

Allison Sliter

MP3, Horoscope Classification

Natural Language Processing, Winter 2015

Well that was fun!

For the **perceptron**:

All of the issues I ran into were conceptual in nature. I had a hard time for a little while organizing in my head precisely how to use all the data points and how to put them together in a way that would produce the results I expected. Once those went together, though, that part was relatively easy and I turned my attention to developing feature classes.

As for how to use it, I didn't change your API in the slightest, simply implementing how the function declarations required.

For the **classifier**:

I decided to use some of my "domain" knowledge about astrology to guide some of my possible feature set. My hypothesis is that astrologers swim in a sea that associates certain symbols and ideas with certain astrological signs and that there may then be a consistency with the way they develop horoscopes for each sign. My first passes were pretty primitive - I surmised there might be a relationship between the governing planets and the astrological signs, the names for the signs themselves, the meaning of the symbol that represents the astrological sign, or the "elements" that are associated with the signs.

But then I got an idea. There are certain ideas about the kinds of personalities different astrological signs cause - now, I don't know what kind of sign is supposed to be what kind of person, but under the hypothesis that there's a narrative astrologers use, I decided to add a bunch of "personality words" namely, I just googled "personality words" and got a list of adjectives to describe personalities. I also threw in some random "horoscope sounding" words.

I then loaded all the different word sets separately and ran the classifier over the powerset of that those sets. It took all night. I've included the raw console output of that but that lacks the key of which set worked and of course it's in chronological order. So I've also included a csv that shows in descending order the performance of each subset.

Here are the wordsets:

```
elements = set(["air", "fire", "water", "earth"])
```

```
planets = set(["sun", "moon", "mercury", "venus", "mars", "earth", "jupiter", "saturn"])
```

```
meanings = set(["ram", "bull", "twins", "crab", "lion", "virgin", "scale", "scales",  
"scorpion", "archer", "sea-goat", "waterbearer", "fish"])
```

```
personality = set(['Able', 'Accepting', 'Adventurous', 'Aggressive', 'Ambitious', 'Annoying',  
'Arrogant', 'Articulate', 'Athletic', 'Awkward', 'Boastful', 'Bold', 'Bossy', 'Brave', 'Bright',  
'Busy', 'Calm', 'Careful', 'Careless', 'Caring', 'Cautious', 'Cheerful', 'Clever', 'Clumsy',  
'Compassionate', 'Complex', 'Conceited', 'Confident', 'Considerate', 'Cooperative',  
'Courageous', 'Creative', 'Curious', 'Dainty', 'Daring', 'Dark', 'Defiant', 'Demanding',  
'Determined', 'Devout', 'Disagreeable', 'Disgruntled', 'Dreamer', 'Eager', 'Efficient',  
'Embarrassed', 'Energetic', 'Excited', 'Expert', 'Fair', 'Faithful', 'Fancy', 'Fighter', 'Forgiving',  
'Free', 'Friendly', 'Friendly', 'Frustrated', 'Fun-loving', 'Funny', 'Generous', 'Gentle', 'Giving',  
'Gorgeous', 'Gracious', 'Grouchy', 'Handsome', 'Happy', 'Hard-working', 'Helpful', 'Honest',  
'Hopeful', 'Humble', 'Humorous', 'Imaginative', 'Impulsive', 'Independent', 'Intelligent',  
'Inventive', 'Jealous', 'Joyful', 'Judgmental', 'Keen', 'Kind', 'Knowledgeable', 'Lazy', 'Leader',  
'Light', 'Light-hearted', 'Likeable', 'Lively', 'Lovable', 'Loving', 'Loyal', 'Manipulative',  
'Materialistic', 'Mature', 'Melancholy', 'Merry', 'Messy', 'Mischievous', 'Na\x3\xafve', 'Neat',  
'Nervous', 'Noisy', 'Obnoxious', 'Opinionated', 'Organized', 'Outgoing', 'Passive', 'Patient',  
'Patriotic', 'Perfectionist', 'Personable', 'Pitiful', 'Plain', 'Pleasant', 'Pleasing', 'Poor',  
'Popular', 'Pretty', 'Prim', 'Proper', 'Proud', 'Questioning', 'Quiet', 'Radical', 'Realistic',  
'Rebellious', 'Reflective', 'Relaxed', 'Reliable', 'Religious', 'Reserved', 'Respectful',  
'Responsible', 'Reverent', 'Rich', 'Rigid', 'Rude', 'Sad', 'Sarcastic', 'Self-confident', 'Self-  
conscious', 'Selfish', 'Sensible', 'Sensitive', 'Serious', 'Short', 'Shy', 'Silly', 'Simple', 'Simple-  
minded', 'Smart', 'Stable', 'Strong', 'Stubborn', 'Studious', 'Successful', 'Tall', 'Tantalizing',  
'Tender', 'Tense', 'Thoughtful', 'Thrilling', 'Timid', 'Tireless', 'Tolerant', 'Tough', 'Tricky',  
'Trusting', 'Ugly', 'Understanding', 'Unhappy', 'Unique', 'Unlucky', 'Unselfish', 'Vain', 'Warm',  
'Wild', 'Willing', 'Wise', 'Witty'])
```

```
signs = set(["Aries, Taurus, Gemini, Cancer, Leo, Virgo, Libra, Scorpio, Sagittarius, Aquarius,  
Pisces, Capricorn"])
```

```
horoscope_words = set(["ascend", "ascending", "house", "rising", "celestial"])
```

I also tokenized and stemmed and put into the text and stemmed the features, too - so I'd catch morphologically related instances of the words. I also put everything in uppercase as well.

The best performance was from 'elements', 'meanings', 'personality', 'horoscope_words' and that iteration of the classifier performed at 0.1395. You'd think maybe this means that "planets" didn't add much but using just planets for features was the 4th best performing iteration at 0.136. "Personality" seems to do pretty well, showing up in 4 of the top 5 subsets and performs on its own better than any other single set - at number 2 and a score of 0.137. Interestingly, using all the subsets performed near the bottom. More is not necessarily better.

