**Predicting Gender from OKCupid Profiles**

Ashish, Yiqing Zhu, Peter C. Huang, Xingshuo Chen, Haoran Lei

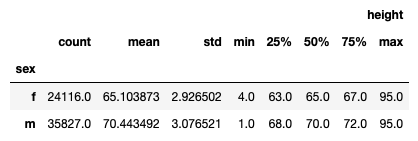
**Introduction** Online dating has long been a staple of the internet. After all, the internet made it so easy to connect and meet new people. However, all chats are NOT done face-to-face and present a new set of challenges for protecting consumers. One famous trick that trolls and scammers often use is called "catfishing," where a user pretends to be the opposite gender and misleads their partner into a false sense of security, leading to scamming or worse. Our final report assesses different machine learning-based models to detect gender based on users' data. Such models include traditional machine learning models such as logistic regression, Naive Bayes, and K-means clustering. On the other hand, we also have a suite of neural network-based models in fully-connected neural nets and bi-directional long-short term memory (Bi-LSTM) neural nets. The goal is to evaluate our suite of models and determine methods that would best allow dating website providers with insight to potentially malicious users.

**Dataset** The dataset is a .csv available on GitHub made freely public by OkCupid. The entries are responses to 31 questions by 59946 people within 25 miles of San Francisco with at least one profile picture online during 2011-2012, all of whom were over 18 (at least according to their profiles). The ages have a range of 18-69 with a mean of 32.3 and a standard deviation of 9.4. This sample is not representative of the population at large. We're only considering 10,000 users (datapoints) when encoding the fully connected neural network data as it doesn't fit into the memory if it's more than that. The dataset was split up into training and test sets of 8000 and 2000 entries, respectively, by random choice.

**Summary of Preliminary Results and Challenges** The dataset collected the volunteer’s general information such as age, height, and marital status, health information such as body type, do the diet and do they smoke, career information such as education, job, and income, the character information about pets and lifestyle, as well as the login record which consist of the location and last online time. The data set also contains nine essays from each volunteer (JSE\_OkCupid, 2015).

From the first five records of the dataset, we can see that each data point contains 31 features: numerical attributes and categorical attributes. Moreover, some values are missing (e.g., NaN), and some of the values have white space, quotation marks, or commas that we want to eliminate in the data analysis process. In particular, the essay features are more complicated than others, which has common string-delimiting symbols such as colon, period, and quotation marks and has some HTML elements such as the line break tag <br>. The complex data types are tricky, and we need to figure out.

In general, males tend to have a higher height than females, predicting users’ gender based on their height. To verify the relationship between gender and height, the mean, maximum, minimum, and standard deviation of the observed data point was calculated.

Figure 1

As we can see, the dataset has 24116 female volunteers and 35827 male volunteers with 65.10 inches of the mean value of height and 70.44 inches of the mean value of height, respectively, which demonstrate our hypothesis. However, the average heights are not faultless. We can easily find out from the observed dataset that some values are like a minimum height (e.g., 4.0 inches for females and 1.0 inch for males) and the maximum height (e.g., 95.0 inches for both females and males) are improper. We can fix them using the lower and upper limit for both genders as "three" standard deviations from mean value as height and replacing all the other values as not available. Therefore, for females, the range will be (56.32,73.88), and for males, the range will be (61.21,79.67).

What's more, we find out that 11% of volunteers do not respond to their educational information. More than 48400 volunteers reject answering their income, which means we lost 80% of income information approximately in our dataset. We may consider dropping the income column since it has the least useful information with fewer responses. For the feature of body type, the majority of answers are "average," "fit," and "athletic," which makes up 65.4% of the overall dataset. However, these words are more neutral and ambiguous; it's hard to conclude whether the volunteer is female or male. By contrast, the answers "skinny," "jacked," and "curvy" are more valuable for gender prediction. However, these answers only account for a tiny portion of the dataset, and we need to consider how to handle this problem.

**Preprocessing:**

**Encoding for Logistic, Random Forest and Decision Tree:**

We perform one-hot encoding on all the categorical attributes: body\_type, diet, drinks, drugs, education, ethnicity, job, orientation, pets, religion, sign, smokes, speaks, status, and sex, which generates 2413 features in total. For the continuous value: height and age, we cut those values into five equal-width bins. We randomly sample 10000 data points from the preprocessed dataset as the full dataset and take 80% of the entire dataset as the training set, the rest 20% as the test set.

**K-means:**

Due to a large number of unique words (65k in total) found in the essays in our dataset and the fact that males and females differ in their use of part-of-speech(POS) in writing according to a recent study in psychology. We lowercase all text, removed all accents from strings, all stop words, all blocks of digits, all string punctuation (!”#$%&’()\*+,-./:;<=>?@[]^\_`{|}~), all-white space between words, all instances of HTML breaks, and replaced unassigned values with empty spaces.

Then in each iteration of the 10-fold cross-validation, we took the following steps:

1. We tokenized each word in the cleaned essays into POS with the standard tokenizer from nltk package for both the training set and test set. We then found the list of unique tokens that occurred in the training set.
2. For both the training set and test set, we then counted the frequency of each token for each person (if a certain token is present in the above list but is absent for a person from the test set, its frequency is 0) and treated a token as a feature of that person and the frequency of this token as its value.
3. For both the training set and test set, we kept the POSes for each person as features and discarded the original essays.
4. We standardized the features and applied PCA on those features created in the above steps with a 95% variance contribution threshold. Then we projected the test set onto the same PCA space as the training set.

**Naïve Bayes:** We used the *sklearn.naive\_bayes.MultinomialNB* as our model for Naïve Bayes. The multinomial Naive Bayes classifier is suitable for classification with discrete features and therefore is an ideal Naïve Bayes model for text analysis. The model typically requires integer feature counts. We preprocessed the essays by *CountVectorizer()* function in sklearn, which will transfer the words into vectors of integers.

**Deep Learning:**

We removed all HTML break instances for preprocessing the essays for deep learning, removed new lines, and converted all words to lowercase. Also, we removed all punctuation except for a single quotation mark. This is because we wanted to preserve conjunctions such as “ I’m “ and “ she’ll. “ An example of our preprocessing is shown below:

**Original: Post Cleaning:**

I come to San Francisco by way of New York, where i was i come to san francisco by way of new york where i   
born, raised and went to school. I think this accounts for my was born raised and went to school i think this  
occasionally blunt, no b\*\*\*\*\*\* attitude, although the bay accounts for my occasionally blunt no b\*\*\*\*\*\*  
area has been hard at work mellowing me out. (some say attitude although the bay area has been hard at work  
it's even been successful)<br /> <br /> mellowing me some say it's even been successful

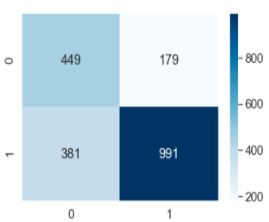
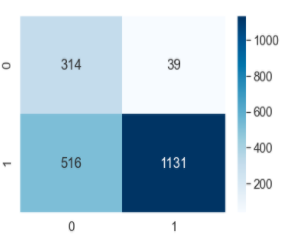
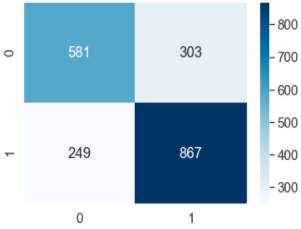
**Models:**

A total of seven different models were trained on the data. Three of these models(Logistic, Random Forest, and Decision Trees) were only performed on the data's attributes, and the nine essays were excluded. Additionally, four more models were trained using the essays from the user's profiles. The explanation and the results of the models are as follows:

**Logistic regression** We apply the logistic regression model on all the categorical attributes to predict whether the user is female(1) or male (0). We use gradient descent to maximize the log-likelihood; however, merely applying the parameter values that lead to maximum likelihood function value in the training dataset may result in overfitting. We then use the L2 regularization in the scoring function to penalize complex models, and the stop criteria of the optimization are set by the number of the maximum iteration, 500. The tolerance for stopping criteria is 1e-6.  
  
**Parameter Tuning:**   
 For our logistic regression model, we also tried to tune the parameters. We used a randomized search over different parameter values for a maximum number of iterations, the inverse of regularization strength, i.e., 'C' and other optimization algorithms including 'liblinear,' 'sag,' 'saga,' 'lbfgs' solvers from the 'sklearn' library. Despite using different parameter values, we observed that the performance of our model stayed consistent. Therefore, parameter tuning did not affect the accuracies of our model much. The reported accuracy values were when we tested our model with the default parameters for the optimization algorithm and 'C.' value with a maximum iteration limit of 500.

**Random Forest & Decision Tree:** A random forest is made by training decision trees on random subsets of the data and averaging their predictions. Random forests help avoid the overfitting that decision trees are prone to. We used 70 estimators (70 decision trees were trained on 70 random subsets of the data) and required a minimum of 25 data points per tree leaf to help prevent overfitting. Splitting the tree further seemed to overfit the data.

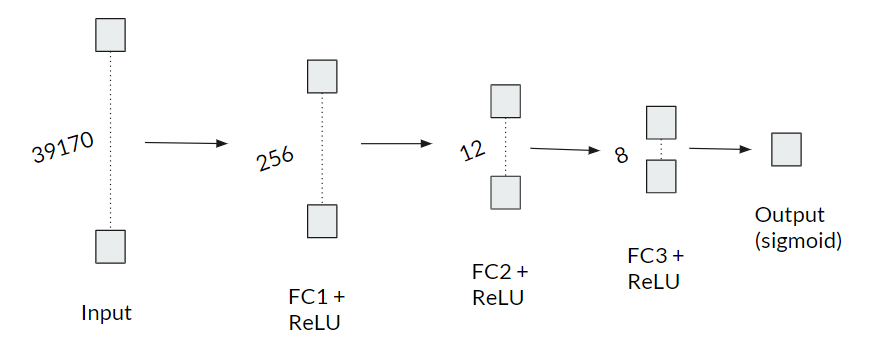
**Parameter tuning:**   
 We used ten-fold cross-validation to search for the best value for tuning hyperparameter—parameters like decision criterion, max\_depth, min\_sample\_split, etc. To get the most straightforward set of hyperparameters, we used the Grid Search method. In the Grid Search, all the mixtures of hyperparameters combinations will pass through one by one into the model and check each model's score. It gives us a set of hyperparameters, which gives the best score.

  
 Confusion Matrix for Logistic Regression, Random Forest and Decision Trees

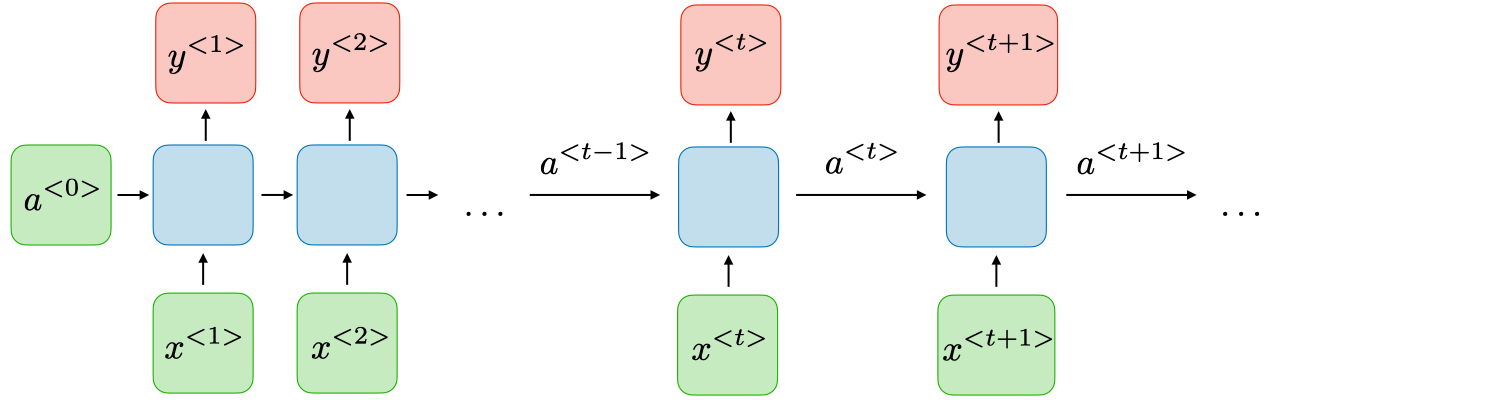
**Naive Bayes:** We trained the multinomial Naive Bayes model on ten essays separately to compare the performance with respect to the essays’ dataset. We also tuned the hyperparameter “ alpha” to get the best model. For each training, we conducted ten fold cross-validation to calculate the training and testing error.

**K-means:**  
 In each iteration of the 10-fold cross-validation, we fitted a k-means model on our training set using methods from the sklearn package. We first determined the optimal K value at each sample fraction. In each iteration of the 10-fold cross-validation, we evaluated the ‘elbow’ of each of the ten WC-SSD vs. k plots and determined the best k of that sample fraction by repeating this procedure, and compared the plots in the rest of the iterations. We then clustered our training set again using the best k. For each resulting cluster, we assigned the same predicted label of either male (1) or female (0) to all the data points in that cluster based on the ratio of male and female according to the true label of these data points in this cluster (if male >= 50% then predicted label = 1). We then fitted the test data to the model, assigned a predicted label to a data point in the test set according to which cluster it belongs to.

**Fully connected model:**  
 Our first neural network model is a simple fully connected network. A picture of the model is shown below

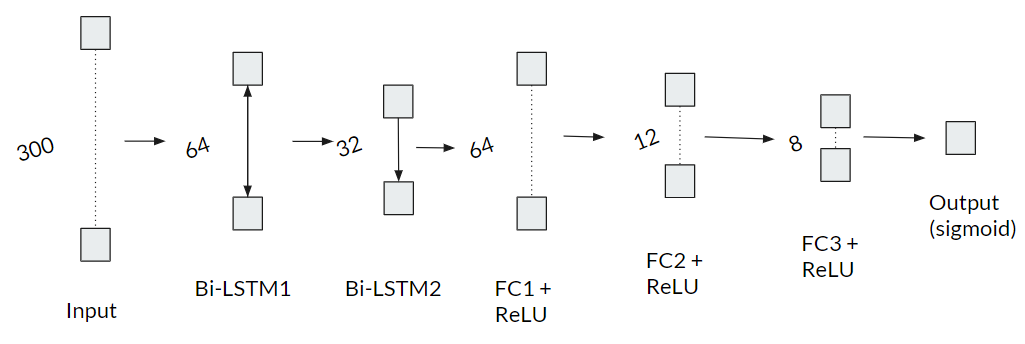
  
The model consists of 39170 input features, three layers of 256, 12, and 8 neurons, respectively, each augmented by Rectified Linear Units (ReLU) for nonlinearity. The final layer is a singular output neuron that utilizes sigmoid nonlinearity for binary output.

**Bidirectional-LSTM model:** The second type of model we used is a long-short term memory model which is based on recurrent neural nets. The idea is that each word can be encoded as an integer and is fed word by word into the model at different time steps. The result of each time step is then fed into the next time step as an additional input.



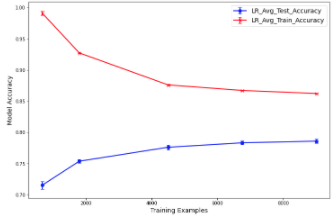
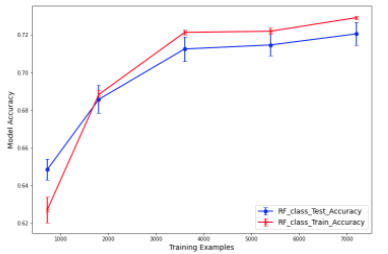
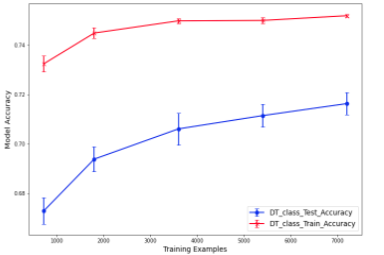
The image above demonstrates this model. a<0> is initialized to 0’s, and alongside x<1>, is fed to the first recurrent unit (visualized as the leftmost blue box). The recurrent unit generates y<1> and a<1> (not shown). a<1> and x<2> are fed into the second recurrent unit and so on. This process repeats for a certain fixed number of inputs. As one can see, normally, this generates a number of outputs equivalent to the number of inputs. However, this is easily changed to any number of outputs by either removing extra outputs (if we want fewer outputs) or adding additional recurrent units with no input other than from previous recurrent units (if we want more outputs). Mathematically, we can describe this model with the following equations

g(x) represents our non-linear activation function. Waa , Wax , and Wya are the weight matrices corresponding to the previous activation, current input, and current activation respectively. The b’s are biases.   
  
LSTM networks augment this architecture by introducing the notion of the update, forget, and output gates. These gates help the network learn when certain inputs are no longer relevant to a sentence's context. For example, if we have the sentence "The cat, which already ate…, was full, but…", the context of cat should extend to "was full" but not past "but." So the network must "remember" the word "cat" until the second comma. Bi-directional implies that not only do we take context from the past, but we also take context from the future.   
  
Our Bi-directional LSTM model is shown below. Each input has 300 input words fed into 2 Bi-LSTM units of output sizes 64 and 32. These outputs are then fed into a fully connected network and are finally binarized using an output sigmoid neuron.



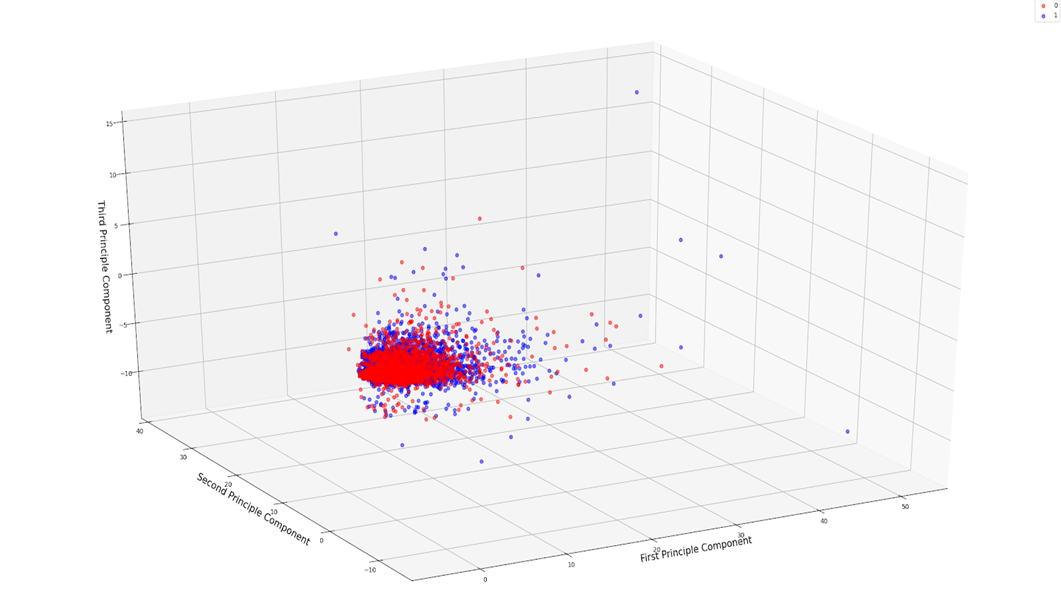
**Results:** We use 10-fold cross-validation using 10,000 randomly sampled data points with a random state(47). We're also using a fraction of the full dataset [0.075, 0.2, 0.5, 0.75, 1.0] with a random state(32) to compare these three models' performance over different data sizes.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Training Set | Test Set | Average K-fold accuracy |
| Logistic Regression | 0.862 | 0.783 | 0.786 |
| Random Forest | 0.730 | 0.719 | 0.721 |
| Decision Tree | 0.752 | 0.598 | 0.603 |

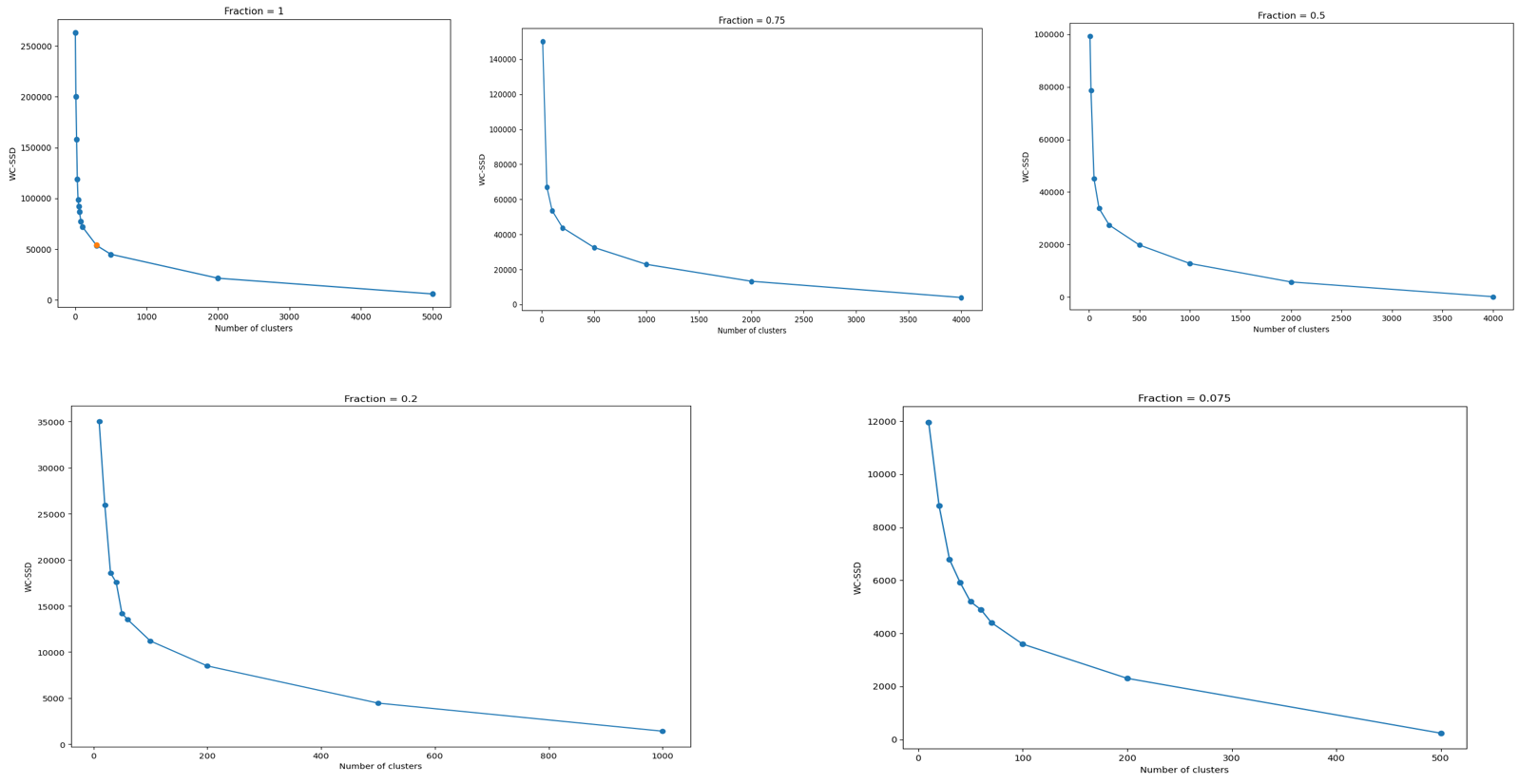
   
**K-means:**



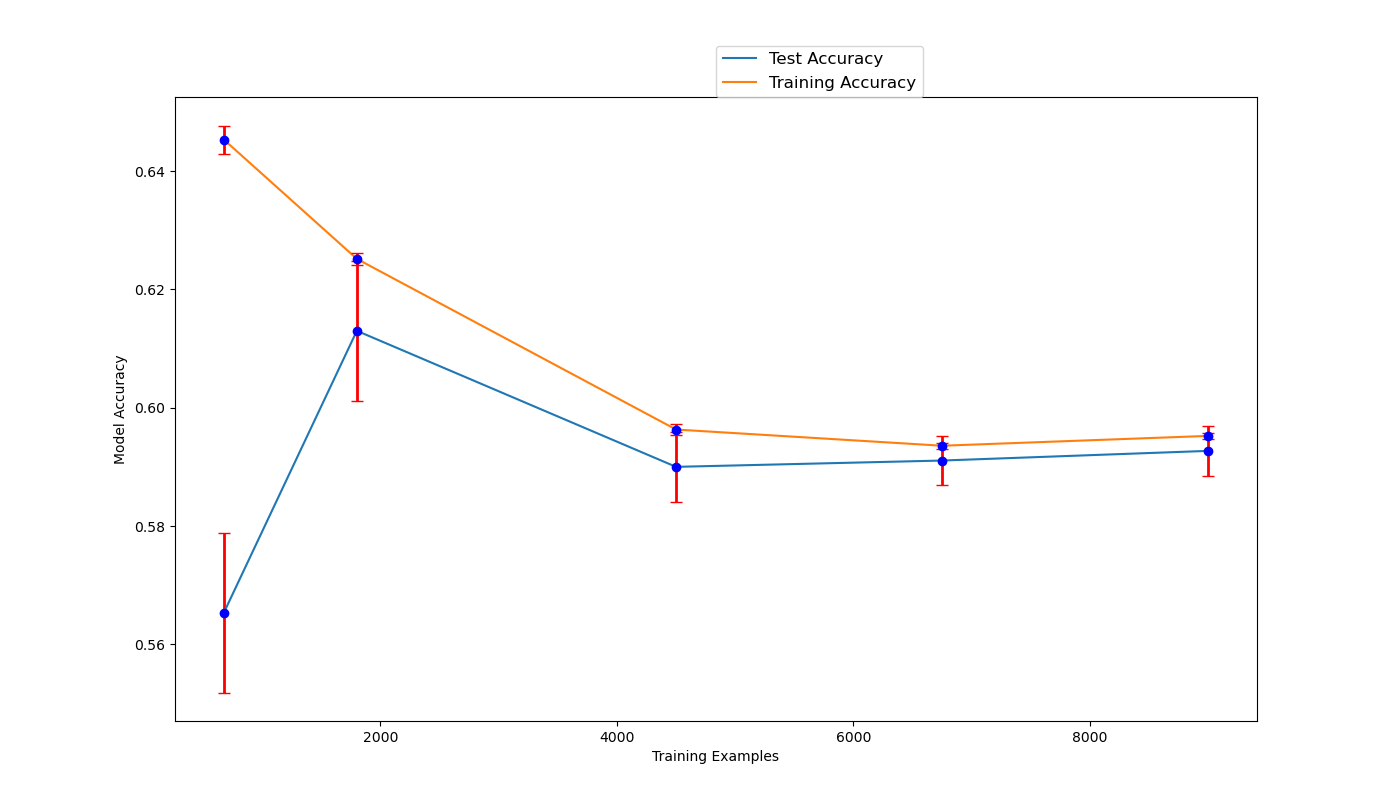
When the sample fraction is 1, 37 POSes in total were found in the training set's essays (less POSes might be found in the test set since it is eight times smaller than our training set). PCA further reduced the dimensionality of our data points from 37 to 28 in the above example. The figure below shows the scatter plot of the first three principal components and their accurate labels (red = female, blue = male).



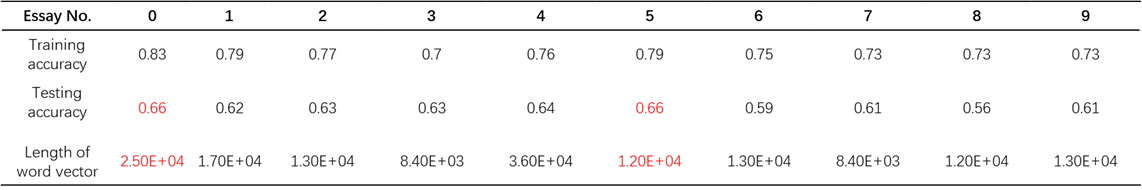
One example of the WC-SSD vs. k plot from each sample fraction is shown in the below figures. We judged that the optimal k for every sample fraction is 50.



The 10-fold cross-validation result is shown in the figure below (the error bar is a standard error). We achieved a test accuracy of 59% when the sample fraction is 1 (9000 data points in the training set and 1000 data points in the test set). This accuracy is lower compared with that of the other models. We assume that this is partly due to our k-means model's unsupervised nature. We were trying to categorize our data points into different classes based on the assumption that a person's use of POS in writing is associated with gender, without directly fitting the features against the gender label. PCA and transformation from more than 65,000 words to 37 POS may have also contributed to the loss of model accuracy by aliasing our original data.



In contrast to the traditional supervised model (Naive Bayes), which is also dealing with the essays, the training accuracy of our k-means model is much lower (more than 0.1 of difference), and the difference between the test accuracies of the two models are not significant (about 0.03 difference). We believe that this is also related to the different data pre-processing or the two models: we used vectorized instead of tokenized essays in our Naive Bayes model, which included more features, and such model can be treated as a special case of a k-means model where k equals the total number of data points (10000 in our data).  
  
**Naive Bayes:** Three are altogether ten essays in the dataset. In the previous report, people tend to train all these data together, which will take a lot of computing power and time. We want to train the model efficiently with the most predicting data. Therefore, we firstly evaluated which essays work best in predicting the gender of the user. We trained our model on these ten essays separately and calculated the training and test errors. It turns out that essays 0 and 5 give the best performance with test error 0.66.  
  
 Interestingly, a larger length of the word vector does not necessarily give a better performance—essay #5 with a better understanding of smaller word vectors while essay #6 the opposite.We also conducted a test on the hyperparameter alpha on essay #5, and it turns out alpha=1 gives the best performance.  
  
 Average accuracy and length of word vector of Multinomial Naive Bayes model on 10 essays

****

Average accuracy of Multinomial Naive Bayes model on different hyperparameter alpha

****

**FC and Bi-LSTM Neural Networks:** We evaluate both models using 10-fold cross validation on 10000 randomly sampled data points (using pandas random\_state = 47). For the FC network, we had 31970 input features which corresponded to the number of unique words in the training corpus. Likewise, we kept our library size to 31970 words for the Bi-LSTM model as well but limited the number of words per input essay to 300. The results are shown below:

Average accuracy for Fully Connected and Bi-LSTM networks

|  |  |
| --- | --- |
| Fully Connected | Bi-LSTM |
| 0.65 | 0.63 |

As one can see, it appears that on average, the Bi-LSTM performs worse than the Fully Connected model. However, we want to know whether this difference is statistically significant. We therefore establish the null hypothesis (no significant difference) and the alternate hypothesis(significant difference). After performing a student’s t-test on the data at an alpha level of 0.05, we find that the resulting p = 0.145 indicating there is insufficient evidence to reject the null hypothesis. Therefore, we say there is no significant difference between the performance of the two models.

We also attempted to tune the network in various ways. One such way was to vary the input length (200 - 500 words) and library size (4000 - 50000 words). However, this only increased model complexity (and model training time) with no statistically significant performance increase. We also added more layers to increase model complexity but found that our current model had already hit a plateau in performance at its current complexity.

**Future Work/Improvements:** One possible future path is the use of n-grams(Kim, 2015). In computational linguistics and probability, an n-gram is a contiguous sequence of n items from a given text or speech sequence. The items can be phonemes, syllables, letters, words, or base pairs according to the application. The n-grams typically are collected from a text or speech corpus. When the items are words, n-grams may also be called shingles. An n-gram of size one is referred to as a "unigram"; size 2 is a "bigram" (or, less commonly, a "digram"); size 3 is a "trigram." Larger sizes are sometimes referred to by the value of n in modern language, e.g., "four-gram," "five-gram," and so on. A simple example could be from our example previously used. For a unigram: "I" "hate" "computers." This would be influenced more by the term: computers, which is generally associated with males. For a bi-gram: "I hate" "hate computers." This would be influenced more by the term: "hate computers" which is generally not associated with males. One possible future path in the deep learning section is using more complex models such as BERT. These complex models were designed for specific uses such as in sentiment analysis and could carry over very well to gender prediction.

**Insights/Recommendations/Conclusion:** With this project, some conclusions could help out OkCupid and its users. We can focus on getting more data, filtering out inconsistent users, and suggestions for users. Getting more data OkCupid does a fantastic job at having many variables so that users can answer some versus very few or even none.   
  
**Suggestion for users:** It is difficult for a user to respond or is unaware that a good full profile will result in more quality matches. A complete profile could mean honesty and transparency for users. This is a lot better than a weak profile, which could lead to a blind date feeling. To help users, OkCupid could use Machine Learning predictions to help users autocomplete their profiles or even encourage users through positive reinforcement. Badges could be earned for complete profiles. This is similar in concept to LinkedIn’s “all-star” profile, where users have a complete profile.

**Filtering out inconsistent users:** Nothing is more frustrating than finding a match online and finding out you have been catfished. “Catfished” is defined as “being deceived by false information online”. This could be tracked by inconsistent responses in a user’s profile. Some ways OkCupid could prevent this is by comparing similar response categories with the average mean. Instead of banning the account, OkCupid could give them fewer matches or matches with other inconsistent users. The reason why OkCupid shouldn’t outright ban inconsistent users is that sometimes, the user is honest. Also, prohibiting users would bring in less revenue overall for OkCupid.

**Final Evaluation:**  
 From our results, we see that the traditional machine learning techniques on the data, not including essays, performed much better on our gender prediction task. On the other hand, our models, which used only essay data, tended to plateau in performance below non-essay data performance. Therefore, we believe it might be better and faster for dating service providers to use non-essay data for gender prediction.

**Contribution** Apart from the below-listed details, all of our team members put equal efforts in documenting the project proposal, Final project report and preparing slides for the Final project presentation.

|  |  |
| --- | --- |
| **Team Member** | **Contribution** |
| Ashish | Random Forest, Decision Tree, Final Project Presentation, Two of the three papers for Literature review, Preliminary analysis and challenges of the dataset. |
| Yiqing | Dataset Analysis and Summary, Data Preprocessing, Logistic Regression model |
| Peter | Essay preprocessing, Deep Learning based models, Final Project Presentation, One literature review paper |
| Xingshuo | Data preprocessing, Naive Bayes model. One paper for Literature review. |
| Haoran | K-means preprocessing, PCA, K-means on essays and Final Project Presentation, Literature review |

References

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Comput. 9*(8), 1735–1780. DOI:<https://doi.org/10.1162/neco.1997.9.8.1735>

# JSE\_OkCupid (2015). https://github.com/rudeboybert/JSE\_OkCupid/blob/master/profiles.csv.zip. Accessed on October 18, 2020.

Kim, A & Escobedo-Land, A. (2015). OkCupid Data for Introductory Statistics and Data Science Courses. *Journal of Statistics Education, 23*(2), DOI: [10.1080/10691898.2015.11889737](https://doi.org/10.1080/10691898.2015.11889737)

Kim, E (2017). *OkCupid Analysis.* https://github.com/kimanalytics/OkCupid-Analysis/blob/master/OkCupid%20Analysis.pdf. Accessed on October 18, 2020.

Newman, M. L., Groom, C. J., Handelman, L. D., & Pennebaker, J. W. (2008). Gender Differences in Language Use: An Analysis of 14,000 Text Samples. *Discourse Processes, 45*(3), 211–236.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J (1985). *Learning internal representations by error propagation*. Tech. rep. ICS 8504. San Diego, California: Institute for Cognitive Science, University of California.

Zhou, C, Sun, C, Liu, Z, and Lau, F. (2015). A c-lstm neural network for text classification. arXiv preprint arXiv:1511.08630.