Analyzing Community Sentiments: A Machine Learning Approach to Survey Responses on the Impact of Technology in San Antonio

Rani Misra, Kashfia Sharmin, Abigail Davis, Cheryl Chiu
IS 6713: Data Foundations
Dr. Nehir Tanyel Worheide

This project is aimed to classify survey results from the City of San Antonio's efforts to learn about the communities' feelings towards the impact of technology. 1000 comments from the survey were analyzed by two annotators as well as the researchers, and classified as Technology related or Public Safety related (the secondary category chosen by the researchers). Creating a machine learning algorithm can support community officials in highlighting areas of concern for public safety allowing intervention opportunities. It could also provide areas to improve the city through the use of technology and make San Antonio a more green and efficient city.

The initial guidelines created for this research highlighted different combinations of the two categories at hand. The Technology category was assigned to comments mentioning technology-related topics such as digital tools, data analytics, software, or e-governance. The Public Safety category was assigned to comments related to safety measures such as street lighting, public health, violence and crime prevention, or traffic safety. Comments that did not directly pertain to either category were to be annotated as 'Neither'.

In the pilot study, annotators labeled an initial set of 50 comments using the guidelines provided to them. To calculate the inter-annotator agreement, the aggregate of choices was first calculated. For example, in Comment #1, all three annotators chose 'Technology' and none chose 'Non-Technology'. In Numpy Array format, this was coded as [3,0] where the first digit represents how many chose 'Technology' and the second digit represents those that did not. The same concept was also applied to the Public Safety category. The combinations of these choices were then used to calculate the inter-annotator agreement. The Fleiss' Kappa method was utilized as it could be generalized to multiple annotators. A Fleiss' Kappa of 0.318 was calculated for the Technology category, which fell between the range, 0.21 – 0.4, thus illustrating that there was fair agreement amongst the annotators when it comes to that category. Similarly, a Fleiss' Kappa of 0.406 was calculated for the Public Safety category, which fell between the range, 0.41 – 0.6, thus illustrating that there was moderate agreement amongst the annotators when it comes to that category. Overall, the lack of strong agreement among annotators for both categories highlighted a need for more refined guidelines that clearly defined the expectations for classifications.

Thus, the guidelines were modified to clearly highlight what could be classified as Technology or Public Safety, and a new category of 'Both' was included for cases where a comment fell under both categories. The pilot study was conducted again with these new guidelines in mind and the resulting data was once again analyzed for inter-annotator agreement using the same methods as before. The Fleiss' Kappa for Technology was now calculated to be

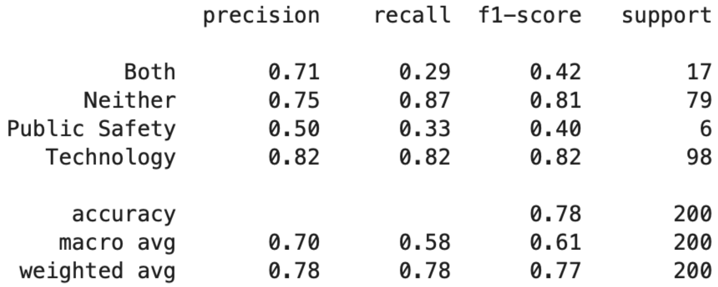
0.871, which fell between the range of 0.81 – 0.1 and illustrated that there was almost perfect agreement amongst the annotators when it comes to that category. A kappa of 0.655 was calculated for the Public Safety category, which fell between the range of 0.61 – 0.8 and illustrated that there was substantial agreement amongst the annotators when it comes to that category. These new Fleiss' Kappa calculations were much more stable and satisfactory to continue with for further analysis.

In the final stage of this research project, all 1000 comments generated from the survey results were annotated using the updated guidelines to form a model that could be used to classify the sentiments in future surveys of the same scope. In order to do this, the gold standard for each comment was first decided by the majority of responses across annotators and the researchers. For example, if two of the annotators agreed that a comment should be classified as 'Both', however one annotator did not agree, the majority choice for the classification was considered as the gold standard.

To begin processing the data for modeling, a CSV file was read in with columns 'Comment' and 'Gold Standard' which included the text data and classification, respectively. Then, missing values, NaN, were replaced with empty strings. Three features were extracted from the text for model training: number of fully capitalized words, number of exclamation points, and a binary indicator of repeated characters. Using CountVectorizer, the text was transformed into numerical features based on word frequency. The `stop_words` feature removed common words like "the," "is," and "and" as they typically don't add meaningful information. This was assigned to `X_test`, a matrix, where each row represented a comment, and each column was a word or token. The three features mentioned previously were extracted into a data frame called `X_numeric`. These were then combined into a complete feature matrix called `X_combined`, which could then be used for training the model. The column of Gold Standards was assigned as the target variable (y).

To begin model training and evaluation, the data was split into 80% training and 20% testing sets, with a random state of 42 to maintain reproducibility by controlling the randomness of the split. Then, a logistic regression model was fit to predict the 'Gold Standard' labels, and identify the relationships between the features and labels. The resulting trained model was then used to predict labels for the test set (X_test). Generating a classification report provided detailed insights into the performance of the model across all possible classes.

For the 'Both' category, 71% of the predictions of this class were correct (Precision: 0.71) and only 29% of actual instances of this class were correctly predicted (Recall: 0.29). The F1 score was low at 0.42 due to imbalanced precision and recall, indicating poor performance for this class. The Support value indicated that there were only 17 true instances of this class in the test data. For the 'Neither' category, 75% of the predictions of this class were correct (Precision: 0.75) and 87% of actual instances of this class were correctly predicted (Recall: 0.87). The F1 score was higher at 0.81, suggesting good performance for this class. The Support value indicated that there were 70 true instances of this class in the test data. For the 'Public Safety' category, only 50% of the predictions of this class were correct (Precision: 0.50) and 33% of actual instances of this class were correctly predicted (Recall: 0.33). The F1 score was poor at 0.40, likely due to the small number of samples. The Support indicated that there were only 6 true instances, making it a highly underrepresented class. Lastly, for the ‘Technology’ category, 82% of the predictions for this class were correct (Precision: 0.82) and 82% of actual instances of this class were correctly predicted (Recall: 0.82). The F1 score was excellent at 0.82. The Support value indicated that this was the largest class with 98 instances. The overall metrics indicate that the model correctly predicted 78% of the instances overall. The model’s performance varied significantly across classes, as reflected in the average metrics. The Macro Average precision of 0.70 indicates fairly accurate predictions overall, but the lower recall of 0.58 highlights challenges in identifying instances from minority classes, such as ‘Both’ and ‘Public Safety’. These underrepresented categories had insufficient data, leading to poor recall and a moderately low F1 score of 0.61, reflecting the challenges in handling minority classes. The average metrics weighted by the number of instances in each class (Weighted Avg) reflect the model’s overall good performance, heavily influenced by well-predicted classes like ‘Technology’ and ‘Neither’.

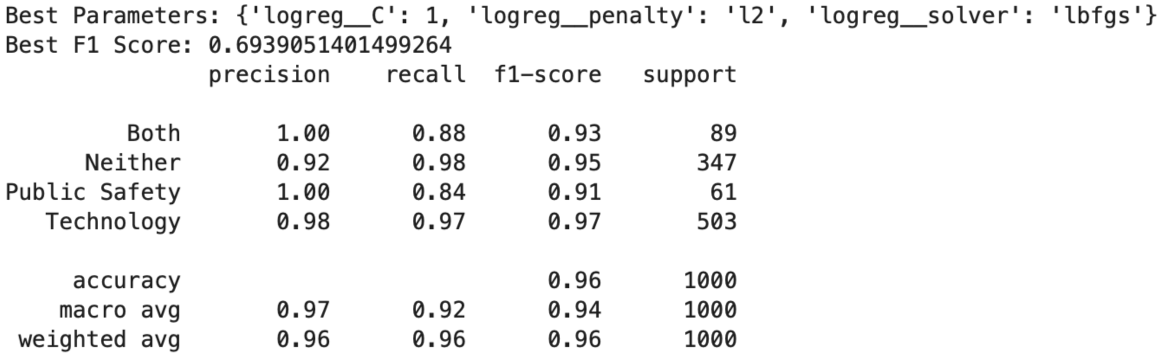


```
               precision    recall  f1-score   support

         Both       0.71      0.29      0.42        17
      Neither       0.75      0.87      0.81        79
Public Safety       0.50      0.33      0.40         6
   Technology       0.82      0.82      0.82        98

     accuracy                           0.78       200
    macro avg       0.70      0.58      0.61       200
 weighted avg       0.78      0.78      0.77       200
```

To further analyze the model, cross-validation was implemented to evaluate the model’s precision, recall, and F1 scores across 5 folds. A pipeline was created to combine the vectorizer and the model into a single object. A custom scorer function was created to calculate weighted averages of precision, recall, and F1-score. 5-fold cross validation was then conducted using the pipeline and custom scoring. The results showed that, on average, 71% of the predicted positive labels across all classes were correct (Precision: 0.71) and that the model did a reasonably good job of avoiding false positives, but there is still room for improvement. It correctly identified 71% of the actual positive labels across all classes and was moderately effective at identifying all relevant instances, but it missed some indicating false negatives. The F1-score balanced precision and recall, yielding an average score of 69%. While the model had decent precision and recall, their balance indicates it might struggle with either false positives or false negatives in certain cases.

To improve the model, hyperparameter tuning using GridSearchCV was used. Another pipeline was created, which combined the CountVectorizer and the logistic regression classifier to ensure consistent application of vectorization and modeling during hyperparameter tuning. Hyperparameters were defined in a grid to include multiple C values to control the regularization strength, a penalty of `l2` which indicates Ridge regularization, and applied optimization algorithms solvers such as `lbfgs` and `saga` which are aimed towards multi-classification problems and large-scale problems with multiple samples and features, respectively (Adam, 2023). The grid search was then set up to try all combinations of parameters defined in the hyperparameter grid, perform a 5-fold cross validation, and compute a weighted F1 score that considered class imbalances. The best F1 score is 0.694, indicating an improvement in balancing precision and recall across all classes during training. The best estimators from the grid search were then extracted to construct the best model and conduct predictions for the entire data set. Generating a classification report provided detailed insights into the performance of the best model across all possible classes.

For the ‘Both’ category, all predicted instances labeled were correct (Precision: 1.00) and 88% of true instances were correctly identified (Recall: 0.88). The F1 score was high at 0.92 due to the high balance between precision and recall. For the ‘Neither’ category, predictions were highly accurate (Precision: 0.92) and almost all true instances were detected (Recall: 0.98). The F1 score of 0.95 reflected the excellent precision and recall. For the ‘Public Safety’ category, precision was perfect (1.00) with some loss of recall (0.84). The F1 score dropped slightly to 0.91 due to the imbalanced support for this category. Lastly, for the ‘Technology’ category, precision was very high (0.98) and nearly all true instances were detected (Recall: 0.97). The F1 score was excellent at 0.97 for this majority class. The overall metrics indicate that 96% of predictions match the true labels. Macro Avg shows an incredible precision of 0.97, slightly lower recall of 0.92 due to underrepresented classes, and an F1 score of 0.94, which is fairly balanced across all classes. Weighted Avg shows the model performs consistently well with a value of 0.96 across all metrics, even with imbalanced data.



```
Best Parameters: {'logreg__C': 1, 'logreg__penalty': 'l2', 'logreg__solver': 'lbfgs'}
Best F1 Score: 0.6939051401499264
               precision    recall  f1-score   support

         Both       1.00      0.88      0.93        89
      Neither       0.92      0.98      0.95       347
Public Safety       1.00      0.84      0.91        61
   Technology       0.98      0.97      0.97       503

     accuracy                           0.96      1000
    macro avg       0.97      0.92      0.94      1000
 weighted avg       0.96      0.96      0.96      1000
```

While the current presented model has excellent performance, it can be further improved to handle class imbalance using complex techniques like Synthetic Minority Oversampling Technique (SMOTE) in which the minority class is oversampled and the majority class is undersampled to improve the receiving operating characteristic (ROC) that measures model performance and to boost recall for underrepresented classes (Chawla et al., 2002). The hyperparameters defined could also be further refined for a more meaningful impact on the model. Additional features could also be included along with those currently selected to improve the overall predictive ability of the model.

References

Adam, F. M. (2023). *Classification algorithms with their solver parameters.* Medium. Retrieved December 6, 2024, from https://medium.com/@fateemamohdadam2/classification-algorithms-with-their-solver-parameters-ce7828599611

Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 359-387. https://www.jair.org/index.php/jair/article/view/10302