

While implementing the Random Forest classification model to predict controversy based on Reddit comments, I encountered a significant issue. Initially, I opted to use TF-IDF vectorization for feature extraction from the comments. TF-IDF is typically effective for highlighting the importance of words relevant to specific contexts in text data, which I anticipated would enhance the model's ability to discern patterns correlating with controversy.

However, upon integrating TF-IDF with the Random Forest model, I encountered a persistent error during the model evaluation phase: "classification metrics can't handle a mix of multilabel-indicator and binary targets." This error suggested that there was an incompatibility between the output of my model and the format expected by the classification metrics functions in Scikit-learn, which are designed to evaluate binary targets. This indicated that the predictions or the label format might have been inadvertently altered during processing or model prediction, possibly being treated as multilabel (where each class label is independent and not mutually exclusive) rather than binary as intended.

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-35-1edcc020a56a> in <cell line: 25>()
    23
    24 # Evaluate model performance
--> 25 accuracy = accuracy_score(y_test, y_pred)
    26 precision = precision_score(y_test, y_pred)
    27 recall = recall_score(y_test, y_pred)

----- 2 frames -----
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py in _check_targets(y_true, y_pred)
    93
    94     if len(y_type) > 1:
--> 95         raise ValueError(
    96             "Classification metrics can't handle a mix of {0} and {1} targets".format(
    97                 type_true, type_pred)
ValueError: Classification metrics can't handle a mix of multilabel-indicator and binary targets
```

Furthermore, a perplexing outcome of using TF-IDF was the observation of feature vectors predominantly composed of zeros, like `[0,0]`. This occurred because the TF-IDF vectorizer, when applied to the dataset, resulted in an overly sparse matrix. Such sparsity arose because many of the words in the comments did not match the vocabulary generated during the vectorizer's fitting process, which only included the most statistically significant words from the training set. Consequently, comments that predominantly contained less significant words were transformed into feature vectors with very low or zero values, indicating an absence of the keywords that the model was trained to recognize as influential.

Given these challenges, I decided to switch from TF-IDF to Count Vectorization. Count Vectorization directly tallies word frequencies and avoids assigning weighted scores like TF-IDF, which I hoped would reduce sparsity in the feature set and better align with the binary classification framework of my Random Forest model. This change aimed to ensure that the feature vectors more robustly represented the presence of training-set words in the test data, improving model performance and compatibility with the classification metrics.

[illegible]