

# Machine Learning - Spam Email Identifier

## CSE 450 Naive Bayes Classifier By Landon Davis

### Data preprocessing:

This was fairly simple and explained in more detail below. The data set was in fairly good shape. I just had to make the features become integers in order to make it used in a naive bayes classifier model.

### Selection, Training, and use of a Machine Learning Model:

While developing this model there were a few things to make it able to classify and do it good enough to where it put an email into a spam or no-spam category. First I was able to take the data that it would train with and clean it. Making it either have a 1 for spam and a 0 for a real email. This allowed the model to have a category system in place.

I was able to train the model using Naive bayes algorithm and I decided to go with a multinomial classifier. This is because the multinomial classifier is used for classification with discrete features. The multinomial distribution normally requires integer feature counts. This is why the features were set to 1 and 0 when cleaning it.

### Interpretation of Results:

The best way to score and rate the spam emails to see if the model was correct was through using accuracy to measure the performance as the naive bayes classifier is best understood this way. The Model was able to train and see what words or phrases occurred often in spam emails. Then when given an email it would see if anything that led to a spam email would occur. My model has a 98% accuracy. The more information that it would train with would improve its performance.

### Communication of Results:

For each image at the bottom there is an array with three numbers a 1 is a spam message and a 0 is a normal message.

Feature 1: You can see the model has a 98% accuracy score when classifying a spam email.

```
[ ] predictions = model.predict(vectorizer.transform(X_test))
print("Accuracy:", 100 * sum(predictions == Y_test) / len(predictions), '%')

Accuracy: 98.99497487437186 %
```

Feature 2: These are random messages given to the model to see if it would predict correctly and it shows all three are spam emails.

```
model.predict(vectorizer.transform(
[
    "congratulations, you became today's lucky winner",
    "Join our club today and your first entry will be free",
    "Had your phone for more than a year",
]))

array([1, 1, 1])
```

Feature 3: These are random messages given to the model to see if it would predict correctly and it shows all three are normal emails.

```

model.predict(vectorizer.transform(
[
    "Thank you, ABC. Can you also share your LinkedIn profile? As you are a good at programming at pyhton, would be willing to see your personal/college projects.",
    "Hi y'all, We have a Job Openings in the positions of software engineer, IT officer at ABC Company.Kindly, send us your resume and the cover letter as soon as poss:",
    "Dear ABC, Congratulations! You have been selected as a Software Developer at XYZ Company. We were really happy to see your enthusiasm for this vision and mission.
])
)
array([0, 0, 0])

```

**Feature 4: More randomized messages given to model to try.**

```

[ ] model.predict(vectorizer.transform(
[
    "congratulations, you became today's lucky winner",
    "Can your please email me back with the information?",
    "Had your phone for more then a year",
])
)
array([1, 0, 1])

```

```

model.predict(vectorizer.transform(
[
    "Winner, You can win more money",
    "How are you doing today?",
    "Landon wants your phone number",
])
)
array([1, 0, 1])

```

## Ethical Implications:

The project and model where ethical

- **There are no ethical implications when organzing emails into spam or not with machine leaning.**
- **When it comes to the model its accurate in a spam email situation and can be trained to find more. Other then that this model is specifically good for this task or type of one.**
- **Its a fairly simple but intriguing model and algorithm to use and the understanding behind naive bayes has an important fuction.**

## APPENDIX A (PYTHON SCRIPT)

**Below is the Google Colab link to the notebook I used during this Project:**

<https://colab.research.google.com/gist/Landonjdavis36/de8b96fb922336905e6f268a07faf0a6/untitled10.ipynb>

**I would give this project a 2-All requirements were met. I have a working model and well explanation of the process, model, and results. I also have visualizations and code included.**