Model Results

1. STEPWISE: (this can be done in SAS EM)

   a. MISC 317 = 6.9%

   b. BIC 2193

   c. Number of Selected Features 38

   d. Screen shot not necessary

2. SKLEARN L1: (must be done in python)

   a. MISC 797 = 17.3%

   b. BIC 5934

   c. Number of Selected Features 57

   d. Penalty saga, c = 0.75

3. SKLEARN L2: (must be done in python)

   a. MISC 315 = 6.8%

   b. BIC 2368

   c. Number of Selected Features 57

   d. Penalty lbfgs, c = 0.75

4. SKLEARN ELASTICNET: (must be done in python)

   a. MISC 797 = 17.3%

   b. BIC 5933

   c. Number of Selected Features 57

   d. Penalty C = 0.75, saga

   e. L1 Ratio 0.25

5. GENETIC ALGORITHM SELECTION: (must be done in python using GA_sonar.py template

   a. MISC __304 = 6.6%____

   b. BIC ___2291____

   c. Number of Selected Features __49_____

   d. Goodness of Fit _____MISC_____

   e. Gen Zero Initialization Method _Features____

   f. Model _____sklearn_____

6. DECISION TREE : (this can be done in SAS EM)

   a. MISC __223 = 4.8%_____

   b. BIC ___NA_____

   c. Max Depth ____20___

   d. Min Leaf Size ____9___

   e. For SAS EM, please attach screen shot of decision tree property window

7. RANDOM FOREST : (this can be done in SAS EM)

   a. MISC ____194 =4.2%___

   b. BIC _____NA_____

   c. Max Depth ___24___

   d. Min Leaf Size ____8__

   e. For SAS EM, please attach screen shot of random forest property window

8. NEURAL NETWORK (SKLEARN) : (this can be done in SAS EM or Python)

   a. MISC ___298 = 6.5%_____

   b. BIC _____NA_____

   c. Hidden Layer Configuration ____10____

   d. For SAS EM, please attach screen shot of FNN property windows, main, network and optimization

9. NEURAL NETWORK (KERAS) :

   a. MISC ___0.076_____

   b. BIC ____NA____

   c. Hidden Layer Configuration
      _____10_____

Contents

Summary of Models

Logistic Regression Models

The logistic models generally performed as expected with the full models having a higher MISC than the penalized models which eliminate extraneous variables that do not contribute valuable information to the overall model. Of all the models ran, the logistic regression model utilizing genetic algorithms for variable selection performed the best.

Stepwise

Stepwise was done with critical limits set to 0.1 and with the backwards stepwise method, meaning variables were removed from the model and could not be added back in. The solver chosen was lbfgs. Based on the model results, it had a decent misclassification rate, however the stepwise model did have more trouble with identifying spam emails. It only selected 38 features for the model.

Lasso Regression (L1 Penalty)

Lasso Regression was tested utilizing the sklearn logistic regression model with the saga solver. Cross validation chose the best C to be equal to 0.75, however this model performed poorly compared to it's counterparts. It had a difficult time with determining emails that were not spam, resulting in a high MISC of 797 or 17.3%. The MISC for emails that weren't spam (class 0) was 22.5% which was significantly higher than what was seen in the stepwise regression model.

Ridge Regression (L2 Penalty)

The ridge regression performed better than the lasso penalty which is interesting to note as they both used the same limiting factor of 0.75 in their evaluation. Ridge had a lower overall MISC, and yet a higher MISC for identifying spam emails. This model performs better when required to identify non-spam emails. The MISC is similar to stepwise regression, however the ridge regression has a higher Bayesian Information Criterion (BIC) as seen in the summary page.

Elasticnet Regression (Elasticnet Penalty)

The Elasticnet regression performed similarly to the lasso regression utilizing the L1 penalty. It had a higher MISC, but performed poorly at identifying non-spam emails. Surprisingly, it's model metrics in addition to BIC were nearly the same as the lasso except the optimal l1_ratio was determined to be 0.25.

Genetic Algorithms

Using genetic algorithms to choose variables for the model was the most successful out of all the logistic regression models. It had the lowest MISC at 6.6%, but it chose 49 features instead of 38 like the stepwise model did. The increased number of features accounts for why the BIC is slightly higher than stepwise. It still had some trouble with correctly identifying spam emails, yet unlike stepwise it did this slightly better as seen. **Figure 1** gives an idea of how the overall fit compared with the number of features selected by the model.

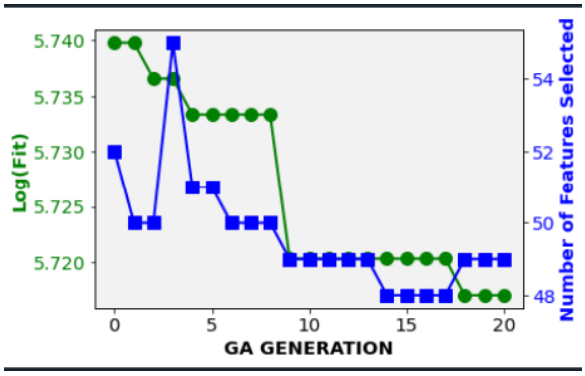

Figure 1: Genetic Algorithms for Logistic Regression Model.

Decision Tree Model

The decision tree model performs quite well as seen in the summary page. The model has a very low MISC compared to the logistic regression models making it an optimal model. Based on the model metrics, the chosen decision tree of a depth of 20 and a min leaf size of 9 does a reasonable job at classifying both spam and non-spam emails. The model perhaps fails more when it comes to spam emails yet not by much. Decision trees does give us an idea of feature importance as seen in **Figure 2**, and it is worthwhile to compare it to the features selected by genetic algorithms in **Figure 3**. The feature with the highest level of importance according to the decision tree model is C$. This tells us that emails with dollar signs are more likely to be spam. This begs the question of how the model deals with emails from banks as people are likely to receive vital transaction alerts via email in regards to their account.

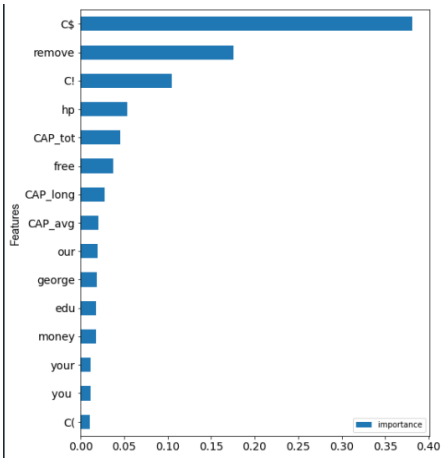

Figure 2: Decision Tree Feature Importance.

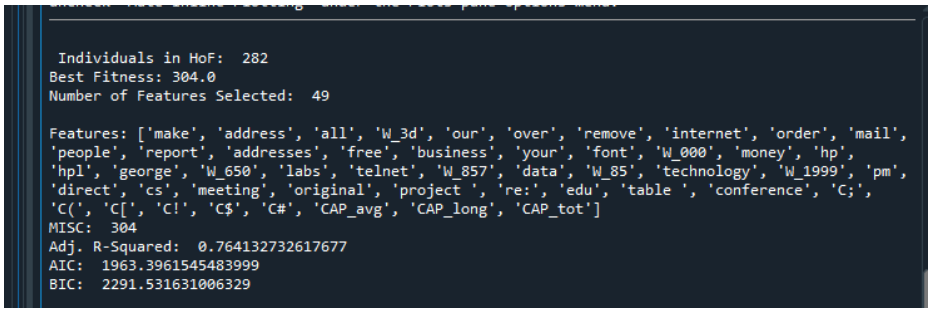

Figure 3: Genetic Algorithms Selected Features.

Random Forest Model

The results of our decision tree model enable one to tune the random forest model with the appropriate parameters. The number of trees was limited to 50 as per specifications, and cross validation indicated that a depth of 24 and a minimum leaf size of 8 yielded the best results. This closely aligns with the decision tree model, and may have been higher than expected due to the constraint on the number of trees. As seen on the summary page, the MISC for this model is quite low. What is shocking is the best maximum features it recommends: 10! The top 15 are shown in **Figure 4** below. This is quite a bit lower than what the logistic regression model with genetic algorithms indicated. These features of importance mirror the ones in the decision tree model in **Figure 2**. C$, C!, and remove are the top three features in decision tree and random forest. The one pitfall of random forest is the MISC for identifying spam emails as it is about 5% higher than the MISC for non-spam related emails.

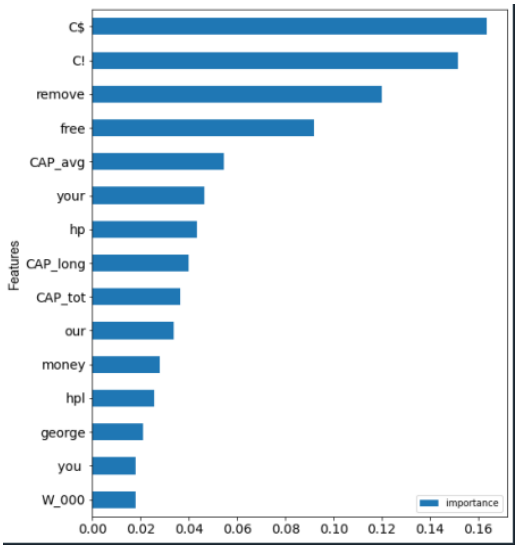

Figure 4: Random Forest Top 15 Features of Importance.

Neural Networks

Sklearn

Using the sklearn method generates a highly sensitive and precise model. The top features used in the configuration are the ones generated by the random forest model in **Figure 4**. By using these features and the sore list, a neural network model with a low MISC is generated at 6.5% with a 10 single layer 10 neuron configuration. The only downfall of this model is that it is not good at classifying spam emails which seems to be a recurrent theme. The sklearn neural network model does not actually perform better than the random forest or decision trees models which is unexpected. Adjusting parameters to choose the best configuration may improve the model performance.

Keras

Numerous configurations were attempted with Keras, but 30 epochs and a batch size of 2 was the easiest to run. Other variations either ran indefinitely or had higher MISC rates. Keras still chose a neural network with 10 neurons in a single hidden layer with 30 epochs and a batch size of 2. However, it's calculated MISC is the lowest out of all the models making it the most desirable model to utilize.

Recommendation

Out of all the models implemented, keras, random forest, and decision trees performed best. Knowing which models had the lowest MISC helps us to determine which might be the best overall model. The next step would be to do some form of validation for these three models to see if there's a risk of overfitting. After the models are assessed, a model can either be chosen or parameters tweaked in order to reduce overfitting if present. It would be ideal to modify the epoch and batch size in the keras model further, however there is a liability for the model to run for a significant amount of time. Dependent on the end goal of what this dataset will be utilized for in the future (i.e. determine spam in all future models, to continuously improve the model to predict when a spam email will be sent etc) will determine which model should be used. Understanding the business needs or goals for the model decides which model should be implemented. If a model is needed to predict or continuously identify spam emails over time after numerous training datasets, the keras neural network may be best. If this is a static data set that will not change with time, then the random forest or decision trees may be the most efficient. They require less processing time and are easier to implement. Still the next step in this process would be validation of the three models.