New\_name\_classifyer Documentation

**Data structures:** The preliminary analysis relies on two dictionaries:

* Surnames: The first dictionary contains a list of surnames and the ethnic breakdowns (% black, % white, % hispanic, % asian, % americn indian). This data was mined from mongobay.com which has historical counts of last names by ethnicities for the top 50,000 American names.
* First names: A first name dictionary was made in a similar fashion which was based on ethnic counts of the top 1100 or so baby names in NYC from 2011. While the sample is definitely biased, it was the only data source of first names that provided statistical breakdowns by first name that we could find.

**Classification strategy:**

This model operates under the assumption that surnames are often more indicative of ethnicity than first names. Moreover, the model assumes that first names and last names are independent. The algorithm first searches for various combinations of the first name and last names in our databases. If there is no dominant ethnicity for either one, return the maximum probability. In the case that both names are found but neither one has a dominant ethnicity, we use a Naïve Bayesian Classifier to decide which ethnicity is most likely. If neither name is found, then we use pattern matching based on common name prefixes and suffixes to classify the last name. Lastly, if there were no found prefixes or suffixes, a fuzzy matching algorithm is used to find “closest” match for the surname. If the match is above 90%, then the probabilities from the match are used as the final classification.

**Naïve Bayesian Classifier**

This model uses both first name and last name to calculate the probability of being a certain ethnicity. This model is primarily used when both names are found but neither one has a dominant ethnicity that exceeds the chosen threshold. For every ethnicity, the posterior probability is calculated by taking the prior probability (chance of observing an ethnicity given the last name) and weighing it by new evidence – the first name’s ethnicity probability. However, instead of dealing with the conditional probabilities, this model uses odds ratios which greatly simplifies the calculations. The calculation essentially becomes:

and from there, the posterior probability can be easily calculated from the new odds. For example, suppose we had a first name F that was white 60% of the time and a last name L that was white 80% of the time. Then the odds are calculated with following formula:

The odds are computed for both first name and last name (OddsFirst and OddsLast). In this case, the OddsLast is analogous to the “prior probability” because the model starts by looking at the last name. The new evidence is the first name. Then, the total odds are calculated by taking the product of the name odds.

The last step is to convert the new odds to a probability. This can be done with the following equation which is derived by solving for P(White) in the odds equation above.

In this case, the odds ratios are 60:40 = 1.5 for first name and 80:20 = 4 for last name. The total odds is 6, so the posterior probability = **6/(1+6) = 6/7 = 86%**

**Phase 1: Simple Search**

We start by searching for the last name. If we can find it and its max. probability is above the chosen threshold (90% in our case) then simply return that ethnicity and its associated probability. If not, then there are three possible cases:

1. Surname found but no dominant ethnicity
   1. If this is the case, then leverage the first name. Search for it in the same fashion as above.
      1. If found and the max probability exceeds the threshold, return that max. ethnicity and associated probability.
      2. If found but no dominant ethnicity, then feed both names into Bayesian Classifier (explained above)
      3. If not found, then try Pattern Search (Phase 2)

**Part 2: Pattern Searching**

If neither first name nor last name is found in the dictionary, then the surname is searched for common prefixes and suffixes. This process is analogous to the first phase in many ways. We have two possible sources of information (prefix and suffix). Only now, instead of searching for matching first names and last names, we search for matching prefixes and suffixes. Thus, the logic from before is directly applicable here. If a prefix or suffix is found with a dominant ethnicity (over 90%), return that classification and that probability. If both a prefix and suffix are found but neither is above the threshold, the name is fed into the Bayesian classifier explained above.

*Mining the Data*

Lists of common prefixes and suffixes were scraped from Wikipedia and compiled into dictionaries.

*Calculating the Ethnic probabilities*

For each prefix and suffix, I needed to measure the likelihood of being an ethnicity given an observed prefix or suffix. To do this, I searched for each pattern in the list of 50,000 surnames. For each name with that pattern, I kept track of its probabilities and frequency (count). After the search was complete, I found the total occurrences and calculated overall probabilities as a weighted average of each name’s ethnic probabilities where the weight was the name’s frequency.

Calculating the probabilities

List of prefixes and list of suffixes

Search names for prefixes and suffixes

If found, follow same flow as simple search strategy

Bayesian classifier for prefixes and suffixes

**Part 4: Fuzzy Matching (Jaro Distance)**

In this phase of the algorithm, the name is compared to all other names inj the dictionary using what is called the Jaro distance. The jaro distance gives an intuitive idea of how similar strings are. (1.0 means perfect match, 0.0 means no matching characters.