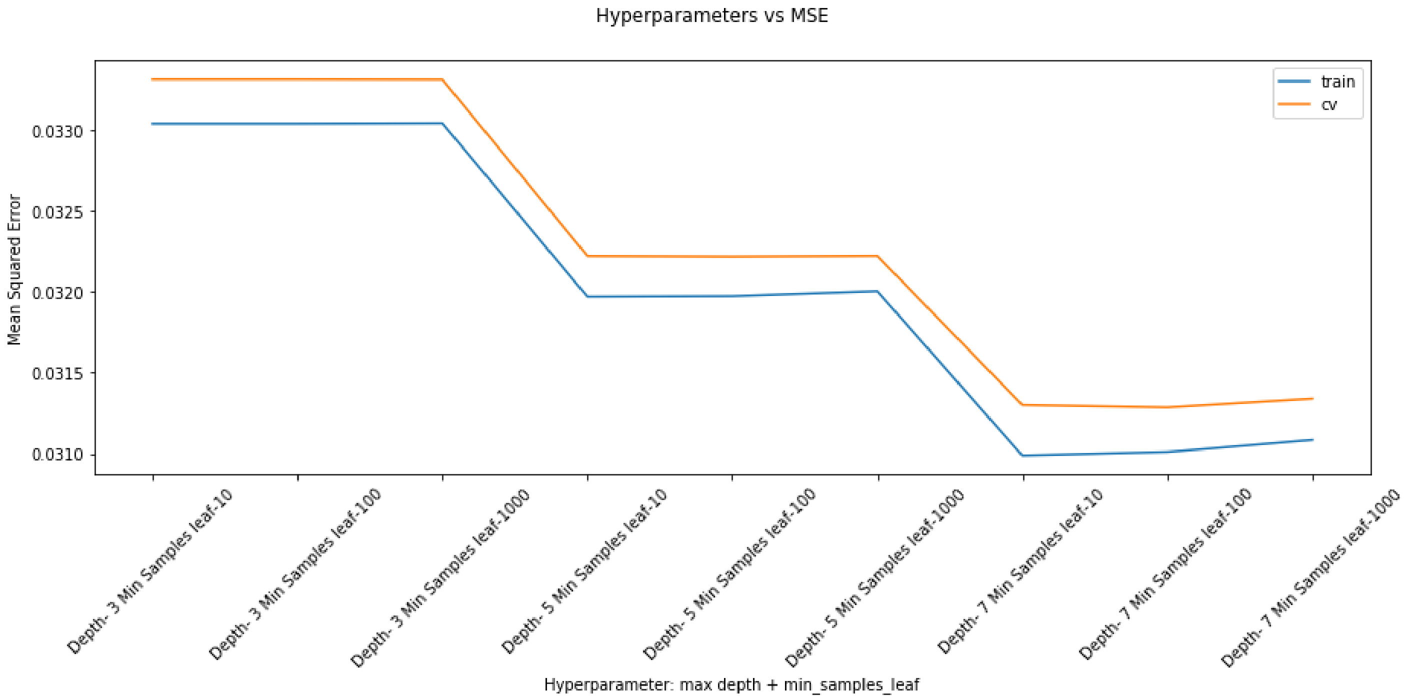
1. **Apply for machine learning models**
   Applying for the models for the preprocessed dataset and there are three models including SDBGRegressor Model, Decision Tree Model.

2. Model Comparison
   The result of each model performance are great and all of them achieve great score.

   Below is Bag of Words(BoW) model performance in SGDRegressor:



Below is the Hyperparameter tuning plot in Decision Trees:



Conclusions:

1. **BagOfWords:**
   - *SGDRegressor:*

- Hyperparameters Tuned Values: learning_rate(alpha): 1e-05 and penalty: l2
- Train MSE Loss: 0.02281
- CV MSE Loss: 0.02326

- *Decision Tree:*
    - Hyperparameters Tuned Values: max_depth: 7 and min_samples_leaf: 100
    - Train MSE Loss: 0.0310
    - CV MSE Loss: 0.03128

2. **TFIDF:**
    - *SGDRegressor:*
        - Hyperparameters Tuned Values: learning_rate(alpha): 1e-05 and penalty: l2
        - Train MSE Loss: 0.02556
        - CV MSE Loss: 0.02584
    - *Decision Tree:*
        - Hyperparameters Tuned Values: max_depth: 7 and min_samples_leaf: 100
        - Train MSE Loss: 0.03073
        - CV MSE Loss: 0.03122