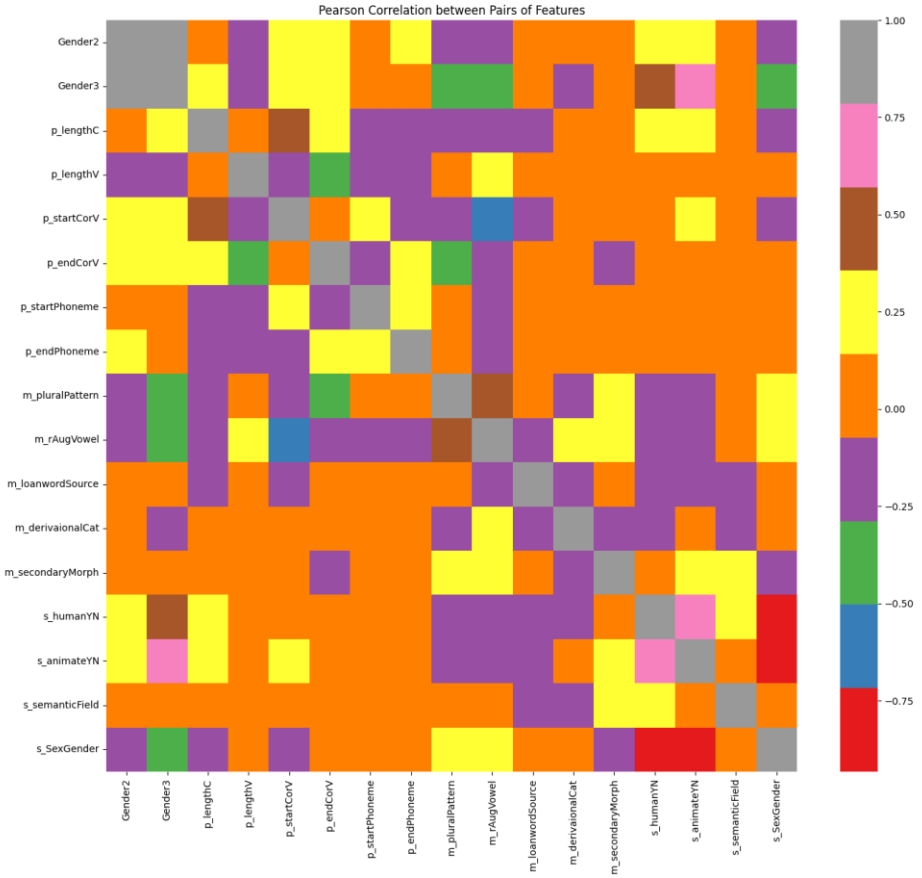
Explanation for choices in machine learning models and metrics

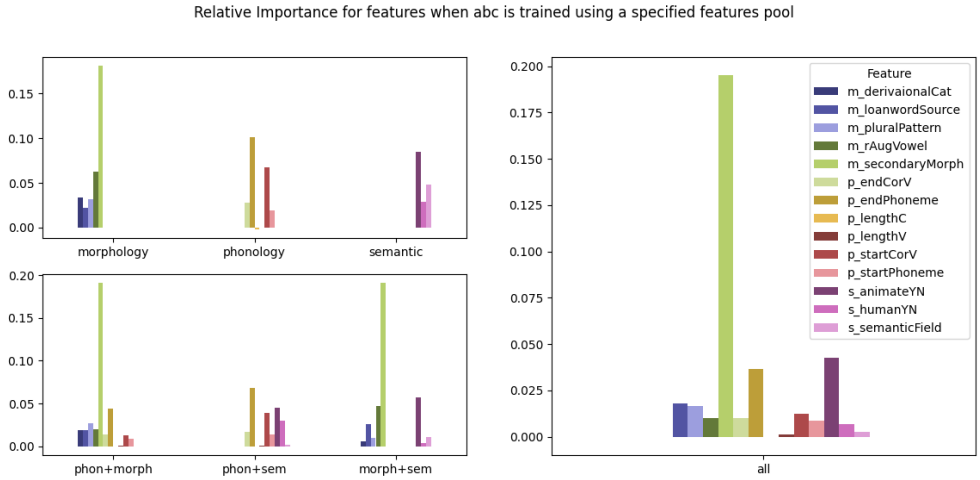
1. Using Label Encoder and Rationale on encoding categorical features by encoding categories as integers rather than having one hot encoded features.
    a. One concern is that, if a one-hot encoding is used then the number of dimensions in all would be 34 + 50 + 2 + 2 + 22 + 4 = 114 features instead of 12. With the increase in the number of dimensions (and not necessarily new information), performance of machine learning models can potentially go down, (training time will certainly go up)
    b. Second, is the correlation between the one-hot variables. For example: consider the Theme Initial Segment feature which has 25 distinct values. If we create 25 distinct features, then these features will be highly correlated. Since all data points have unique values, so all but one will have zeroes.
    c. By labelling features, a potential bias is an ordering that is introduced with the different values of the features. I think this can be a potential problem because Random Forest, Gradient Boosting and Adaboost generally uses comparison operation and by introducing such labels, clearly that would get biased.
    d. A suggestion that I have received is to use the one hot encoding as well and see if there any potential benefits in doing that. Results using One-Hot encoding provided at the end of document.
2. Correlation between features especially when we have an arbitrary labelling to the variables.
    a. First the Pearson correlation is defined as: $\frac{\sum(x_i y_i - \bar{x}\bar{y})}{\sqrt{\sum(x_i-\bar{x})^2 \sum(y_i-y)^2}}$ where $\bar{x}, \bar{y}$ are
    b. Regarding the meaning of correlation, and a specific example in this regard would be positive high correlation between ‘semantic field’ and ‘start letter’ would mean that a high value of ‘semantic field’ corresponds to a high value of ‘start letter’. A negative high correlation would mean the inverse: high value of ‘semantic field’ corresponds to a low value of ‘start letter’.
    c. Now, from the correlation plot ‘sexgen’ is highly negatively correlated with ‘human’ and ‘animate’. This is because all inanimate objects (s_lexiconAnimateYN == N) which are labelled as 1 don’t have the sexgen (s_lexiconSexGender) specified which was labelled as 4. This is why they are negatively correlated. Same goes with humanness.
3. Dropping variables: dropping sexgen but not human or animate.
    a. The correlation between human and animate is 0.765 whereas correlation between sexgen and animate was -0.93; between sexgen and human it was -0.73. I considered a cutoff magnitude of 0.8, and so removed sexgen. Again this is not cast in stone, so ‘human’ or ‘animate’ can be removed as well and the result can be seen.
4. On the choice of models:
    a. The models cover a range of complexity with support vector machines being the simplest and the ensemble model being most comprehensible.
    b. These models can happen high dimensional datasets and extract complex relationships between target and features.
    c. We avoided Neural Nets or Deep Learning because the data was not enough to train neural nets.
    d. Some other machine learning models that we could have tried were: Logistic Regression, Categorical Naïve Bayes. (we worked with logistic regression but don’t report as results were no very good, (we can put this in supplementary)), I couldn’t work with Categorical Naïve Bayes due to technical issues.
5. On the choice of AUC-ROC score:

a. AUC ROC score is said to handle class imbalance well. Though the dataset is only moderately imbalanced (75% masculine; 25% feminine), it is good to have a metric which can handle class imbalance.
b. AUC ROC score is a summarization of confusion matrices at different cutoffs and so reports the model performance across cutoffs.
c. We also report the Brier Loss Score which covers the limitation of AUC ROC score that it is independent of the probability magnitude values.

6. On ranking models:
    a. Here I mean that I am evaluating the models and then arranging them based on the AUC ROC score. A 5 fold CV repeated 20 times is basically the data is partitioned into 5 subsets. 4 subsets are used to train the model and then tested on the 5th set and the AUC ROC score is calculated. This is repeated 20 times and the mean AUC ROC score from the 20 runs is used to rank the models.
7. Feature Importance
    a. I have modified the code and also created new figures which are compatible. They are there in github.

Correlation Figure with new labels



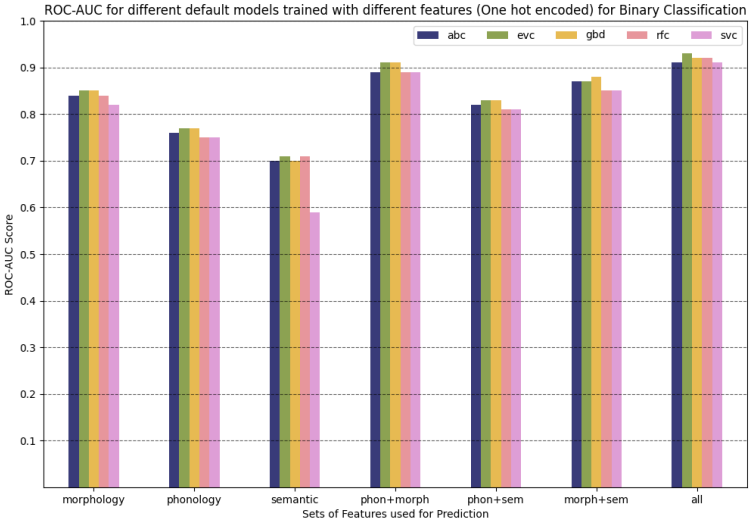


One Hot Encoding

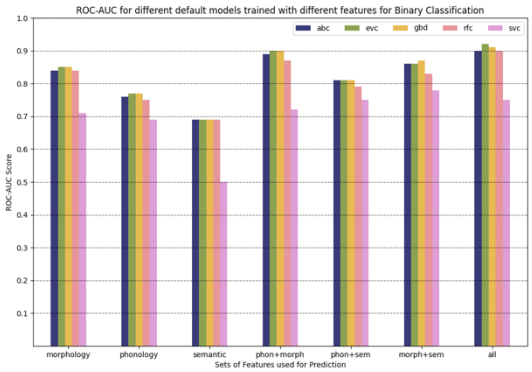
One hot Encoding was performed besides labelling for the categorical features. Generally, under one hot encoding, expansion in the number of columns take with each column (being a 0 or 1) denotes whether the data point has that particular value or not. So for example, m_PluralPattern : (Internal, External, Mixed, Internal); the encoding would correspond to the values: [Internal, External, Mixed] and the encoding looks like: ([1, 0, 0], [0, 1, 0], [0, 0, 1], [1, 0, 0]). So the three values expand to give three columns.

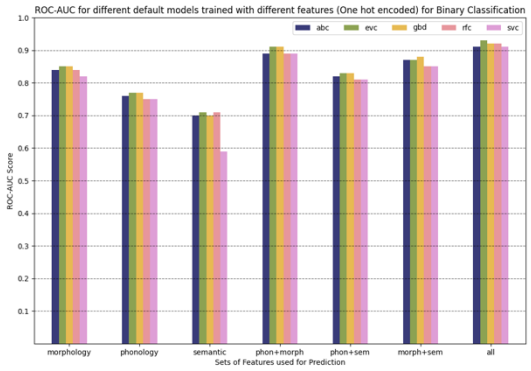
Now applying One Hot Encoding to the categorical features of dataset, the number of columns expand to 121.

The ROC-AUC metric for cross validation across different sets of features are as follows:



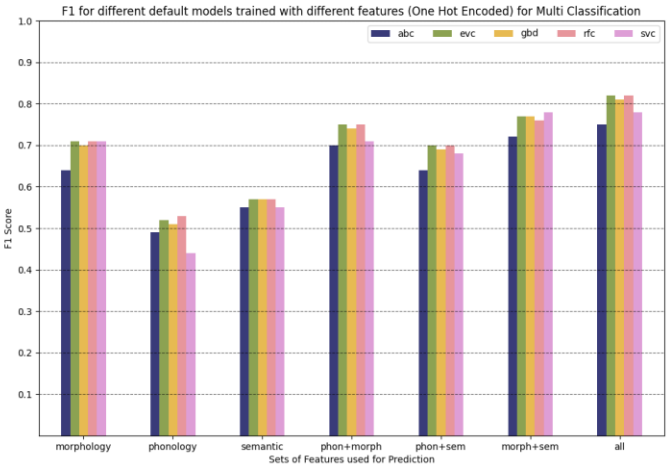
A highlight in the results is the better performance for Support Vector Machines. They now are very comparable in performance to other models as compared to when the features were being labelled. Also, the performance of other models seems to remain the same across the different set of features.



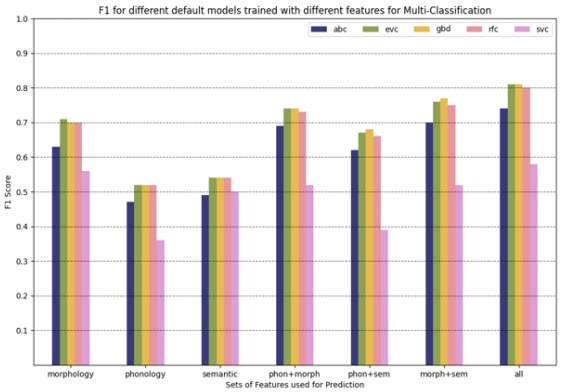


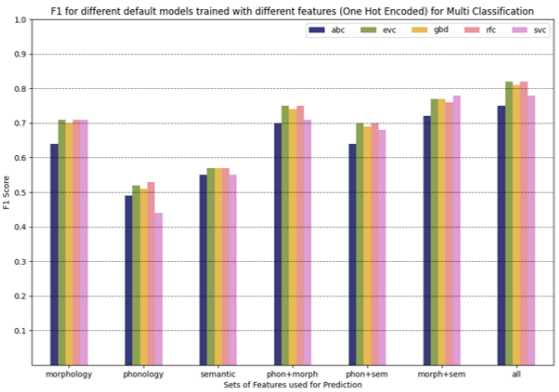
Results for Binary Classification with two different feature encoding. Left side is simple labelling, right side is one hot encoding.

When considering multi-classification (masculine, feminine and neutral), the F1 metric is reported as follows:



Again, there is a surge in the performance of support vectors with one hot encoding as compared to labelling. But not in the other models.

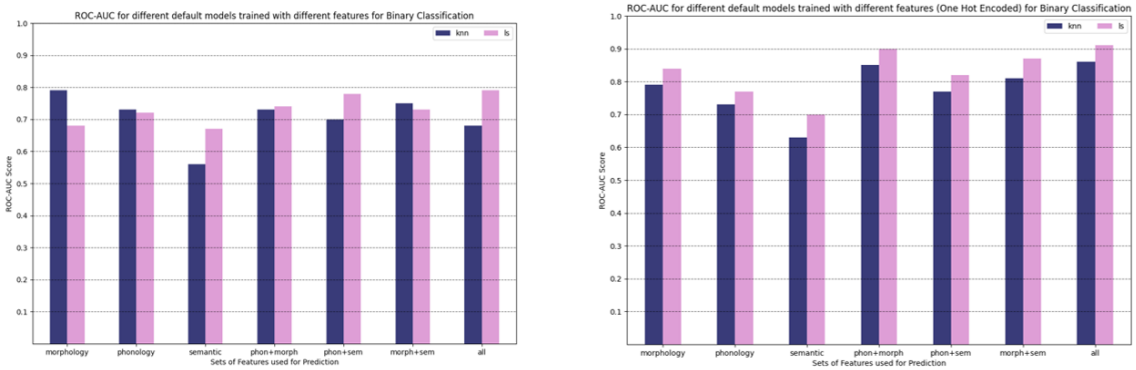




Results for Ternary Classification (Masculine, Feminine or Neutral) with two different feature encoding. Left side is simple labelling, right side is one hot encoding.

Supplementary Classification using Logistic Regression, K-nearest neighbor

Using two more models: Logistic Regression (with L2 Regularization) and K-Nearest Neighbor (number of neighbors: 5) for binary classification. The performance of the two model jumps on moving from simple labelling to one hot encoding. On using one-hot encoding the performance of Logistic Regression is comparable to more complicated models like Random Forest and Gradient Boosting. This is clearly very interesting as logistic regression is a much simpler model as compared to Random Forests.



Results for Ternary Classification (Masculine, Feminine or Neutral) with two different feature encoding. Left side is simple labelling, right side is one hot encoding.