OkCupid Profile Data Gender Classification

Mark Chang
University of California, San Diego
mac179@ucsd.edu

Minjoo Kim
University of California, San Diego
mek017@ucsd.edu

Eric Kang
University of California, San Diego
ekang@ucsd.edu

ABSTRACT

We will examine the features which make an online dating profile conform to a binary gender. We analyze all aspects of profile data aside from photographs, numerical, categorical, and text data, to train and test a model to predict the gender of the dating profile user.

1 INTRODUCTION

Online dating sites and applications have seen an increase in popularity with the coming of the 'online' era. Where previously this personal affair had been a matter of finding romance in one's social circle, dating websites have allowed users to broaden their scopes and search for their soulmate on the internet. With this, new trends have emerged and the big data created by this transition to the data-driven world has become more easily accessible and analyzed.

OkCupid is a website and app used for online dating where a user can upload pictures and basic information about themselves for other members to view. Data such as age, gender, and sexual orientation are collected for potential matches to filter and view upon. User created descriptions can be added for a more personalized profile.

With this data we will investigate trends among genders, more specifically whether we can predict gender off of text from user created profile descriptions along with a combination of other variables from the profile.

2 DATASET

2.1 Description

The [1] dataset consists of information available on a user's OkCupid profile. There are 59,946 entries in this dataset with 31 different variables/columns available; columns include basic profile data provided by the user such as age, gender, dating preferences, and 10 small "essays" to provide more detail. The sample is sourced from the Northern California Bay Area in 2014 and is not completely representative of online dating as a whole. Of the 31 different variables in this dataset, only 3 are quantitative: age, height, and income.

For the dataset as a whole, the mean age is 32.34 years, with a standard deviation of 9.45, min of 18, median of 30, and max of 110. The mean height is 68.29 inches, with a standard deviation of 3.99 inches, min of 1 inch, median of 68 inches, and 95 inches. The mean income is $20033.22, with a standard deviation of $97,346.12, with a min and median of -1 and max of $1,000,000 - however, 80.8% of these values were imputed as "-1," which states that the individual would rather not respond. After further investigation, there was little to no difference in income based on gender, and this column was dropped. However, as we will investigate trends related specifically to gender, the descriptive statistics of our dataset split via gender is as follows:

Table 1: Descriptive Statistics on the Age of Males vs Females

	Male Age (Years)	Female Age (Years)
Count	35,829	24,117
Mean	32.02	32.82
Standard Deviation	9.03	10.03
Min	18	18
Median	30	30
Max	110	110

Table 2: Descriptive Statistics on the Height of Males vs Females

	Male Height (In.)	Female Height (In.)
Count	35,827	24,116
Mean	70.44	65.10
Standard Deviation	3.08	2.93
Min	1	4
Median	70	65
Max	95	95

Table 3: Count and Percent of Null Values

	Count of Null	Percent
Income	48442	80.8%
Offspring	35561	59.3%
Diet	24395	40.7%
Pets	19921	33.2%

For the other columns, null values were also looked at (the "-1" was interpreted equal to a null value for income). Figure 3 shows the top 5 columns with the most amount of null values; every other column had less than 30% null.

2.2 Exploratory Data Analysis

Our first analysis involved demonstrating differences in the summary statistics of features to be used in the model. We wanted to select features that would best differentiate a profile between male and female to provide data that would best train the classifier. Figures 1 & 2 visualize the difference in the numerical values of age and height.

There seems to be the greatest margin of difference in the features of height, as shown in Figure and Table 1: female profiles generally had lower height values than their male counterparts, revealing a logistic trend between the two data points. The ages between men and women generally follow the same distribution; both are right skewed and peak at the age range 25 - 30. It should be noted

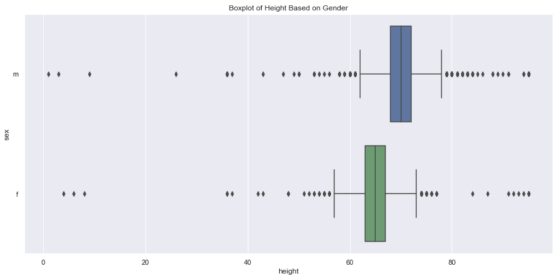

Figure 1: Boxplot of Heights

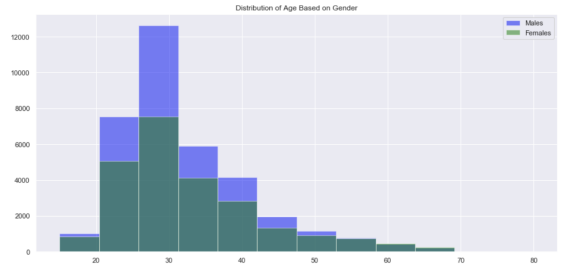

Figure 2: Histogram of Age Distribution

Figure 3: A brief look into the top used words in each sex's profile descriptions, above is for male profiles, below is for female

that although the max of males and females were 110, this was not visualized as we can assume it is not indicative of their true age, and only 1 value for each gender received this age.

We next wanted to examine the difference in word choice between the two sexes in their provided "essays". Each profile is asked to respond to 10 different prompts, with the option skipping one or more. To analyze word usage, a new column was created to combine all the text written by a user regardless if they answered a certain question. From there we calculated the occurrences of each word across both genders and only preserved important words by filtering out those that occurred the most in both genders. We created a word-cloud Figure 3 to demonstrate the most popular words used by each gender. The first word cloud being the top words for males and the second for females.

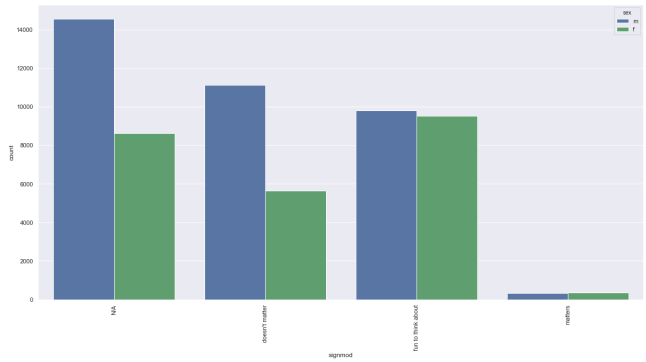

Figure 4: Barchart of Astrology

From this, we found that both genders often state their gender by revealing that they are a "girl" or a "guy" in the profile description. This gives us cause for a decision on whether or not to keep these revealing gender specific words as it may be too obvious and easy for our model to classify upon. Aside from these blatantly obvious words, we also identified that the two genders shared different favorite hobbies, interests, and words to describe oneself. Females frequently mentioned "dancing", "wine", "happy," and "smile", while males frequently mentioned "games", "maybe", "bad", "talk", and "francisco", although this most likely is referencing their location, as the dataset was taken from those in the San Francisco area. This data gives good recourse to fitting a model on word usage as the vectors based on word usage is now shown to be linearly separable.

Other features which showed a correlation between gender included the amount of emphasis placed in Astrology signs. Figure 4 shows that, although the data set was slightly unbalanced with a higher count of males than females, more females stated that astrology signs mattered. Likewise, almost as many females to males stated that signs were "fun to think about", and it was clear that the ratio of males to females who did not care about astrology signs were much higher. Another feature visualized was sexual orientation, as seen in Figure 5. Although approximately 85-86% of men and women are straight, 11.1% of men are gay and only 2.2% of men are bisexual. Women, on the other hand, are gay at a rate of 6.6%, far lower than the rate of gay men, and bisexual at a rate of 8.3%, far higher than the rate of bisexual men.

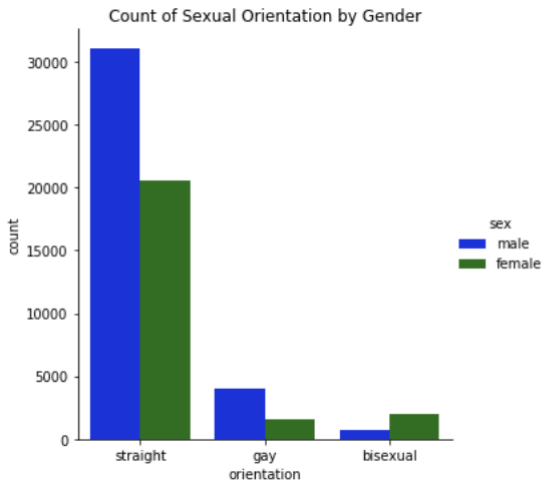

Figure 5: Barchart of Count of Sexual Orientation

3 PREDICTIVE TASK

Our goal is to predict the gender of an OkCupid user based off their profile (non-text and text), using accuracy as our metric, and using naive and simple text features as our baseline. Further specifications regarding metrics will be addressed in Section 4 (Model).

3.1 Feature Engineering

For analysis, the data was separated by sex resulting in two data groups that we could run analysis on to find distinctive characteristics.

Income was not included as they had an overwhelming amount of missingness along with blatantly falsified data from both sexes; the information that was not falsified and more reasonable showed no clear difference between men and women. Location information was also not included as our dataset only collected data from the San Francisco Bay Area, therefore location was only restricted to one or two cities. Although several columns had high null values, the values that were not null proved to show differences between males and females, so were kept. Of the features that we did decide to keep, a few required some data cleaning and are as follows:

For the profile text data (listed as "essay" 0-9), we had to do some data cleaning in order to be used in our model. We first combined all 10 "essay" columns into one combined column that put basically any text that the user wrote into one source. Any null values were imputed with an empty string to avoid concatenation issues. Strings such as "
" and "\n" which indicated line breaks, were removed. We then found the most common 150 words in each sex and identified the words that were commonly used by both genders; any occurrence of these words were removed from the combined text columns alongside any stop words. As we decided to focus solely on text for our baseline model, punctuation and capitalization was also removed; however, this may be worth further looking into. The resulting text data is put into lowercase and made into a feature vector based on the top 5000 words existing in both sex datasets. This feature vector will then be used to train and test the model.

For the profile non-text data (which was essentially every feature other than the "essay" 0-9, age, and income), we encoded the categorical variables into a workable form, primarily through label and one hot encoding. When the feature space was too large for a given feature, we aimed to simplify the data points into a few labels, but otherwise we used a simple label encoding. For example, we identified people's sentiment about zodiac signs into 3 groups: very important, fun, and unimportant from the original sign column, which included their zodiac signs along with their sentiments. These 3 groups were enumerated, label encoded and fed into the model. Similar processes were taken for each categorical feature: the education column, which provided information on where individuals were in their educational path, was narrowed down to whether or not an individual had decided to pursue an advanced degree. Drug and smoking usage were ordinally encoded based on the degree of how often they partook in each activity.

As for data imputation, we generally looked to impute the most frequent data point in the column, since for every categorical variable in the dataset, there was one decisive mode.

3.2 Relevant Baselines

Our primary baseline model we examined was a classification model based on the text, or essay, portion of the dataset. We saw from the EDA portion that there are a few words that are more popular and unique to a single sex. Two words that stood out were "guy" and "girl"; "guy" was very commonly used in male profile essays while "girl" was most often found in female profile essays. We wanted to create a baseline model that counts the number of occurrences of both words in the profile text and predict male if "guy" appeared more often, or female if "girl" appeared more often.

$$f(text) \rightarrow gender$$

$$\text{gender} = \begin{cases} male & count(\text{"guy"} \in text) > count(\text{"girl"} \in text) \\ female & \text{otherwise} \end{cases}$$

This form of baseline will achieve two things. First, the model is able to provide a quantitative standard for future models to build off of as more aspects of the text data is included for evaluation. Second, it answers the worry of gender-revealing words by telling us whether or not the use of gender-revealing words in the profile description made it too easy for a model to predict upon.

The resulting accuracy score of this baseline model was relatively low at around 0.55, proving that this model only predicts slightly better than guessing randomly. We can now move forward with our analysis knowing that gender-revealing words will not completely sway the performance of our model.

4 MODEL

4.1 Model Evaluation

As our task is that of categorical classification, we want to use accuracy to evaluate the performance of our model. Accuracy is useful in finding the performance of a model by calculating the number of accurately predicted values over the total number of

values. The reason we are satisfied with accuracy as compared to alternative measures (recall, F1-score, ppv) is because the dataset was relatively balanced, meaning that a high accuracy cannot be attained by any one model heavily favoring a single class without reason. Therefore, accuracy provides a good insight to how well the model learns the relationship between the features and the target.

4.2 Model Choice

A possible reason for the low performance of the naive baseline model based on most popular word by sex is that although a user may mention their own gender when referencing themselves, they are also very likely to reference the opposite gender when mentioning their preferred significant other. As a result, our model requires more data to make a prediction such as additional text and non-text features shown in the profile, especially as our EDA showed various columns which highlighted the differences between males and females.

Our models included various classifiers (Logistic Regression, Decision Trees, KNeighbors, Gradient Boosting) built and performed on solely the text data, and the non-text data. Based solely on text data, logistic regression lead to an accuracy of 73%; the decision tree classifier and gradient boosting classifier lead to an accuracy of 60% and 68%, respectively, which were better than our simple baseline model but not very accurate. The model for the text data followed a bag-of-words model, which took the most common 5000 words of men and women (stopwords and top 150 common words taken out). A bag-of-words model was chosen because the essay prompts answered specific questions; eg "What are 6 things you could never live without?", "I'm really good at ...", etc. Because the profiles are answered pre-determined questions, we correctly assumed there would be specific activities, attributes, etc that each gender would present.

Utilizing all non-text data, our Decision Tree Classifier, Gradient Boosting Classifier, and KNeighbors classifier came out to 79.7% accuracy, 86.0% accuracy, and 83.2% accuracy, respectively. Already we can see that the non text variables themselves give much higher accuracy.

Our eventual model choice is the combination of several different profile features fitted into one feature vector for use in prediction. A combination of both text and non text features fed into an ensemble learning model proved to be the best performing. Text data is processed into a bag-of-words approach and the resulting vector is combined with the non-text data processed as mentioned in the feature engineering section. Our justification for using this model is clear because in our EDA we found clear relationships between key non-text features and gender, as well as being able to qualitatively see a relationship between gender and text features. Since the variance contained in both of these feature subspaces don't overlap, it is assuredly worthwhile to include both of them in the feature space of our final model. The results for our final model can be seen in Table 5.

Table 4: Baseline Model Performance

Model	Accuracy
LogisticRegression	0.5578
DecisionTreeClassifier	0.5412
GradientBoostingClassifier	0.5561
KNeighborsClassifier	0.5343

Modern data science competitions and research has shown ensemble learning to be one of the best, if not the best performing algorithm classes. Neural networks and other cutting-edge deep learning solutions do not provide a benefit appropriately proportional with the compute power/complexity of the system. We selected each model to work up the expected performance ladder from logistic regression to ensemble learning. We hoped to see a visible climb in performance for our dataset.

$$f(text, ProfileData) \rightarrow gender$$

$$\text{gender} \simeq \alpha + \sum_{w \in text} count(w) * \theta_w + ProfileData_i * \theta_i$$

Table 5: All Features Model Performance

Model	Accuracy
LogisticRegression	0.69420
DecisionTreeClassifier	0.8213
GradientBoostingClassifier	0.8792
KNeighborsClassifier	0.80950

4.3 Model Optimization

We optimized our each of our models with an exhaustive GridSearch on the hyper parameters. For Logistic Regression, we adjusted the inverse of regularization strength (C), the tolerance for stopping, and looped through each solver. For the Decision Tree, we looked at both gini and entropy, and had max depth, min samples split, min samples leaf, and min weight fraction leaf gridsearched to find the optimal set of hyperparameters. For Gradient Boosting we adjusted the learning rate, num estimators, and the subsample. For K nearest neighbors, the hyperparameters we searched through were n_neighbors, p (power parameter for the Minkowski metric) and the leaf size.

We saw no issues with scalability, but certainly do forecast that in large datasets, our approach would have to be altered or exclusively use algorithms that are parallelizable. Our feature space is too large to support tens of millions to hundreds of millions of rows, so caution is advised there. In order to combat any potential overfitting, we employed K-folds cross validation with K=8. Over multiple iterations, we found no evidence showing that overfitting was a large issue in our models. However, there was a noticeable (but not dramatic) amount of overfitting in our Decision Tree and KNN classifiers.

The strengths of our highest performing classifier (GBC) are evident, as it is the only ensemble learning classifier that we used. Ensemble learning has clear researched benefits on most, if not all relational data. The primary disadvantage in Gradient Boosting is that it is more sensitive to outliers than other models. However, since rhe domain of each feature is relatively tight, this did not impact the model performance a significant or noticeable amount. Gradient Boosting is able to avoid overfitting while still learning

a large feature space relatively well. Logistic regression has some issues when we consider large feature spaces, as complex relationships a,e dififcult for logistic regression to learn. It also constructs linear boundaries, which sometimes are not entirely appropriate. Logistic regression does have the benefit of measuring feature importance and a directionality to that vector. Both Decision Trees and KNN suffer from a big issue in that they overfit the data easily, but fortunately we were able to combat that with our k-folds cross validation. In addition, both of these do not have the advantage of mixing multiple learners.

5 RELATED LITERATURE

The OkCupid Dataset was provided in the Journal of Statistics Education July 2015, and it can be assumed there has been various works done with this specific dataset. However, as we are looking into text processing (along with other encoded features) to predict gender, related works in which text was used to predict gender were looked into. There have been several related works, specifically, works that look into gender classification based on text, a major component of our classification model. In [2] Chat Mining for Gender Prediction, led by Can, Kucukyilmax, Cambazoglu, and Aykanat, text was scraped from a chat server, and consisted of a vocabulary of 50,000 distinct words over 250,000 chat messages. Researchers look at features such as words, message length, stopwords usage, smiley usage, character usage, and punctuation. Their classifiers included a K-Nearest Neighbors (with k = 10) algorithm that utilized cosine similarity as a distance metric, naive Bayesian, and back propagation. Ultimately, the naive bayesian classifier received the best accuracy of 84.2% after removing stop words and vocabulary richness measures. Moreover, the study revealed that males tend to use more slang, have longer chat messages, but shorter words. Females, on the other hand, use more possessive and content dependent words, longer words, often omit stop words and punctuation, but will use more smiley emoticons (a list of 79 smiley emoticons were used to determine if the text contained a “smiley”).

Another work, [3] “Text-Based Age and Gender Prediction for Online Safety Monitoring”, from van de Loo, De Pauw, and Daelemans, utilized 379,769 public chat posts from a Belgian social networking website, NetLog. Although text preprocessing was utilized, such as changing all words to lowercase and cutting extra letters to a maximum of three (e.g. “Hiiiii” was changed to “Hiii” for generalization), character n-grams were collected from the raw text to capture stylistic characteristics. Features included the 2,500 most frequent unigrams and bigrams and 5,000 most frequent character trigrams and tetragrams for their model, which was a Support Vector Machine algorithm utilized 3-fold cross validation on the training set. Ultimately, the recall scores for gender prediction specifically were 79.7% and 53% for females and males, respectively; however, as this dataset was imbalanced, with a balanced subset the recall score was then 65.8% and 72.7%. Finally, in this realm, the [4] state of the art models are from Mukherjee and Liu in their work “Improving Gender Classification of Blog Authors.” Employing web blogs as a dataset, Mukherjee and Liu utilized an ensemble of feature selection criteria and methods to capture the most useful features, as well as finding part of speech (POS) sequence patterns rather than POS n-grams. There was a focus on F-measure, which explores a notion of implicitness of text (rather than explicitness), stylistic features (words appearing with high frequency in the context of a blog), gender preferential features (e.g. women use more emotionally intensive adverbs whereas men express independence), and word classes (grouping words into positive, negative, or neutral words) which were then modeled by SVM classification, SVM regression, and naive bayes. Accuracy was used as the dataset was balanced, and SVM regression resulted in an 88.56% accuracy, which was far better than current state of the art methods.

We were able to get an accuracy comparable to the current state of the art methods; however, we were only able to do so with outside non-text features (the related literature above used solely text). As our focus was using the entire dating profile to predict gender, our text preprocessing was not as extensive as related works; however, further research into looking at punctuation, variances of words on purposes, stop word usage, and more could prove helpful to our model.

6 RESULTS

We concluded that predicting the sex of a profile based off of data displayed by a user is entirely probable. We saw the best results when factoring in all aspects of the profile data; numerical, categorical, and text data were all used to train the model and each on their own did not perform as well as when combined. Our results show that the GradientBoostingClassifier performed the best on the combined data with an accuracy score of around 0.88; this value is significantly better than the alternatives of our naive baseline model and our text-only model, and only slightly better than the model with only non-text features. We can see that, although our accuracy was quite high, our text analysis, preprocessing, and model was the weaker component when predicting gender, as it was a simple bag of words model. In terms of feature representations, it is clear that our non-text features were represented fairly well, as including our text features increased our accuracy by around 2%. Feature representations of text could have been extended further beyond bag of words to include parts of speech usage, use of punctuation, use of capital words, sentiment analysis, and slang, as our literature review showed that males and females tend to use different word lengths, punctuation as emoticons, and varying amounts of parts of speech. Moreover, our bag of words model included utilizing the top 5000 words used by both genders, as this provided the highest accuracy compared to top 1000 words, 2000 words, and 10000 words. However, there may have been a more optimal word count utilized for our bag of words model. In terms of hyper-parameters, our gradient boosting algorithm performed the best, with an increased learning rate to 0.15 from the default of 0.1, a number of estimators of 200, and a sub-sample change to 0.9. As we performed a grid search to get the optimal hyper-parameters and model on our data, we were successful in achieving a model with an accuracy of 88%. However, it should be noted that different feature representations could work better or worse for some models - past research into similar predictions showed that different feature representation combinations, especially for text, can lead to improved results and ideally, a grid search with various feature representations in various models could be performed; in this research, the same feature

representations were utilized for all models. Nonetheless, the model far passed our baseline and provides room for further research.

7 BIBLIOGRAPHY

[1] Kim, Albert Escobedo-Land, Adriana, OkCupid Profile Data for Intro Stats and Data Science Courses, (2015), JSE_OkCupid, https://github.com/rudeboybert/JSE_OkCupid

[2] Kucukyilmaz, Tayfun Cambazoglu, Berkant Aykanat, Cevdet & Can, Fazli. (2006). Chat Mining for Gender Prediction. 274-283. 10.1007/11890393

[3] Loo, Janneke. (2016). Text-Based Age and Gender Prediction for Online Safety Monitoring. International Journal of Cyber-Security and Digital Forensics. 5. 46-60. 10.17781/P002012

[4] Mukherjee, Arjun & Liu, Bing. (2010). Improving Gender Classification of Blog Authors.. EMNLP 2010 - Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference. 207-217