**Analysis**

This project proved that classifying dad jokes is a very difficult task for humans, let alone computers. In hopes of building a classifier that utilized relevant information about a joke, instead of using a generic classifier such as the majority classifier, we attempted to build features from jokes and custom-make a classifier that used this information. Some features that we considered using were:

1. The presence of words with multiple meanings (such words are likely to be used as puns)
2. References to named entities (such words are likely to make references to general knowledge or pop culture)
3. Word count (we found from our EDA that lame jokes were likely to be the longest, and witty jokes were most likely to be concise)
4. A bag-of-words-type feature that rewarded classifications towards words that were likely to be found in witty, funny, or lame jokes and penalized classifications towards words that were not likely to be found in the three categories
5. Detecting the presence of homophones and homonyms (such words are also likely to be used as puns)
6. Detection of unexpected usages of a word's part of speech (a noun used as a verb, for example, may be indicative of its status as a pun in the joke)
7. Humor detection, including sarcasm

However, after some research, we found that some of these features were incredibly difficult to include in our project, as they are current topics of active research in the field of natural language processing that do not have a well-defined method to accomplish these tasks. In particular, although we considered using features 5-7, these specific features were not feasible to use in our own classifier for several reasons. Firstly, there were not any datasets that had reliable collections of homophones and homonyms, and creating one from scratch was entirely out of the scope of the project, as that alone is an effort that requires far more resources than the three of us can find or expend with our given constraints. Secondly, for feature 6, although it was possible to classify a given word's part of speech, it was another matter to classify that decision as "unexpected". For example, one of our witty jokes was "Little known fact, the mods of /r/Jokes are all actually Peruvian owls… I think they're Inca hoots…". The punchline here is "Inca hoots", a play on words for "in cahoots". The subtleties in this deceptively simple joke alone are enormous. "Inca hoots" will most likely be classified as a noun phrase (an adjective followed by a noun, or maybe erroneously as a noun followed by a verb), but the intent of the joke is to mean it as a prepositional phrase. Classifying the part of speech is already a task that state-of-the-art natural language processing might have trouble with, but to recognize that a simple two-word phrase is meant to serve as a completely different part of speech, and to recognize the phrase it is supposed to sound like phonetically, is practically impossible with current technology. Finally, despite current research into humor detection, these jokes transcend basic sentiment analysis and even sarcasm detection. Many of these jokes may seem like they have negative sentiment, but are actually not meant to be negative at all (i.e. “I bought the world’s worst thesaurus yesterday. Not only is it terrible, it’s terrible"), which was characterized as funny. Furthermore, the majority of the dad jokes did not even utilize sarcasm, instead relying more on wordplay, puns, and deadpan humor to land the punchline. Thus, for these reasons, we chose to only use features 1-4.

After training the model, we used a confusion matrix and found that the most common predicted label for all of the true labels was ‘witty,’ meaning that even for jokes whose true labels were ‘funny’ or ‘lame,’ the model most often classified these as ‘witty.’ This has a few possible explanations. First of all, ‘witty’ was the predominant class out of the three labels in the data: slightly over 50% of the data were classified as ‘witty,’ while ‘funny’ and ‘lame’ each made up around 25% of the data. Because of this, the model may have been biased toward ‘witty.’ Another possible reason is that in general, ‘witty’ jokes were the most structurally consistent, almost always being short with some play on words. However, the structure of ‘funny’ and ‘lame’ varied a lot more with some jokes being ‘lame’ because they were too long and others being ‘funny’ because they had a long context that allowed for a better punchline. Among the ‘funny’ and ‘lame’ jokes, there were also jokes that had similar structure to ‘witty’ jokes, which could have made the difference between classes ambiguous to the model. The features that we created, such as references to named entities and detection of homophones, are all features that can be indicative of ‘witty’ jokes, but their presence is not necessarily indicative of not being ‘funny’ or ‘lame.’ In other words, ‘witty’ jokes generally have qualities such as homophones and reference to named entities, but these qualities can also be present in ‘funny’ or ‘lame’ jokes. The qualities of a ‘witty’ joke are not mutually exclusive with the qualities of a ‘funny’ joke or a ‘lame’ joke. Because of this, when the model detected homophones, for example, these homophones could have been in a ‘funny’ joke but since homophones are more associated with ‘witty,’ the model would have classified the joke as ‘witty.’ Overall, ‘witty’ was both the most common class and the easiest class to describe in measurable characteristics, whereas ‘funny’ and ‘lame’ were more varied and ambiguous. As a result, the model predicted ‘witty’ most of the time.

Our model is certainly not free of bias and systematic mistakes. There were several jokes that we determined to be lame that had a few common words, including "Trump", "Covid-19", and "women". This is not to say any of these topics are necessarily lame, and in fact branding these words to have a negative connotation introduces bias. However, many jokes involving these topics were, frankly, made in bad taste.

Perhaps the most significant concern throughout the whole project was the fact that categorizing these dad jokes was a task that had many nuanced moving parts to it, and it often required making decisions based on subjective humor. What one person finds funny may not be funny to another. Two people may be equally fluent in English, but one person may not have seen a TV show that a joke references while the other person did. Detection of puns (and, therefore, usually the punchline) requires knowledge of pop culture, general information, parts of speech, homophones/homonyms, and multiple meanings of words. Throughout this project, we felt, time and time again, that jokes truly test a person's (or machine's) mastery of a natural language, and that this deceptively simple task actually demands the most subtle and advanced knowledge of a language.