Abstract
The Author Profiling Task 2019 aims to identify the nature of Twitter accounts, detecting if the writer is a Bot or a Human being and in the last case, the gender of the account owner. Moreover, the task is proposed in two different languages: English and Spanish. For each instance of the problem and each language, we address the problem differently. We use an ensemble architecture to solve the Bot Detection for accounts that write in English and a single SVM for those who write in Spanish. For the Gender detection we used a single SVM architecture for both the languages, but we pre-process the tweets in a different way. Our final models achieve accuracy over the 90% in the bot detection task, while for the gender detection, of 84.17% and 77.61% respectively for the English and Spanish languages.

Dataset
For each account we have 100 tweets. All the tweets are raw, hence we can determine if it is a tweet or a Retweet. The length of the tweet can vary a lot, with a minimum of only one character and a maximum of more than 900.

Bot Detection Features
- **Emojis**
  The average number of emojis used in each tweet.
- **Web link**
  The average number of links shared in each tweet.
- **Hashtag**
  The average number of hashtags used.
- **Len of Tweets**
  The average length of the tweets.
- **Len of Retweets**
  The average length of the retweets.

Original Text
RT @BIBLE_Retweet: Great men are not always wise – Job 32:9

Distorted Text
I don’t know. Just making conversation with you, Morty. What do you think, H-H... know everything about everything?

Pre-Processing Gender
- **English**
  TweetTokenizer
  SnowballStemmer
- **Spanish**
  TweetTokenizer
  SpaCy Lemmatizer

Gender Detection Features
- **English**
  Words1-5 grams
- **Spanish**
  Words1-8 grams

Classifier Description
In the first layer, we have a single SVM and an AdaBoost instance. AdaBoost is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases. In the second layer, we have a Soft-Voting Classifier that ensemble the predictions of the previous layer. We sum up for each class the probability of being the right one as predicted by our classifiers, then we pick as final prediction, the class with the highest value.

Results

<table>
<thead>
<tr>
<th>Feature</th>
<th>EN score</th>
<th>ES score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emojis</td>
<td>71.05</td>
<td>70.65</td>
</tr>
<tr>
<td>Web link</td>
<td>79.19</td>
<td>87.93</td>
</tr>
<tr>
<td>Hashtag</td>
<td>76.53</td>
<td></td>
</tr>
<tr>
<td>Len of Tweet</td>
<td>59.59</td>
<td>53.36</td>
</tr>
<tr>
<td>Len of Retweet</td>
<td>86.12</td>
<td>81.41</td>
</tr>
<tr>
<td>Semicolons</td>
<td>64.27</td>
<td>55.21</td>
</tr>
<tr>
<td>Cosine Similarity</td>
<td>60.00</td>
<td>64.67</td>
</tr>
<tr>
<td>SA: Neutral</td>
<td>61.53</td>
<td>66.19</td>
</tr>
<tr>
<td>SA: Compound</td>
<td>51.37</td>
<td>68.36</td>
</tr>
<tr>
<td>Text Distortion</td>
<td>90.48</td>
<td>81.73</td>
</tr>
<tr>
<td>ALL</td>
<td>77.90</td>
<td>76.73</td>
</tr>
</tbody>
</table>

Bot Feature Results

Conclusion and Future Work
We develop 4 different classifiers, one for each problem subset. For the Bot detection of English written messages, we used an ensemble architecture where the AdaBoost outputs are ensemble by a soft-voting classifier. For each one of the other problems, we use a single fine-tuned SVM. We achieve excellent performances, especially in the bot detection task, where we record a score of about 95% on the English Dev and the English Final Test set. Globally our models perform better on the English accounts than the Spanish ones, so we believe that more work is needed to fill this gap.