Toronto

Capstone Project: Sentiment Analysis of Violation Level

Manan Soni[a]

Kunal Sindhu[a]

Dhairya Dave[a]

[a] *College of Professional Studies, Master of Professional Studies in Analytics.*

Subject: *ALY6140*

Introduction:

This data is based on the country's general expansion in the 21st century in areas of Technology, Infrastructure, Healthcare, and Finance. Over the past 30 years, the food business has been expanding. The population's health and well-being are essential given this fast increase. Every nation in the world has a food safety department to deal with this. The Boston Department of Inspectional Services' Health Division is the source of our dataset on which we are going to conduct a study. This division makes certain that all eating places adhere to hygienic practices and international requirements. They are in charge of regularly inspecting the food to spot hygienic issues, store maintenance issues, or illnesses brought on by a certain store or food. The legacy dataset that contains the data of individual examinations and outcomes is the one we'll be using.

Dimensions of the Dataset:

- There are around 660,000 rows and about 23 columns

a) These comprise owner, addresses, brand names, and licencing details (duration of validation)

b) Causes of violations

c) Suggestions for improvement, etc.

Unit of Analysis:

Conducted a inspection of food business by a person in charge and recorded multiple inspection if there is any violation. It was possible to record multiple violation of a single restaurants

Preprocessing of Data:

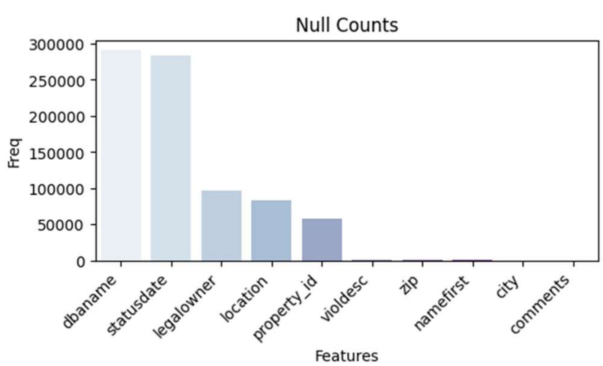

Figure 1, Null Counts by Columns

Initially, Data has null values which we counted according to figure 1. As our goal is sentiment analysis by comment and description received, there is hardly any other column to guess the unique of samples. So removed no comment as well as exact duplicate comments for a single restaurant.

```
In [56]: inspection_df = inspection_df.drop(inspection_df.index[inspection_df['comments'] == ' '])
         inspection_df = inspection_df.drop(inspection_df.index[inspection_df['comments'].duplicated()])
```

Figure 2. Removing duplicates using duplicated function

As shown in Fig.3, high frequency of restaurants exists which failed the inspection no matter having license active or not.

violstatus		Fail	Pass
licstatus			
Active	0.002400	0.969810	0.027789
Inactive	0.006025	0.957074	0.036902

Figure 3.Violation Status vs License Status

There are just 2 numeric values available which are licenseno and property_id. This data is large still consumes 60 MB to load into system. Out of 24 hardly 3 columns will be used for analysis at prediction/modelling phase.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 293267 entries, 0 to 727596
Data columns (total 26 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   businessname   293267 non-null  object
 1   dbaname        2678 non-null    object
 2   legalowner     196906 non-null  object
 3   namelast       293267 non-null  object
 4   namefirst      293237 non-null  object
 5   licenseno      293267 non-null  int64
 6   issdttm        293267 non-null  object
 7   expdttm        293267 non-null  object
 8   licstatus      293267 non-null  object
 9   licensecat     293267 non-null  object
 10  descript       293267 non-null  object
 11  result         293267 non-null  object
 12  resultdttm     293267 non-null  object
 13  violation      293267 non-null  object
 14  viollevel      293267 non-null  object
 15  violdesc       292423 non-null  object
 16  violdttm       293267 non-null  object
 17  violstatus     293267 non-null  object
 18  statusdate     9402 non-null    object
 19  comments       293266 non-null  object
 20  address        293267 non-null  object
 21  city           293259 non-null  object
 22  state          293267 non-null  object
 23  zip            293173 non-null  object
 24  property_id    235908 non-null  float64
 25  location       210409 non-null  object
dtypes: float64(1), int64(1), object(24)
memory usage: 60.4+ MB
```

Figure 4, Information of Data

Fig.5, Showes number of license categories with description which shows eating and drinking restaurants are more around 50% of samples.

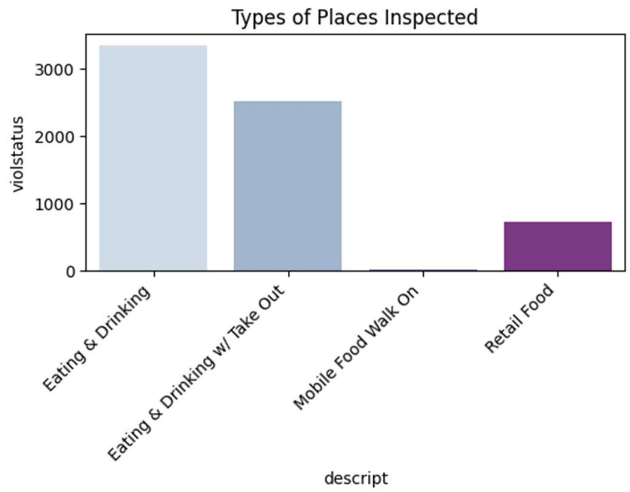

Figure 5, Types of Food places vs Count

Minor violation has a greater number of samples compared to higher and severe. Above 200k samples exists after cleaning.

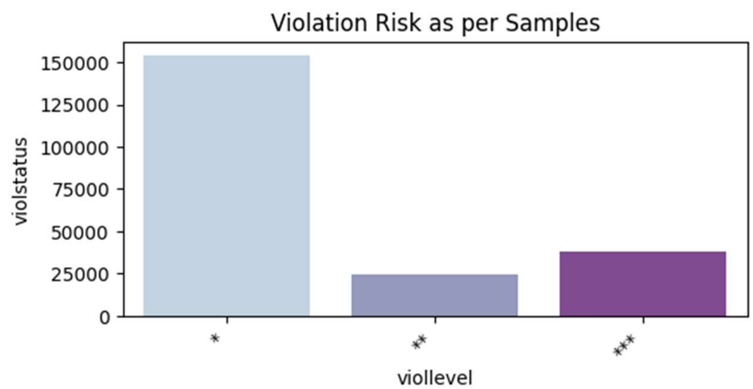

Figure 6, Violation Status vs Count

Figure 7, Legal Owners with Highest Violations Received

Mccoy Richard firm has received highest number of violations around 1050, Following that Gillette Cafeteria Edward C. Coleman is at second position. According to figure 6, these are top 10 legal owners who received this number of violations by person in charge.

Fig 8 demonstrates, number of places. Colour is according to density as we can see coastal area has more restaurants and places to eat compared to inner city areas.

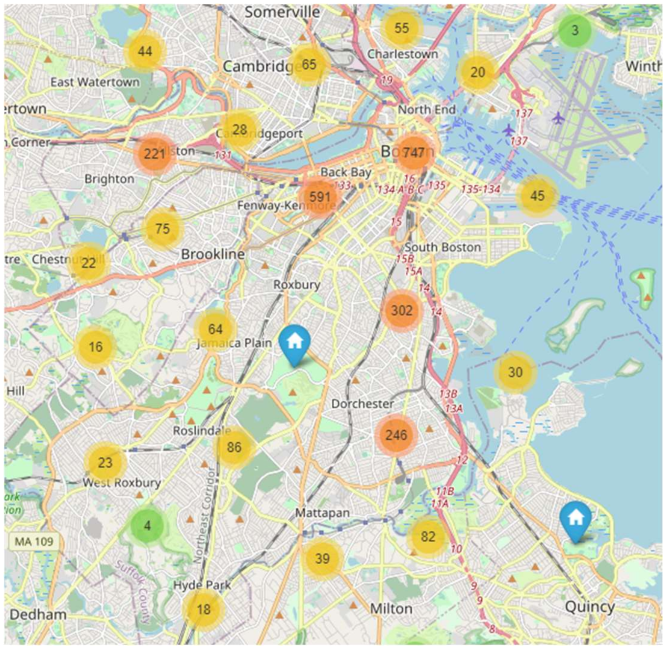

Figure 8, Map with Locations of Eatery Places

Figure 9 Words that has most frequencies

Classification using Bag-Of-Words

- Bags-of-words is a multiset of words disregards grammar. Sentences converts into dictionaries and having unique keys for each word and later it generates multiple words around the training words (As shown in Fig).
- This data will be used for Naive bayes classifier.
- Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle.
- It uses conditional probability method.
- Gradient boost classifier is an ensemble technique that generates decision trees sequentially having parameters like learning rate, loss.
- Unlike Random Forest it has learning which helps this algorithm to improve over the iterations.
- The confusion matrix basically shows recall and precision means how many predicted rights and wrong.
- According to Fig. 10, diagonal shows correctly classified samples where as other 0.85% to 3.79% values are false positives and negatives. This is same for Fig. 11.
- It's seeming like Gradient Boosting is better for 1 star violation whereas naïve bayes is better for other categories.

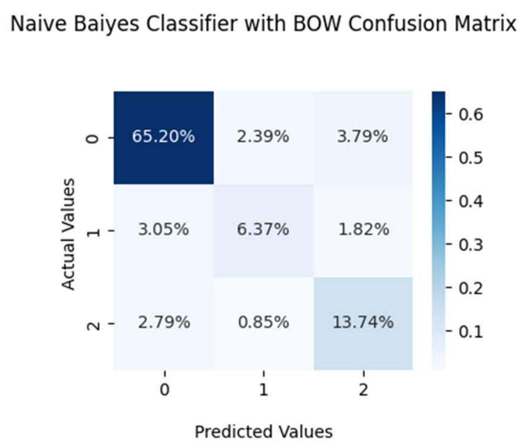

Figure 10 Naive Bayes Classifier

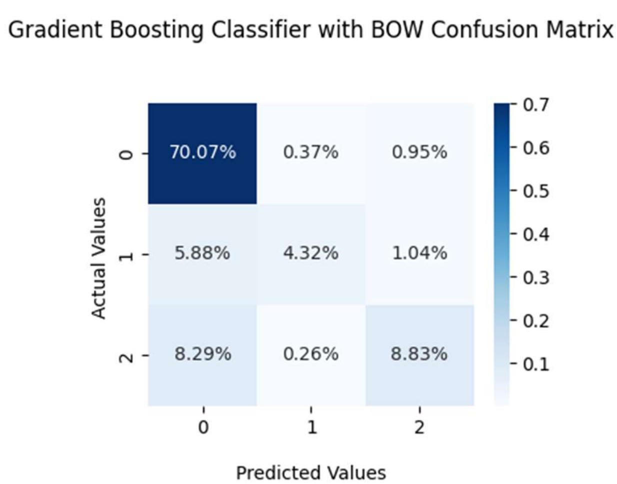

Figure 11 Gradient Boost Classifier

Classification using Artificial Neural Network

Implemented Sequential Neural Network which is considered as simple neural network with no direction with no explicit machine learning model usage. It was kind of complicated and difficult to implement a neural network.

```
model = tf.keras.Sequential([
    tf.keras.layers.Embedding(vocab_size, embedding_dim, input_length=max_length),
    tf.keras.layers.GlobalAveragePooling1D(),
    tf.keras.layers.Dense(16, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

Figure 12 Modal Definition

Activation functions such 'ReLU' and 'Sigmoid' is to calculate weights coming from previous hidden layers. There are pros and cons for each individual activation functions.
Loss function is mentioned as per TensorFlow documentation. We have 3 sentiments so need to go for categorical_crossentropy instead binary. Adam also employs an exponentially decaying average of past squared gradients in order to provide an adaptive learning rate.

```
num_epochs = 8
history = model.fit(training_padded, /
                    training_labels, /
                    epochs=num_epochs, /
                    validation_data=(testing_padded, testing_labels), verbose=2)
```

```
Epoch 1/8
3098/3098 - 127s - loss: 0.0000e+00 - accuracy: 0.7112 - val_loss: 0.0000e+00 - val_accuracy: 0.7099 - 127s/epoch - 41ms/step
```

Figure 13 Epoch Execution

Epoch is basically runs iteratively and gets lowest loss function and best accuracy. We received around 72% accuracy and val_accuracy is validation accuracy which shows difference between testing accuracy and training accuracy.
For Each epoch, changing the dimention from 50 to 200 was costing 2 hours of time to train 8 only.

```
sentence = ["Dark spots on chopping boards."]
sequences = tokenizer.texts_to_sequences(sentence)
padded = pad_sequences(sequences, maxlen=max_length, padding=padding_type, truncating=trunc_type)
y_pred3 = model.predict(padded)
print(f"For this comment '{sentence[0]}' it classify '{y_pred3[0][0]}' mean level 1 violation.")
```

```
1/1 [==============================] - 0s 11ms/step
For this comment 'Dark spots on chopping boards.' it classify '0.0' mean level 1 violation.
```

Figure 14 Prediction With ANN

"Dark spots on chopping boards" is minor violation and its out of dataset. It classified Correctly.

Conclusion

From the comment now we can predict which level of violation can classify. For more scope, using this we can get reviews and shows on map that which restaurant is not hygienic or provide rating for that as well. This way NLP is useful like for web searches, reviews, comments, clothing.

By this project, we learned how to process data for NLP. How vectors work with padding 0 and more. We learned the implementation and terms of artificial neural networks. Though it was difficult to implement at first but trial and error we learned it. Advance things like implementing TF-IDF and Word2Vec could be addon. These both helps to improve weights for neural networks.

References

1. “Seaborn Bar Plot With Sns.Barplot() - Examples for Beginners - MLK - Machine Learning Knowledge.” *MLK - Machine Learning Knowledge*, 24 Jan. 2021, machinelearningknowledge.ai/seaborn-bar-plot-with-sns-barplot-examples-for-beginners.
2. “Food Establishment Inspections - Food Establishment Inspections - Analyze Boston.” *Food Establishment Inspections - Food Establishment Inspections - Analyze Boston*, data.boston.gov/dataset/food-establishment-inspections/resource/4582bec6-2b4f-4f9e-bc55-cbaa73117f4c. Accessed 4 Dec. 2022.
3. “Stemming and Lemmatization Nlp at DuckDuckGo.” *Stemming and Lemmatization Nlp at DuckDuckGo*, duckduckgo.com/?q=stemming+and+lemmatization+nlp&atb=v340-1&ia=web. Accessed 4 Dec. 2022.