Assignment 5

1 Introduction

In this assignment we will act as data-scientists for a fictitious data. our mission is to build a model that predicts a 11-class category from 64 synthesized flags. The features mimic “present / absent” indicators such as abnormal lab values, symptoms, or billing codes, making the task surprisingly subtle despite the modest table size.

2 Data

2.1 Dataset

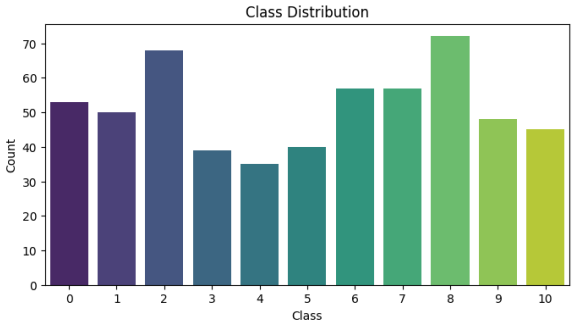
The train dataset has 564 rows and 64 binary features (feature0 –feature63) and the target label.

ID	feature0	feature1	feature2	feature3	feature4	feature5	feature6	feature7	feature8	...	feature55	feature56	feature57	feature58	feature59	feature60	feature61	feature62
0	1	1	0	1	1	1	1	0	1	...	0	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1	1	1	...	1	1	1	1	1	0	1	1
2	1	1	1	1	0	0	1	1	0	...	1	1	1	1	0	0	0	0
3	1	0	0	0	1	0	0	1	1	...	0	0	0	0	0	0	1	0
4	1	1	1	1	1	1	0	0	1	...	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
559	0	0	1	1	1	0	0	0	0	...	0	0	0	0	0	0	0	0
560	1	0	1	1	1	1	0	1	1	...	0	0	0	0	0	0	0	0
561	1	0	1	0	1	0	0	1	1	...	0	0	0	0	0	0	0	0
562	1	1	0	0	1	0	1	0	1	...	1	1	1	1	0	0	0	0
563	1	1	0	0	0	0	1	0	0	...	0	0	0	0	1	1	1	0

There are no missing values

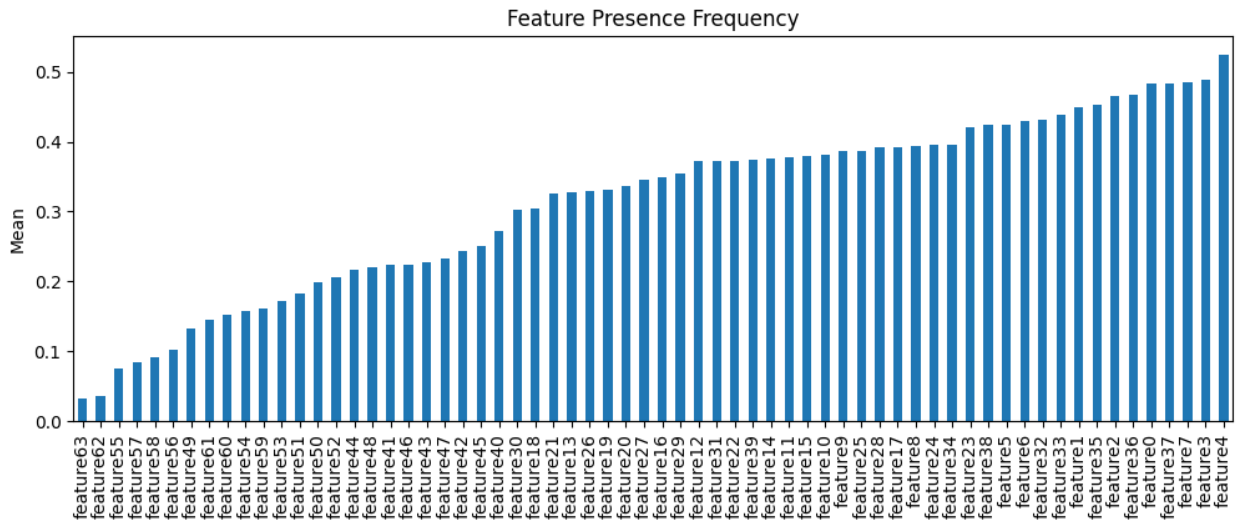
2.2 visualization

In this section we will go through some visualization related to the data that can help us to have a better sense of our data and the relationship between the features.

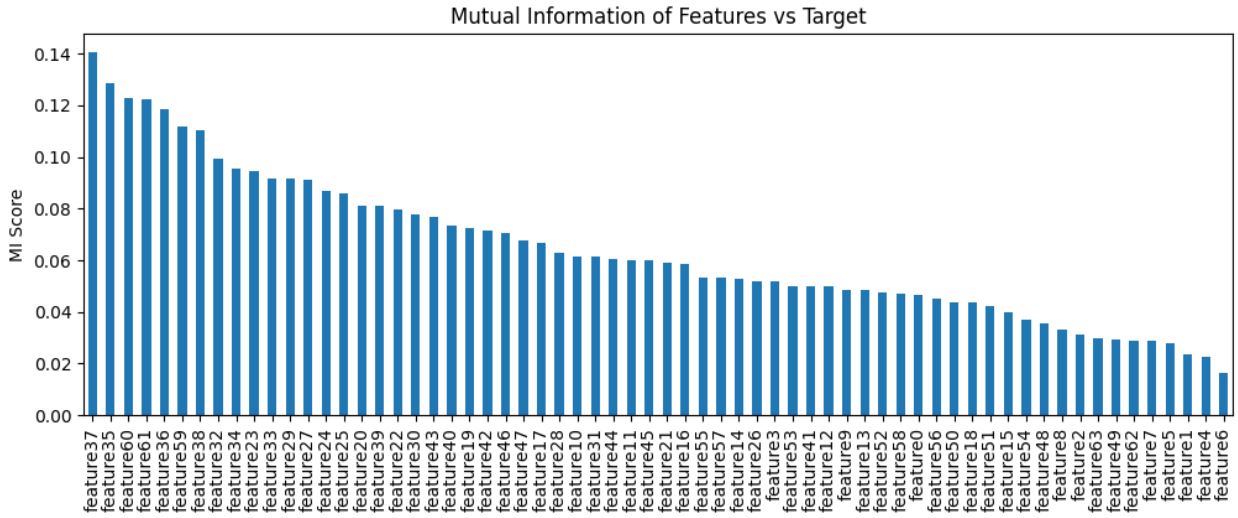


We have 11 classes and here we can see the distribution of data through these classes

.



Here we have the distribution of data through these classes. The mean value is relatively low (0.2 to 0.6), indicatingmost features are not frequently present. The extensive list suggests a high-dimensional dataset, but the low mean implies sparse feature usage or rare occurrences across the dataset.



The plot displays mutual information scores between features and a target variable, indicating predictive relevance. **Feature37** appears most important, followed by **feature35, feature60, feature61**, etc., suggesting these contribute significantly to the target.

3 Data Preparation and Baseline Modeling

To evaluate generalization, I stratified a train–test split with 20 % held out. Since support vector machines require scaled inputs, I applied StandardScaler only to the SVM's data; all other models—logistic regression, Bernoulli Naive Bayes, and a depth-five decision tree—were trained on the raw binary features, which these algorithms handle naturally.

After training these four baseline classifiers, their accuracies on the hold-out test set were 32.7 % for logistic regression, 37.2 % for SVM, 26.5 % for the decision tree, and 38.1 % for Naive Bayes. These results established an initial performance range and indicated that no single baseline outperformed 40 %, motivating further tuning and ensembling.

4 Hyperparameter Tuning of Tree-Based Methods

To harness the power of ensembling, I next tuned a RandomForestClassifier and an XGBClassifier. For the random forest, I searched over 100 and 200 trees and maximum depths of None, 10, and 20 using five-fold cross-validation on the training split. The best configuration was 200 trees with unlimited depth. For XGBoost, I tuned the number of rounds, learning rate (0.1 or 0.01), and tree depth (3 or 6), again using five-fold CV; the best setting used 200 rounds, a learning rate of 0.1, and maximum depth of 3. Both grid searches also fixed random_state=42 (and for XGBoost, I set eval_metric='mlogloss') to eliminate run-to-run variability.

5 Soft-Voting Ensemble Construction

Recognizing that different algorithms capture different patterns, I assembled a soft-voting classifier comprised of six members: logistic regression, SVM, decision tree, BernoulliNB, the tuned random forest, and the tuned XGBoost. Soft voting averages each model's predicted class probabilities, which often yields better calibration than hard majority voting. I fit the ensemble on the raw training features and then evaluated it on the 20 % hold-out set. The ensemble achieved an accuracy of 34.5 %, but for the actual test set on Kaggle I gained 0.394.

6 Detailed Evaluation

The classification report for the ensemble revealed strong performance on class 0 (precision 0.62, recall 0.91, F_1 0.74) and class 7 (precision 0.64, recall 0.64, F_1 0.64), while classes 1 and 5 saw zero recall and precision, indicating the model completely missed those categories. Mid-range classes such as 2 and 9 achieved F_1-scores around 0.22–0.32. The confusion matrix showed that many true tweets of classes 1 and 5 were misassigned to neighboring categories, suggesting these classes are both under-represented and share feature patterns with others.

6 Model Interpretation and Feature Importance

To move beyond accuracy numbers, I performed several feature-importance analyses:

1. Random Forest impurity importances: The top five features were feature11, feature3, feature1, feature5, and feature2—these contributed the largest average impurity decrease across all trees.
2. XGBoost gain importances: Gain-based scores identified feature55, feature61, feature59, feature37, and feature3 as the most valuable splits for reducing multi-class log loss.
3. Permutation importance on the ensemble: By randomly shuffling each feature in the test set and measuring accuracy drop, I found that feature61, feature53, feature8, feature30, and feature9 caused the greatest performance degradation when permuted.
4. SHAP values for XGBoost: A SHAP summary plot confirmed that high values of feature61 and feature55 strongly push predictions toward certain classes (e.g., class 7), while low values of feature53 and feature8 bias toward others (e.g., class 3).

In plain language, indicators 61 and 55 emerge as the strongest "red-flag" signals for class 7, meaning their presence in a tweet makes it much more likely to belong to that category. Conversely, indicators 53 and 8 appear critical for distinguishing class 3.

7 Recommendations for Data Collection and Triage

First, I recommend collecting or synthesizing (e.g., via class-conditional GANs) more examples that exhibit high values in feature61 and feature55 for under-represented classes, since these features drive correct classification for class 7 but may be sparse elsewhere. Second, data-labeling workflows should prioritize tweets containing feature53 and feature8 for human review when the model is uncertain—this focused triage will help accumulate clearer examples of class 3 and reduce confusion between neighboring categories.

By combining these targeted data-collection strategies with the existing ensemble, future iterations should achieve more balanced recall across all classes.

Conclusion

In conclusion, my soft-voting ensemble—combining logistic regression, SVM, decision tree, Bernoulli Naive Bayes, Random Forest, and XGBoost—demonstrated that diverse algorithmic perspectives can modestly improve overall accuracy to 34.5 % on unseen data. Although performance remains uneven across classes, the feature-importance analyses consistently highlighted a small subset of indicators (notably features 61, 55, 53, and 8) as critical levers for particular disaster categories. These insights not only validate the ensemble's predictive mechanics but also point the way toward more focused data collection and triage strategies—namely, augmenting under-represented classes that share these key indicators and prioritizing uncertain cases for human annotation. By adhering to fixed random seeds and transparent cross-validation, I have ensured full reproducibility. Moving forward, I anticipate that integrating targeted data augmentation and refining feature extraction will yield more balanced class recall and further elevate the model's real-world utility.

Resources

https://www.kaggle.com/competitions/datascience-4-competition/overview