1. **For each honeypot server what attribute created the best probabilities?**

We ended up using the Naive Bayes algorithm for this project. We tested the algorithm with text, title, and author id features together and individually. We wanted to see which feature could most accurately predict the entity of a post. This would tell us what entities have in common, an possibly what the best way to detect a spam bot would be.This dataset consisted of bots posting or commenting on a server in different languages that included English, French, German, and others. In our feature extraction process we did not use any language-specific features, as over 75% of the content was in English. We focused on features such as author id, content, title, and language. Language was a two letter identifier stating what language the content was posted in. We extracted the top twenty words for content and title by entity and used a bag of words representation for these features in our naive bayes model. We expected content to be the best predictor, as bots within the same entities should be blogging about the same content. Using a naive bayes algorithm to calculate the maximum probability for a document class (in this case entity) we were able to predict with over 75% accuracy the entity of a post based on content for each of the honeypot servers. Author id, title, and language also produced good baseline accuracies but content was the best.

For the server ggjams Surprisingly title words were the worst features with a total accuracy of 65%. Author id was the next best feature with 85% accuracy. Text content words were the best features for separating the data with 87% accuracy. For ggjx we got a 60% accuracy for title, 67% accuracy for author id, and 82% accuracy for text. Lastly, for npcagent title provided a 76% accuracy, text provided a 93% accuracy, and author id provided an 86% accuracy.

2. **How do text features affect classification? What were the top words for each server?**

After extracting the best-predictor feature we wanted to find out what kind of content spam bots were posting. From the previous section we saw that users within the same entity were blogging about the same content. Finding out what this content is may help detect the likelihood of a user being a bot based on what they post. Below we found what the most popular words for each server. In this section we filtered out common stop words such as the, I, a, and, so on, as these do not tell us anything interesting or unique about the spambots. In order to get words that were relevant we used nltk stopwords to filter out these common English words. Again, we also focused on English content, as most of the content was posted in English.

The following text features were the most common across the three different servers for all entities. They give some insight as to what people were posting about one each server.

The top 20 words for the gjams server were:
also, get, one, make, well, de, like, time, may, use, could, would, protein, need, best, much, really, way, find, it's

The top words for the ggjx server were.
de, also, <a, la, well, make, get, protein, one, time, use, like, en, could, may, would, much, un, whey, carpet

The top words for the npcagent server were.
also, well, <a, <a, training, one, could, get, time, survivalist, like, waistline, actually, waist, make, use, may, midsection, corset, site

It appeared words like protein, whey, waistline, waist, and midsection were common amongst the servers which may point to some kind of health-related content as spam. More analysis needs to be done and it would be interesting to use a clustering algorithm to come up with possible topics posted in each server.

3. **Accuracy by entity, which entities were easier to predict and which were more difficult?**

Entities with more posts have a higher probability of appearing in both the training and test sets. The more examples that appear in the training set, the more likely the accuracy of the classifier will be exactly 1.0. The same goes for entities that are very sparse in the dataset - because there is not as much information to train on, these were harder to predict. Aside from the number of training samples, we wanted to see why certain entities were harder to predict than others, and if this had anything to do with content. We also had to keep in mind that these were the top 20 entities for each server, and in the future we could look at a larger range of entities. For this project we chose to narrow down our analysis to only the top 20.

For gjams, the following entities had the highest accuracy by entity: 1734 with 484 samples, 1730 with 32 samples, 1725 with 223, 1722 with 55 samples, 1714 with 39 samples, 1684 with samples. The following entities had the lowest accuracy by entity: 1732 with 23 samples, 1717 with 119 samples, and 1721 with 50 samples.

For ggjx, the following entities had the highest accuracy by entity: 3331 with 85 samples, 3326 with 55 samples, 3322 with 33 samples, 3293 with 51 samples, 3262 with 35 samples. The following entities had the lowest accuracy by entity: 3312 with 30 samples, 1011 with 66 samples, 3329 with 67 samples, and 3310 with 93 samples.

For npcagent, the following entities had the highest accuracy by entity: 1529 with 55 samples, 1528 with 10 samples, 1524 with 20 samples, 1523 with 30 samples, 1522 with 11 samples. The following entities had the lowest accuracy by entity: 1499 with 13 samples, 618 with 7 samples, 1530 with 17 samples, and 541 with 29 samples.

Although we were looking for common words between entity content that would show why some entities were harder to predict than others, we were not able to draw any insight from this. Although sample size and training/testing split can affect the classifier performance, there still may be other reasons for why some entities were harder to predict. In the future we would use more sophisticated natural language processing tools such as nltk to analyze the relationship between entities.

4. **Whether entities with more posts also have more hits per post.**

Tuples with format (postLength, hits)

Gjams data top 60
[(1429, 5312), (1380, 5333), (1358, 5312), (1323, 5387), (1307, 5312), (1303, 5310), (1295, 5312), (1290, 5387), (1273, 5451), (1270, 5333), (1268, 5387), (1259, 5333), (1227, 5402), (1225, 5402), (1219, 5333), (1198, 5387), (1176, 5451), (1163, 5310), (1133, 5312), (1091, 5310), (1087, 149), (1080, 5451), (1076, 27478), (1074, 5387), (1061, 313), (1060, 5402), (1057, 21550), (1040, 5451), (1030, 5333), (1026, 7713), (1019, 8513), (1010, 27825), (1008, 26933), (1003, 5402), (1002, 8254), (1002, 5451), (1000, 5170), (998, 20995), (996, 5406), (996, 5333), (995, 27964), (995, 5308), (989, 7124), (989, 5364), (988, 5457), (986, 5170), (985, 8218), (983, 8802), (982, 21550), (981, 9278), (980, 5402), (977, 21035), (976, 26933), (975, 20995), (975, 5333), (973, 26992), (972, 5457), (964, 5170), (963, 27825), (963, 26933)]

Ggjx data top 60
[(11414, 141517), (4216, 136488), (4192, 136482), (4185, 126998), (4173, 136896), (4163, 127567), (4156, 136919), (4127, 127567), (4108, 136896), (4091, 126998), (2654, 141573), (2313, 141552), (2258, 141573), (2049, 143665), (2049, 143665), (1976, 141552), (1967, 141573), (1946, 127567), (1940, 136896), (1880, 126998), (1423, 141552), (1384, 141850), (1346, 141560), (1318, 127447), (1317, 141978), (1310, 141517), (1289, 140535), (1269, 142037), (1266, 141656), (1258, 142950), (1256, 142213), (1248, 142542), (1241, 142176), (1238, 142133), (1230, 139424), (1226, 141749), (1221, 141573), (1208, 142113), (1190, 138687), (1180, 139072), (1173, 141560), (1173, 140812), (1131, 140979), (1116, 142030), (1110, 142030), (1108, 141049), (1101, 115636), (1097, 140535), (1074, 141573), (1066, 140535), (1055, 142129), (1054, 142433), (1053, 142030), (1039, 139072), (1033, 141004), (1027, 142993), (1015, 142026), (1011, 142239), (1008, 141957), (1008, 141870)]

Npcagent data top 60
[(1545, 37096), (1525, 36450), (1516, 37099), (1507, 37094), (1506, 36450), (1504, 37101), (1500, 37098), (1491, 36450), (1486, 37097), (1398, 37147), (1387, 36921), (1370, 36966), (1366, 33354), (1356, 33057), (1331, 33495), (1320, 36914), (1316, 3917), (1306, 36959), (1293, 36764), (1288, 32822), (1287, 33354), (1276, 36951), (1274, 32951), (1273, 3917), (1271, 3917), (1264, 36747), (1255, 3917), (1253, 37167), (1245, 3917), (1244, 3917), (1243, 33906), (1239, 37062), (1239, 32951), (1231, 33057), (1221, 33354), (1218, 33671), (1215,

32822), (1207, 33129), (1206, 37083), (1200, 3917), (1200, 3917), (1199, 3917), (1191, 33057), (1186, 33057), (1179, 36929), (1178, 36923), (1176, 35357), (1175, 32929), (1175, 3917), (1169, 33057), (1162, 34745), (1158, 36874), (1155, 18589), (1153, 16494), (1141, 36896), (1137, 35357), (1124, 33921), (1112, 33671), (1105, 36747), (1103, 36747)]

5. **Are many users for the same entities all from the same IP space?**

Not all entities have multiple IP addresses, but many entities have multiple usernames associated with them. This indicates that most entities are probably spammers with multiple users on each unique server. However, most entities are not large enough to contain many IP addresses from an IP address subnet space. The entities that do contain IP subnets also contain multiple IP subnets as well as some IP addresses that don't seem to be part of an IP subnet space. These seemingly random IP addresses could still be unique to an individual networking device or they could mean that these entities represent coordinated botnets of spamming computers. Entities with multiple IP address could also be posting spam from multiple networks.