Project Milestone 3 - Ensemble

Majority Ensemble

Members of the Majority Ensemble are:

1. Best SVM Bag Of Words Dataset
2. Best SVM Glove Dataset
3. Best SVM TFIDF Dataset
4. Best Decision Tree Misc Dataset
5. Best Simple Perceptron TFIDF Dataset
6. Best Aggressive Perceptron TFIDF Dataset

Goal

My initial goal was to save the weights and tree and load them to make new predictions, even though I used the Numpy save and a load of weights. When I did the dot product of these weights and examples. I got an error "UFuncTypeError: ufunc 'multiply' did not contain a loop with signature matching types (dtype('<U4892'), dtype('<U32')) -> None ". I tried my best to fix the error but could not. Nevertheless, I used the hack below mentioned.

Hack

In the previous milestones, I found the best dataset and parameters work for the above models. Then, I made predictions on the respective datasets and saved them into a CSV. Therefore, instead of loading the weights and making another prediction, we can directly use the CSV and pick the majority label as ensemble prediction, which I used.

```
import json
import random
import numpy as np
import pandas as pd
```

Eval Dataset Import of Members

```
decision_tree_eval_df = pd.read_csv("../../Milestone-01/decision_tree_misc_eval_datase
simple_perceptron_eval_df = pd.read_csv("../../Milestone-02/simple_perceptron_tfidf_ev
aggressive_perceptron_eval_df = pd.read_csv(
    "../../Milestone-02/aggressive_perceptron_tfidf_eval_dataset_prediction.csv"
)
svm_glove_eval_df = pd.read_csv("../../Milestone-03/svm/results/glove_lr_0.001_tradeof
svm_bow_eval_df = pd.read_csv("../../Milestone-03/svm/results/bow_lr_0.001_tradeoff_10
```

```
svm_tfidf_eval_df = pd.read_csv("../../Milestone-03/svm/results/tfidf_lr_0.001_tradeo


dfs = [
    decision_tree_eval_df["label"],
    simple_perceptron_eval_df["label"],
    aggressive_perceptron_eval_df["label"],
    svm_glove_eval_df["label"],
    svm_bow_eval_df["label"],
    svm_tfidf_eval_df["label"],
]


eval_df = pd.concat(dfs, axis=1)
eval_df.columns = [
    "decision_tree_labels",
    "simple_perceptron_labels",
    "aggressive_perceptron_labels",
    "svm_glove_labels",
    "svm_bow_labels",
    "svm_tfidf_labels",
]
eval_df
```

	decision_tree_labels	simple_perceptron_labels	aggressive_perceptron_labels	svm_glove_l
0	1	1	1	
1	1	1	1	
2	0	0	1	
3	1	1	1	
4	0	1	1	
...	...	...	...	
5245	0	0	0	
5246	1	1	1	
5247	1	1	1	
5248	1	1	1	
5249	1	1	1	

5250 rows × 6 columns

Majority Predictions & Results

```
def export_prediction_to_csv(file_name, prediction_list):
    df = pd.DataFrame(prediction_list)
```

```
    df.to_csv(f"results/{file_name}.csv", index=True, index_label="example_id", heade


def majority_ensemble_prediction(df):
    prediction_list = []
    for _, row in df.iterrows():
        predictions = row.tolist()

        negative_predict_count = 0
        positive_predict_count = 0
        for prediction in predictions:
            if prediction not in [0, 1]:
                print(f"Invalid prediction: {prediction}")
                break
            elif prediction == 1:
                positive_predict_count += 1
            else:
                negative_predict_count += 1

        if positive_predict_count == negative_predict_count:
            majority_prediction = random.randint(0, 1)

        elif positive_predict_count > negative_predict_count:
            majority_prediction = 1

        else:
            majority_prediction = 0

        prediction_list.append(majority_prediction)

    return prediction_list


prediction_list = majority_ensemble_prediction(eval_df)
export_prediction_to_csv(file_name="ensemble_eval_prediction.csv", prediction_list=pre
```