Humanities Project: Sexism in Movies

Final Report (11th May 2021)

Harshita Sharma: 20171099, Ayushi Kumari Agarwal:

1 About the Project

This project aims to study gender bias in movies and understand how widespread it is over time, in different genres etc. through the means of a variety of tasks using Machine Learning, NLP and sentiment analysis.

The motivation behind some of the tasks comes from a similar study performed on 2000 films compiling the number of words spoken by male/female characters in movie scripts. This study is done by The Pudding (a digital publication that explains ideas debated in culture with visual essays) to obtain a much more objective view of gender in films.

The focus of this project was on creating a model to find the gender bias in movies from their movie scripts because it is very difficult to obtain the true gender of characters from the web unless the task is done manually. We have also faced a lot of issues while extracting this gender data. Hence, given a script, our model will present the distribution of dialogues across male and female characters. Not only this, but we have also extended our project to find the distribution of characters, speaking roles etc in a movie.

In all milestones covered as part of this project, the focus is on automation and achieving accuracy as high as possible because most of the research that exists today that deals with movie scripts is done manually.

2 Project Milestones

	Milestones
1	Reading research papers related to the analysis of sexist texts, analysis of movie scripts etc.
2	Data Collection
3	Male/Female distribution of dialogues in a movie
4	Male/Female distribution of characters in a movie
5	Male/Female distribution of named characters in a movie
6	Male/Female distribution of character with most dialogues in movies
7	Male/Female distribution of Speaking roles in a movie

3 Data Collection

The data required for the project is scripts of English movies, characters of English Movies, year of the movie release, gender of each character.

This milestone can be divided into the following tasks:

	Tasks
1	Web Scraping of Movie Scripts and Year of Release of Movies
2	Characters in a movie
3	Gender of each character

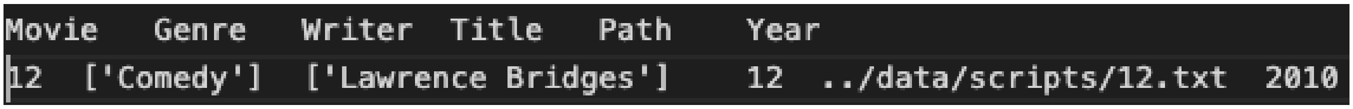
3.1 Web Scraping of Movie Scripts and Year of Release

Movie Scripts were scraped from the IMSDb website using python.

1. Collected 1210 scripts out of which ~1145 had a similar format and were available in HTML. On further examination of the scripts, ~8 were empty. Finally, 1137 scripts have been finalised(by the first intermediate meeting).
2. Meta-information regarding the scripts like Movie Title, Genre, Writers was collected from the same website.
3. The year of release was collected separately using the IMDb library.
4. Missing or non-uniform information was checked and corrected manually.
5. The dataset ranges from the years 1915-2022. These years mark the year of release of the movie.
6. The dataset covers movies belonging to a total of 23 genres listed below:

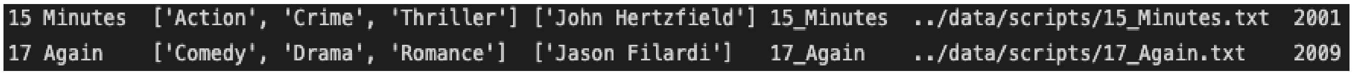
   Comedy, Romance, Drama, Sci-Fi, Thriller, Adventure, Action, Crime, Horror, Mystery, Animation, Fantasy, Family, Musical, Western, War, Biography, Music, Film-Noir, History, Short, Sport, Action.Thriller, Horror.Mystery.

7. The meta-information of the movies in the dataset is organised in a CSV file like this:



```
Movie   Genre   Writer  Title   Path    Year
12  ['Comedy']  ['Lawrence Bridges']    12  ../data/scripts/12.txt  2010
```

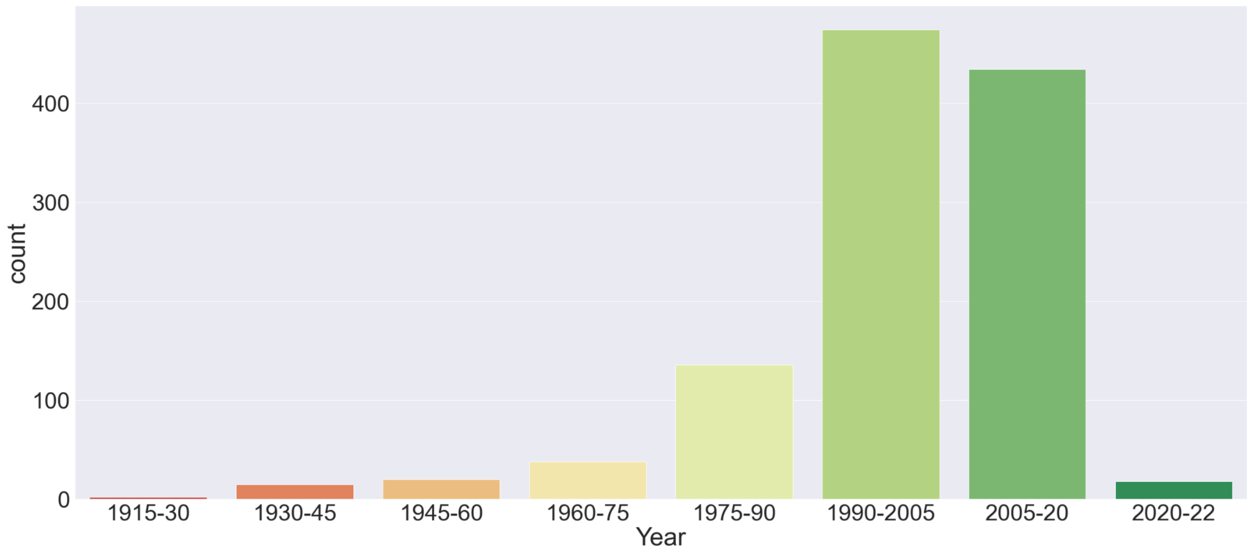
   More examples show that a movie can be categorised in several genres:



```
15 Minutes   ['Action', 'Crime', 'Thriller'] ['John Hertzfield'] 15_Minutes   ../data/scripts/15_Minutes.txt  2001
17 Again     ['Comedy', 'Drama', 'Romance']  ['Jason Filardi']   17_Again     ../data/scripts/17_Again.txt    2009
```

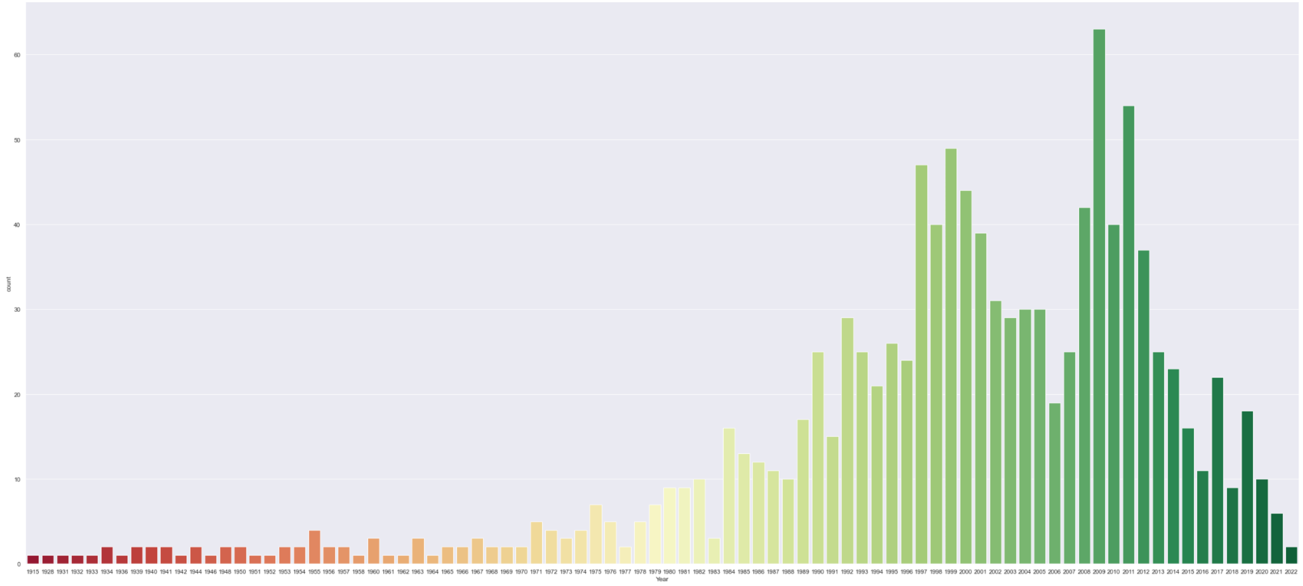
3.1.1 Data Visualisation [Movie Scripts]

1. Visualising the collected data by year (groups of15 years): The dataset was divided into groups, grouped by year e.g. 1915-30, 1930-45…..2005-20, 2020-22.

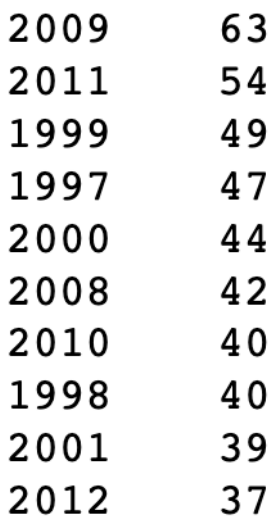


Most of the movies in the dataset are from 1990-2020 counting up to 907 movies out of 1137 whereas 1915-30 has movies as little as 2. For more clarity on the distribution of the dataset, we look at the next visualization.

2. Visualising the collected data by exact year of release: The following figure shows the non-uniformity in the dataset if we look at the year of release of movies.

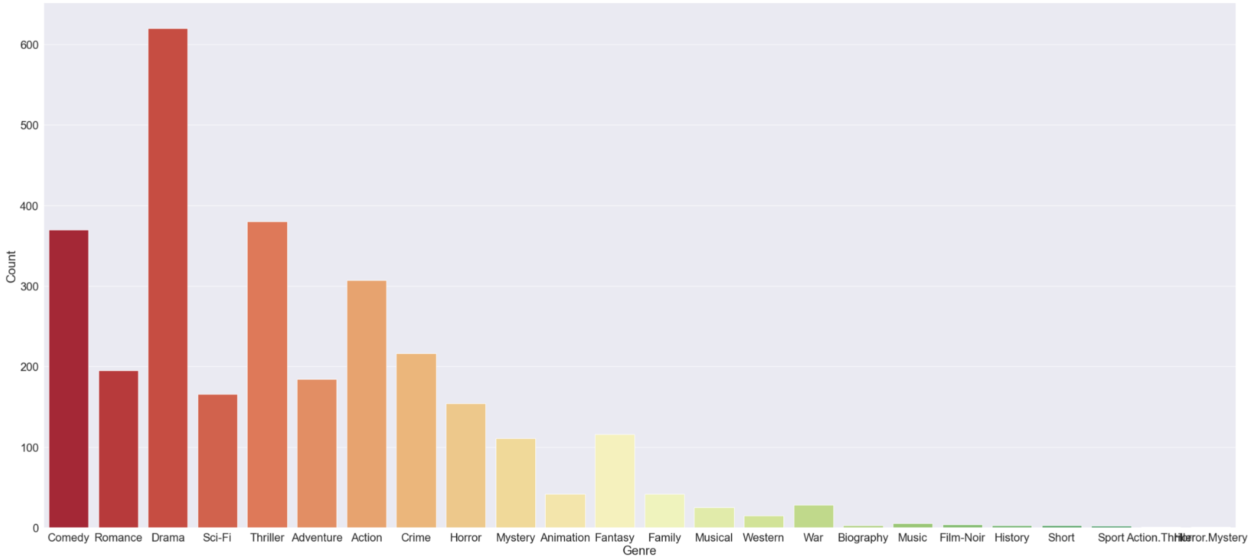


Some years in the dataset with the highest number of films in the dataset:



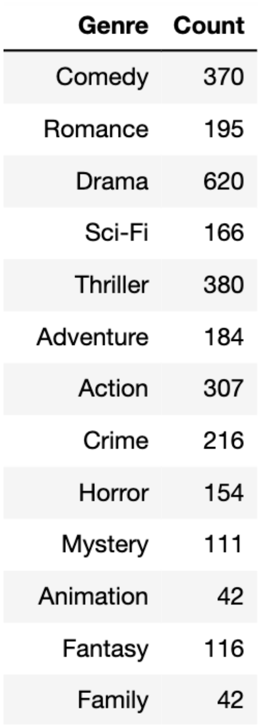
```
2009      63
2011      54
1999      49
1997      47
2000      44
2008      42
2010      40
1998      40
2001      39
2012      37
```

3. Visualising the collected data by genre: Again, a non-uniformity can be seen in the distribution of movies over the different genres.



Genres and the number of movies in each genre:[1]

[1] Each movie can have more than one genre so the sum of the counts will be greater than the number of movies

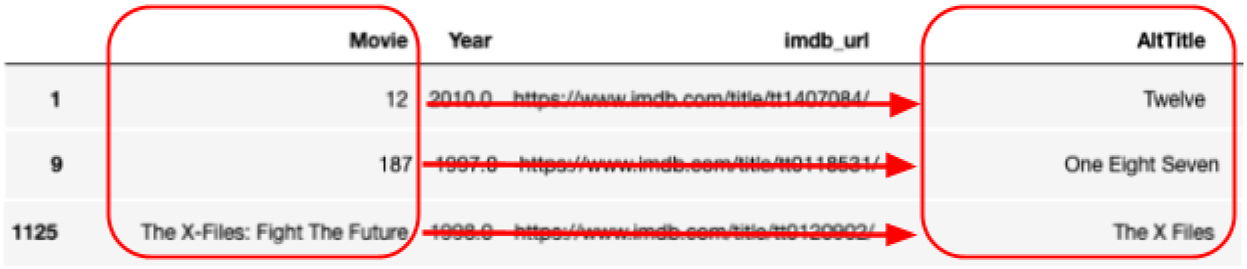


Genre	Count
Comedy	370
Romance	195
Drama	620
Sci-Fi	166
Thriller	380
Adventure	184
Action	307
Crime	216
Horror	154
Mystery	111
Animation	42
Fantasy	116
Family	42

3.2 Characters in a movie

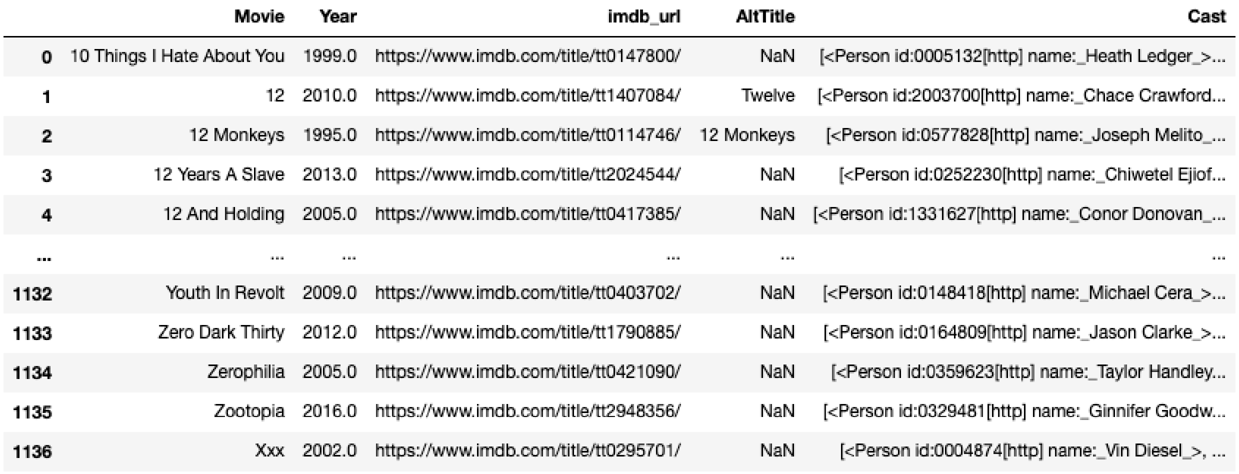
For Milestone 5 and 6, which deals with the characters and named characters of any movie, we collect three kinds of information:

1. each character from the movies in the scripts dataset and compile another dataset that contains the character name and actor who plays that character in a given movie.
2. To ease the looking up of movies in the IMDb dataset/website, we have also collected the IMDb ID of each movie in our dataset, using which one can easily access the movie details from hundreds of other IMDb datasets available, IMDb libraries and the IMDb website itself.
3. The Alt-Title aka alternate title of each movie was collected because some Movie Titles were written differently in our original dataset while they were different on IMDb. For example:



	Movie	Year	imdb_url	AltTitle
1	12	2010.0	https://www.imdb.com/title/tt1407084/	Twelve
9	187	1997.0	https://www.imdb.com/title/tt0118531/	One Eight Seven
1125	The X-Files: Fight The Future	1998.0	https://www.imdb.com/title/tt0120902/	The X Files

This is how our final characters dataset looks like:



	Movie	Year	imdb_url	AltTitle	Cast
0	10 Things I Hate About You	1999.0	https://www.imdb.com/title/tt0147800/	NaN	[<Person id:0005132[http] name:_Heath Ledger_>...
1	12	2010.0	https://www.imdb.com/title/tt1407084/	Twelve	[<Person id:2003700[http] name:_Chace Crawford...
2	12 Monkeys	1995.0	https://www.imdb.com/title/tt0114746/	12 Monkeys	[<Person id:0577828[http] name:_Joseph Melito_...
3	12 Years A Slave	2013.0	https://www.imdb.com/title/tt2024544/	NaN	[<Person id:0252230[http] name:_Chiwetel Ejiof...
4	12 And Holding	2005.0	https://www.imdb.com/title/tt0417385/	NaN	[<Person id:1331627[http] name:_Conor Donovan_...
...	...	...	...	...	...
1132	Youth In Revolt	2009.0	https://www.imdb.com/title/tt0403702/	NaN	[<Person id:0148418[http] name:_Michael Cera_>...
1133	Zero Dark Thirty	2012.0	https://www.imdb.com/title/tt1790885/	NaN	[<Person id:0164809[http] name:_Jason Clarke_>...
1134	Zerophilia	2005.0	https://www.imdb.com/title/tt0421090/	NaN	[<Person id:0359623[http] name:_Taylor Handley...
1135	Zootopia	2016.0	https://www.imdb.com/title/tt2948356/	NaN	[<Person id:0329481[http] name:_Ginnifer Goodw...
1136	Xxx	2002.0	https://www.imdb.com/title/tt0295701/	NaN	[<Person id:0004874[http] name:_Vin Diesel_>, ...

3.3 Gender data

<Ayushi>

4 Male/Female distribution of dialogues in a movie

During the collection of gender data (not mentioned in this report), we faced a lot of problems while finding the accurate gender of each character. We could find only ~7000 characters from specific movies and their genders explicitly out of the ~48,000 speakers in our dataset. The rest of the characters were taken care of. However, this is a clear example to show how difficult it is to get this information if one is dealing with a lot of movies, wherein the collection of gender data manually becomes tiresome. Hence, in this task, to ease this problem we create a model to generate the number of dialogues spoken by male/female characters in a movie given its script.

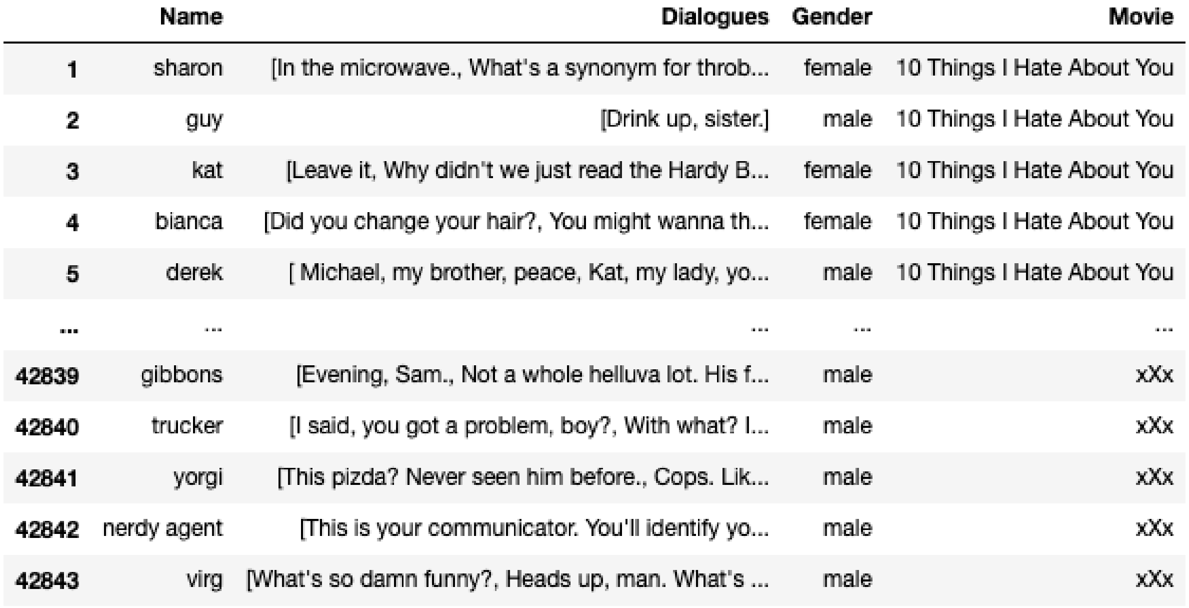
To do this, a number of tasks follow:

	Tasks
1	Data Preprocessing
2	Feature extraction
3	Training and Testing Models using Gender data
4	Prediction of gender using the model and gender distribution across dialogues

4.1 Data Preprocessing

<Ayushi>

Once, the speakers and dialogues were extracted, they were processed again to compile them in a particular format. While doing this the true labels extracted in the Data Collection module were also used to assign gender to each character.



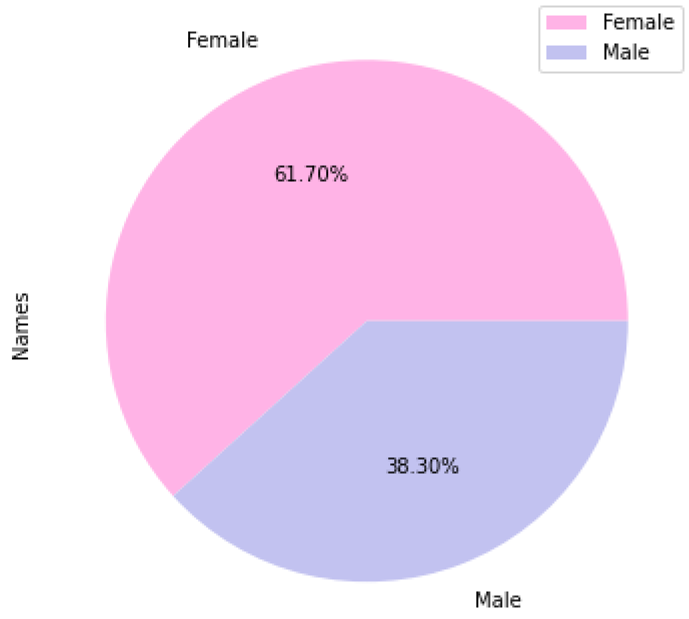
	Name	Dialogues	Gender	Movie
1	sharon	[In the microwave., What's a synonym for throb...	female	10 Things I Hate About You
2	guy	[Drink up, sister.]	male	10 Things I Hate About You
3	kat	[Leave it, Why didn't we just read the Hardy B...	female	10 Things I Hate About You
4	bianca	[Did you change your hair?, You might wanna th...	female	10 Things I Hate About You
5	derek	[Michael, my brother, peace, Kat, my lady, yo...	male	10 Things I Hate About You
...	...	...	...	...
42839	gibbons	[Evening, Sam., Not a whole helluva lot. His f...	male	xXx
42840	trucker	[I said, you got a problem, boy?, With what? I...	male	xXx
42841	yorgi	[This pizda? Never seen him before., Cops. Lik...	male	xXx
42842	nerdy agent	[This is your communicator. You'll identify yo...	male	xXx
42843	virg	[What's so damn funny?, Heads up, man. What's ...	male	xXx

4.2 Feature Extraction

4.2.1 Feature Extraction by First Names

The aim here is to find features that can be used to distinguish between male and female names. To decide whether the extracted features are useful or not we tested these features using 12k distinct common names collected as part of Milestone 4, Gender Distribution of Characters in a movie. We used these 12,000 common names to decide whether a given feature is helpful in predicting gender or not.

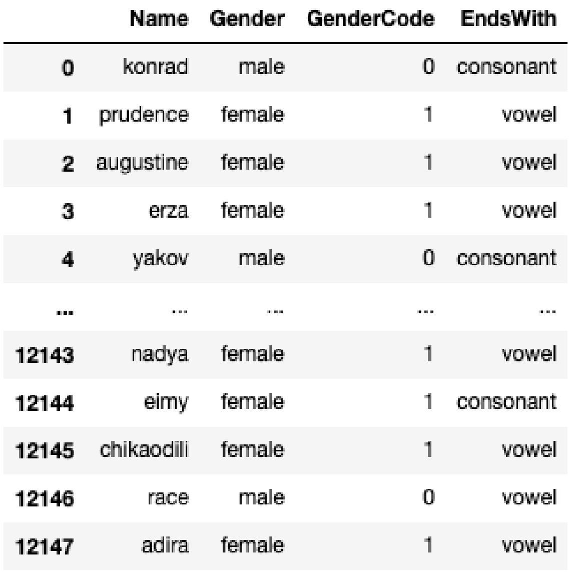
We start by visualising the gender distribution of these 12k names, to find that our database has ~7,414 female names and ~4,603 male names:



The uneven distribution of names will not affect our feature selection criteria since the features will be observed for names of each gender separately.

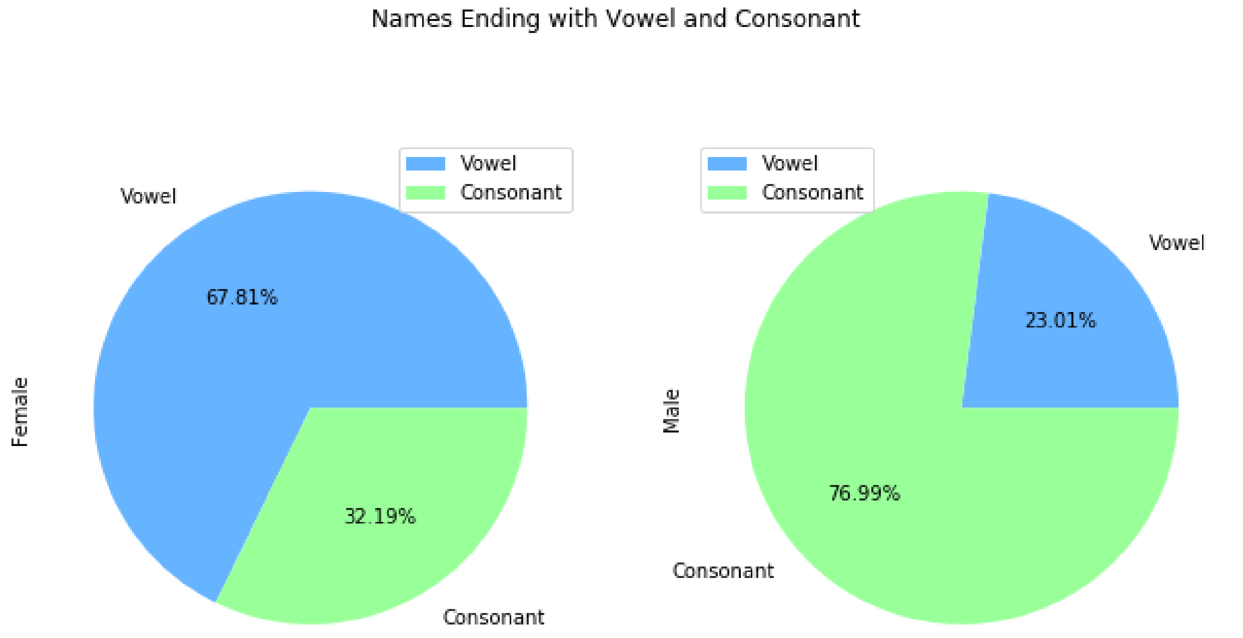
Feature 1: Name ends with Vowel

We find whether each name ends with a vowel or not and use it to check if there is a difference in how this feature behaves for male and female names.



	Name	Gender	GenderCode	EndsWith
0	konrad	male	0	consonant
1	prudence	female	1	vowel
2	augustine	female	1	vowel
3	erza	female	1	vowel
4	yakov	male	0	consonant
...	...	...	...	...
12143	nadya	female	1	vowel
12144	eimy	female	1	consonant
12145	chikaodili	female	1	vowel
12146	race	male	0	vowel
12147	adira	female	1	vowel

Turns out, there is one. Most of the female names end in vowels whereas it is different for male names. This can be seen in the pie plot given below:



Hence, we keep this feature to train our model.

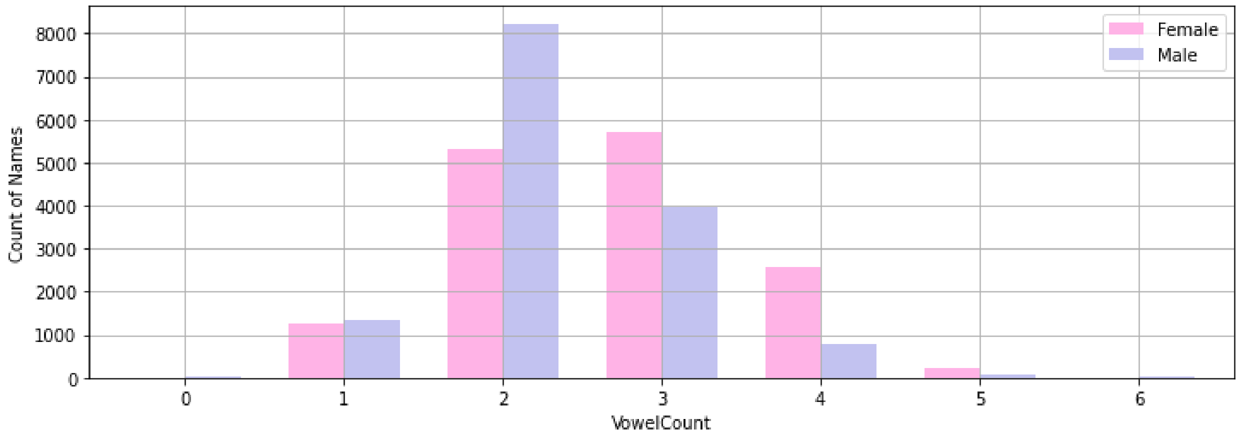
Feature 2 and 3: Count Vowels and Consonants in a name

We find the count of vowels and consonants in each name and use that data as two features.

Feature 4: Classification of the count of consonants and vowels

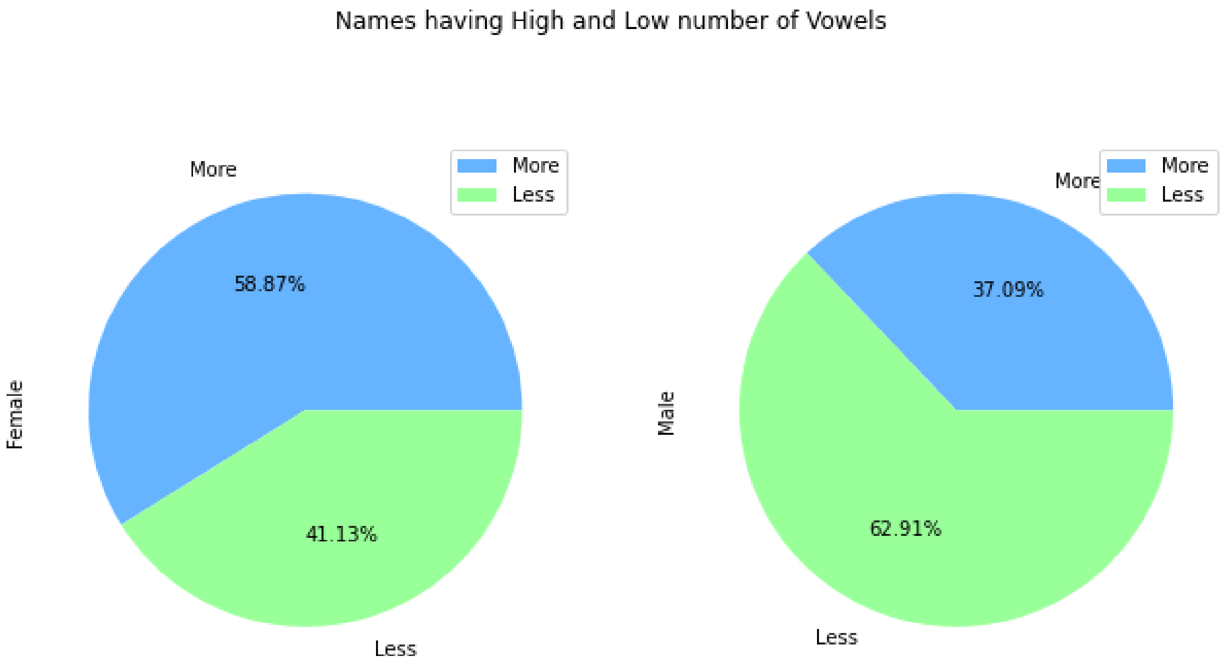
Using the previous two features 2 and 3 we computed the next features which can be called similar to 2 and 3 since they are derived from them. These features are just another check to see if the female names and male names differ.

To find a threshold that can be used to classify the count of vowels as high or more, we used a dataset of Popular baby Names by the state of Austin that has the count of names as well.



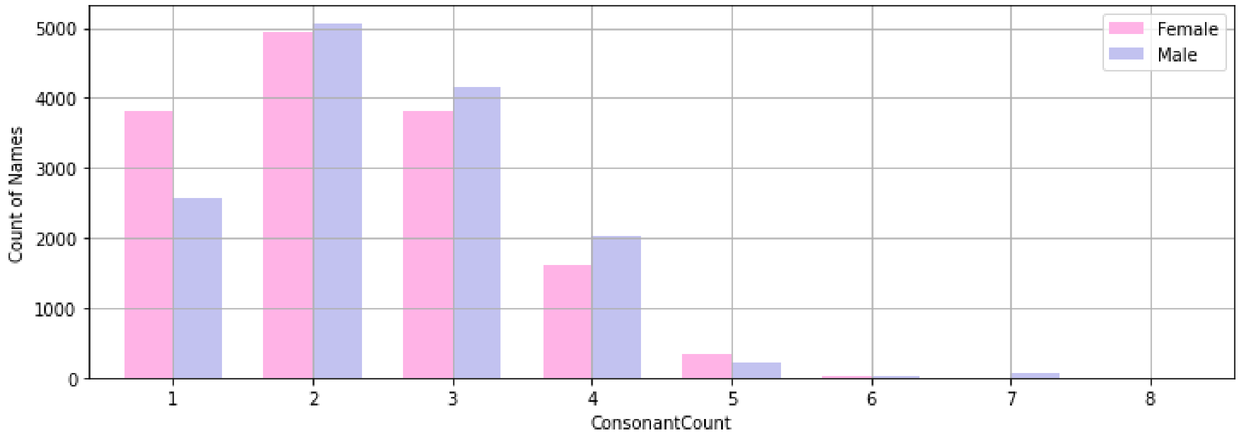
Through this plot, we can clearly see that the vowel count shifts sides from male to female once it crosses 2. So, we kept 2 as the threshold i.e. if VowelCount is greater than 2 it is a high count else, low.

Using this we classified names having HighVowelCount or LowVowelCount and observed the statistics and how it differs in male and female names using the pie chart given below:

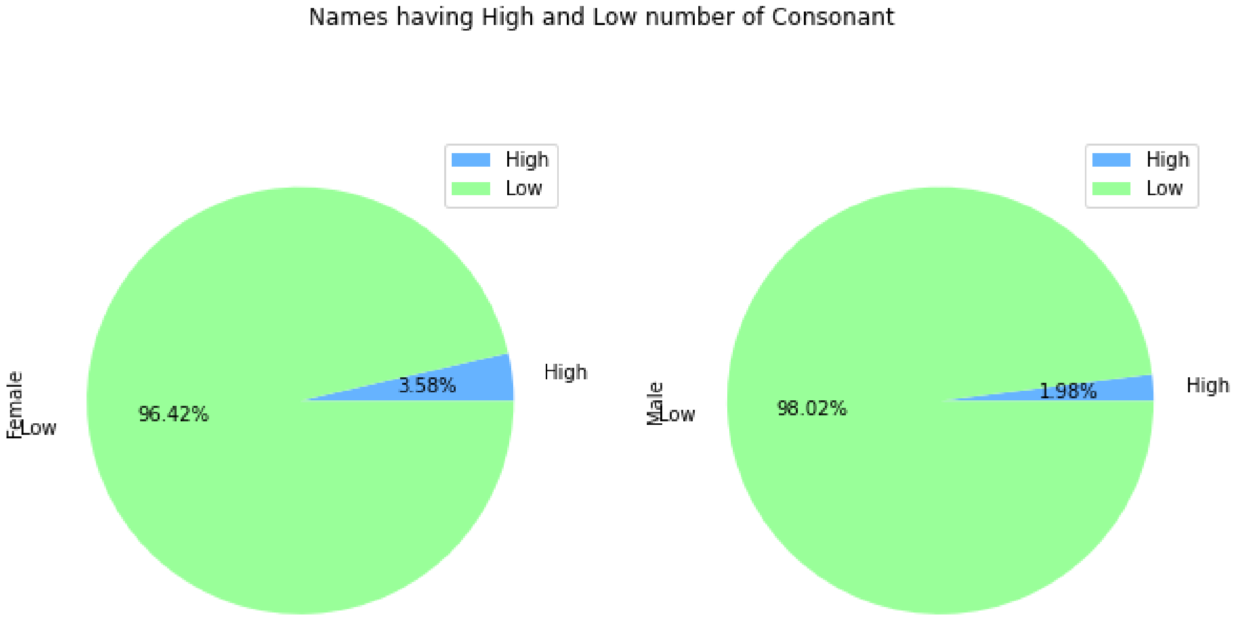


Turns out, VowelCount can be used to distinguish between male and female names, so this feature was also accepted.

Next, we studied the same trend for Consonants:



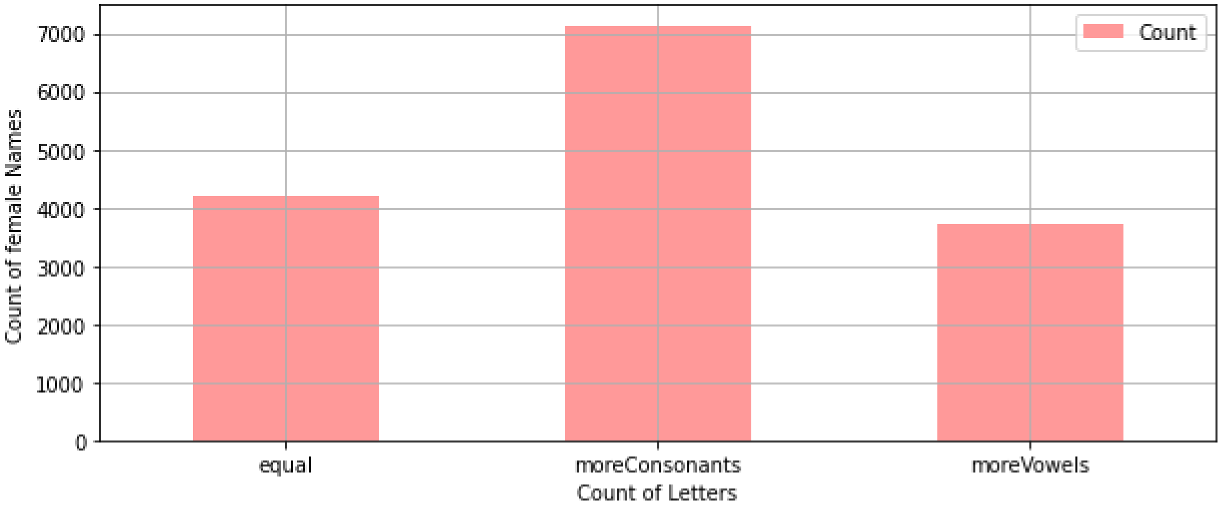
And we saw the sides switch when the consonant count was 1. So it was kept as a threshold. Talking without statistics, we weren't confident about this feature as it doesn’t seem to be a very sensible one. Nevertheless, we classified names having HighConsonantCount or LowConsonantCount and observed the statistics and how it differs in male and female names using a pie chart and our suspicion was found to be correct.

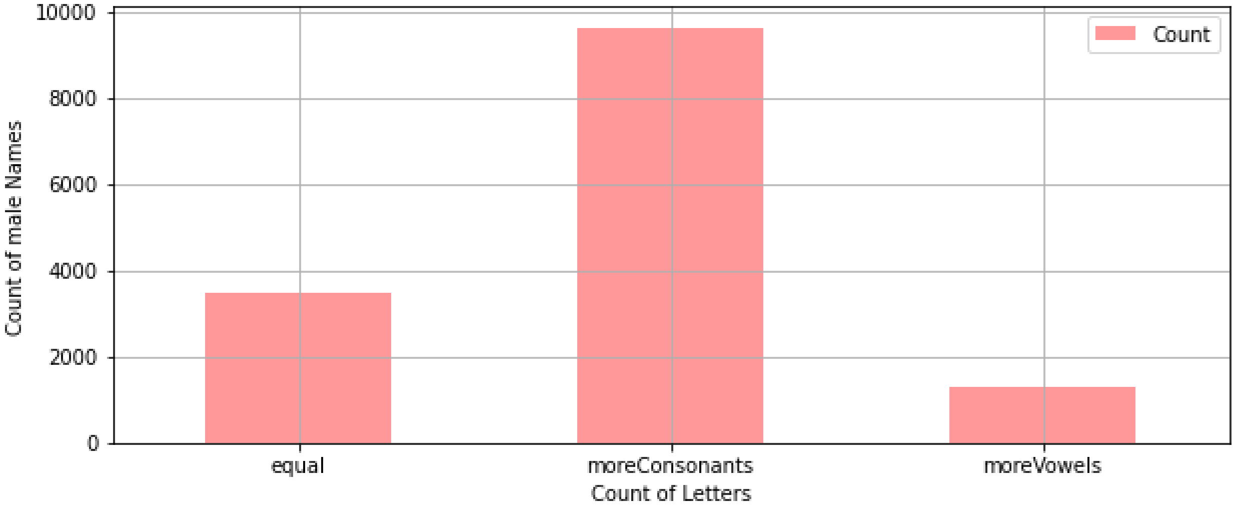


Since only the vowel count shows a difference we used only that as a distinguishing feature.

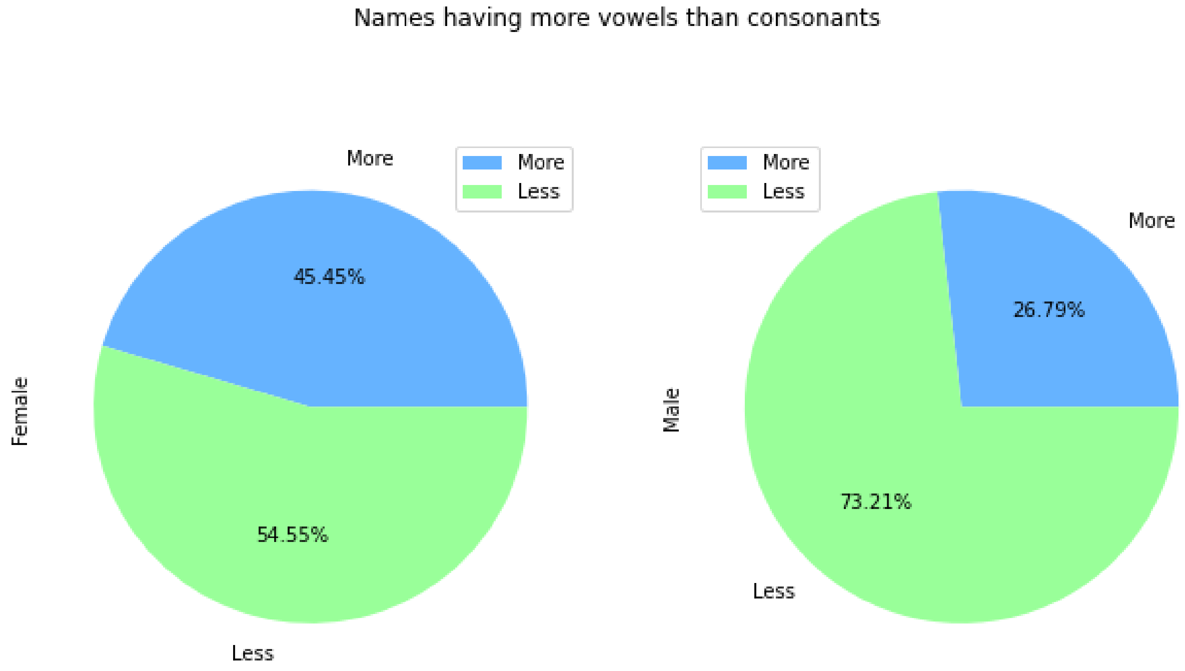
Feature 5: Difference in Vowels and Consonants

For the next feature, we start by comparing the number of vowels and consonants in each name. If the number of vowels is greater than consonants, a column 'vcCompare' is assigned the value 'more', similarity values 'equal' and 'less' are assigned. The statistics of the 'vcCompare' column on the Autin Baby Names Dataset is given below:





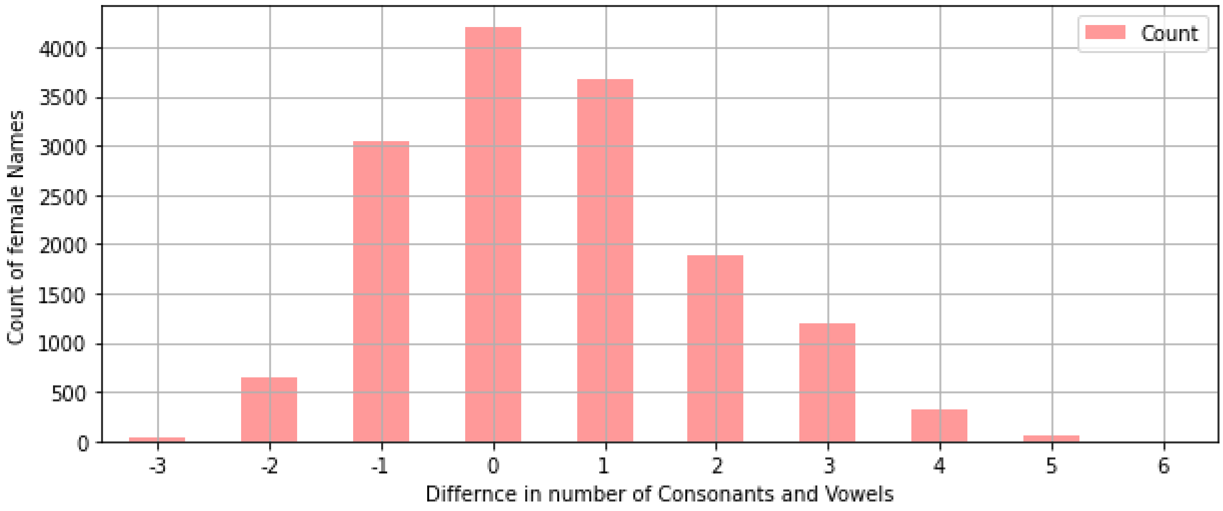
Since it was very obvious that the names having 'moreConstants' will be more, it wasn't the point of focus, instead, these statistics will help us decide if the 'moreVowels' feature can be used or not. Hence, we grouped 'moreVowels' and equal together to get a class of MoreVowels i.e. if the ‘vcCompare’ had a value ‘moreConsonants’, then ‘moreVowels’ feature will hold the value ‘less’ else ‘more’. The plot below shows how female and male names behave when it comes to having more vowels than consonants in a name. This plot confirms that these can be used as a feature:

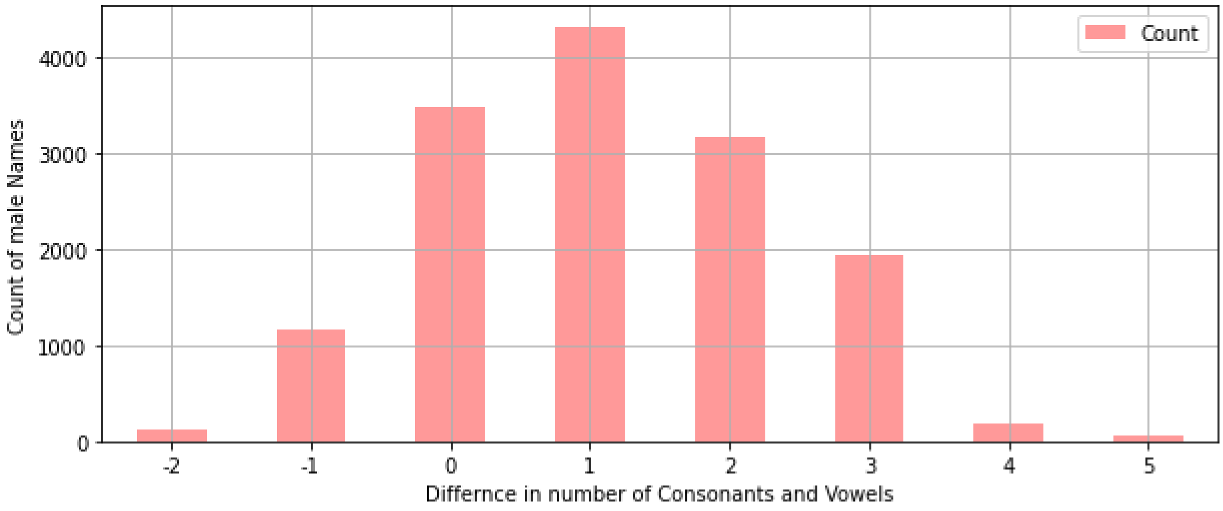


Feature 7 and 8: Difference in Vowels and Consonants

We calculated the actual difference between the number of consonants and the number of vowels and named it ‘vcDifference’.

We used ‘vcDifference’ to classify names into classes of more difference and less difference. To do this we needed a threshold, which again was found using the Austin Popular names dataset. The same plot is given below:

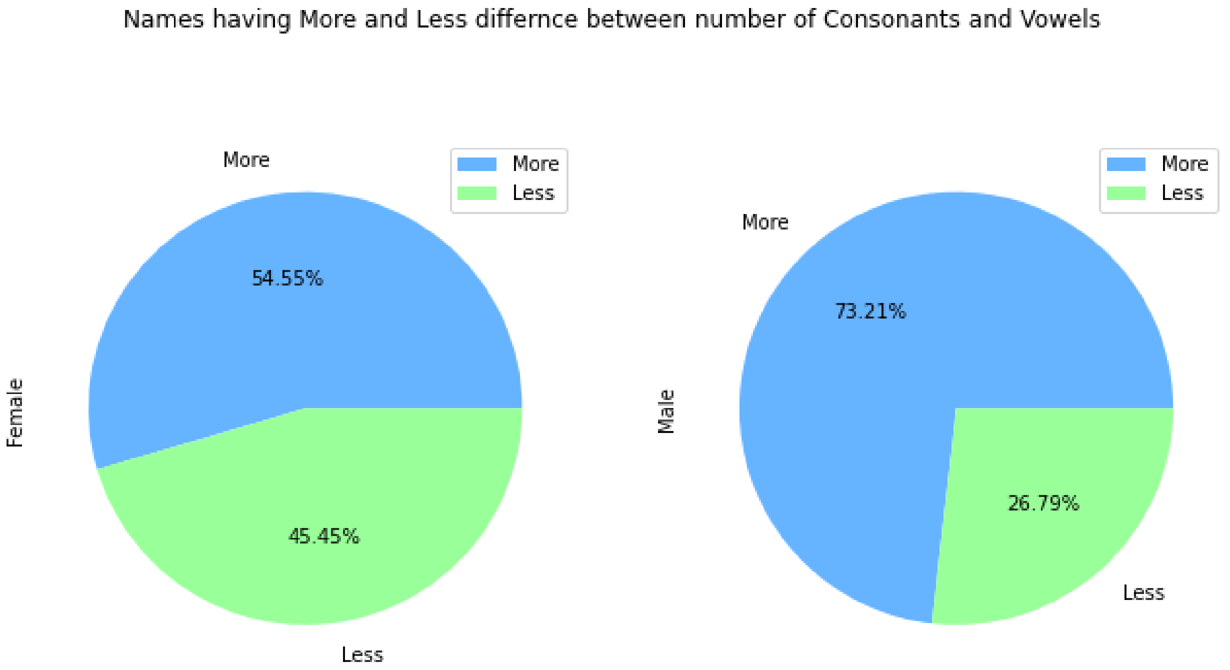




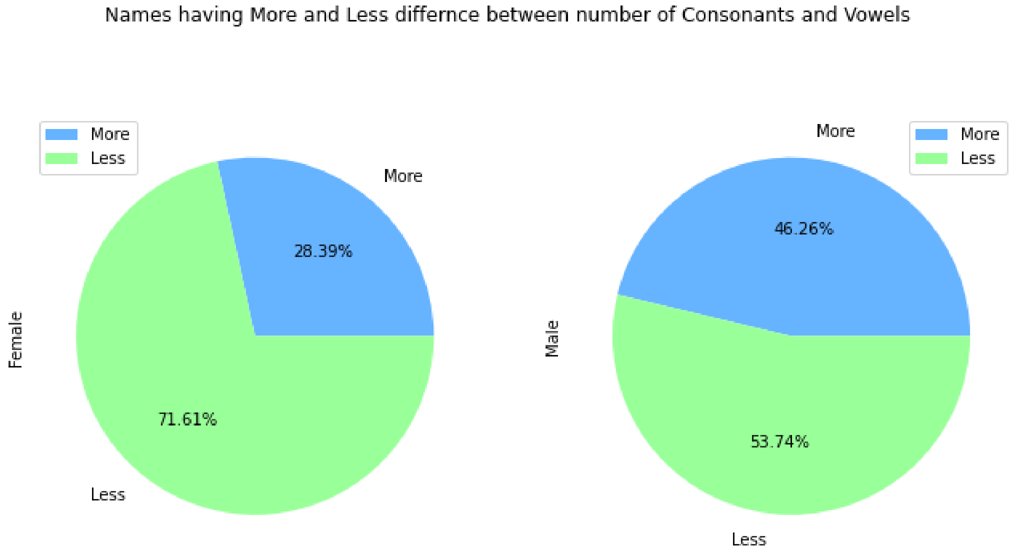
We realised that both male and female names show similar results, so a threshold couldn't be obtained using these graphs. So, to confirm the statistics we classified using threshold as zero. As predicted, choosing <0 as a threshold gave the same result as the previous feature.

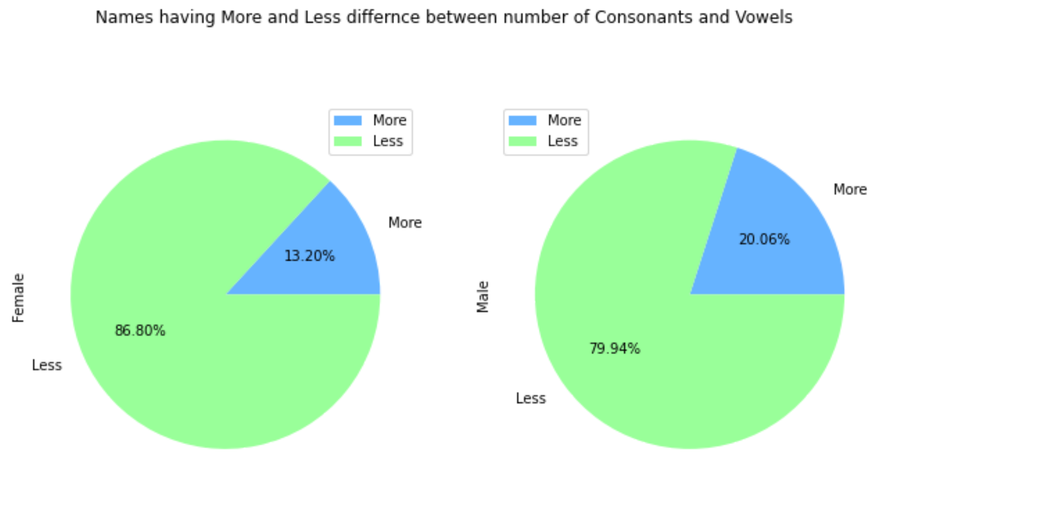
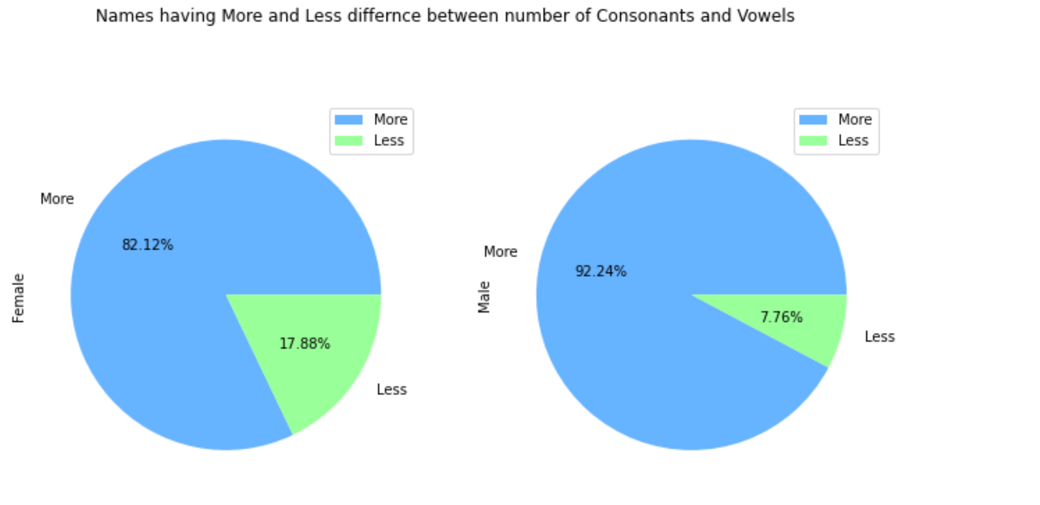
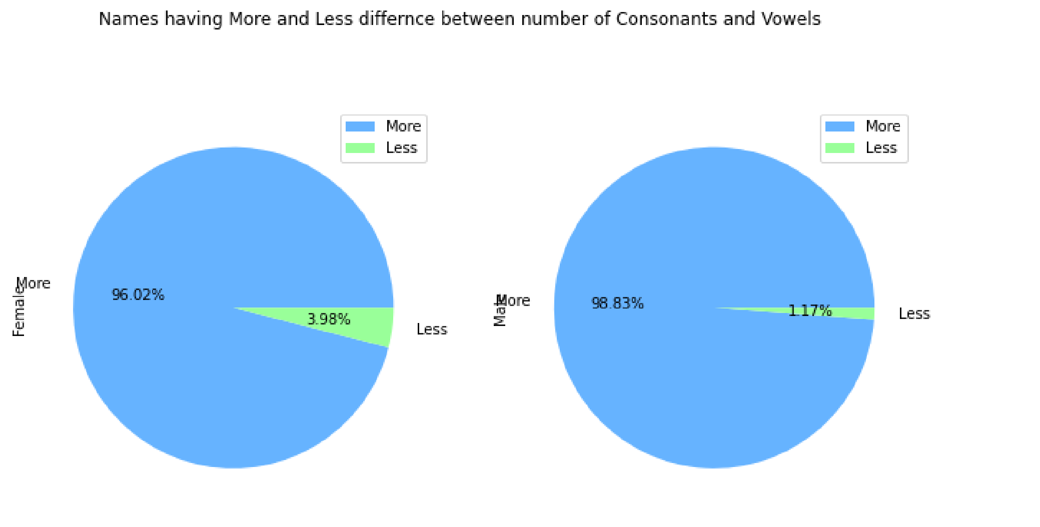
Why? Choosing threshold zero meant, choosing names with difference 0 or less as less difference which implies an equal number of vowels and consonants or more vowels than consonants. Which is exactly what the previous feature dealt with.

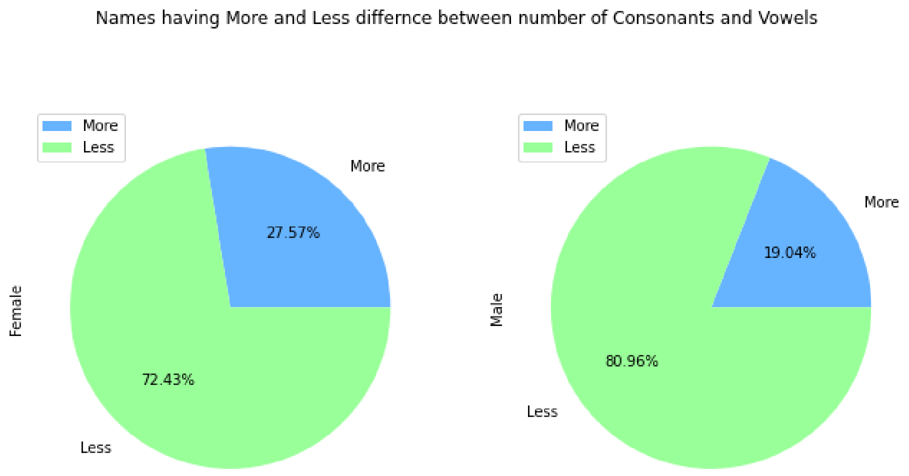
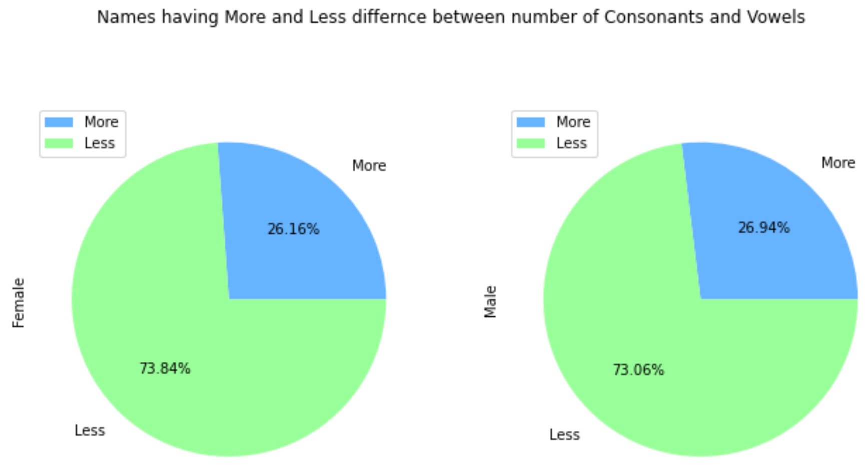
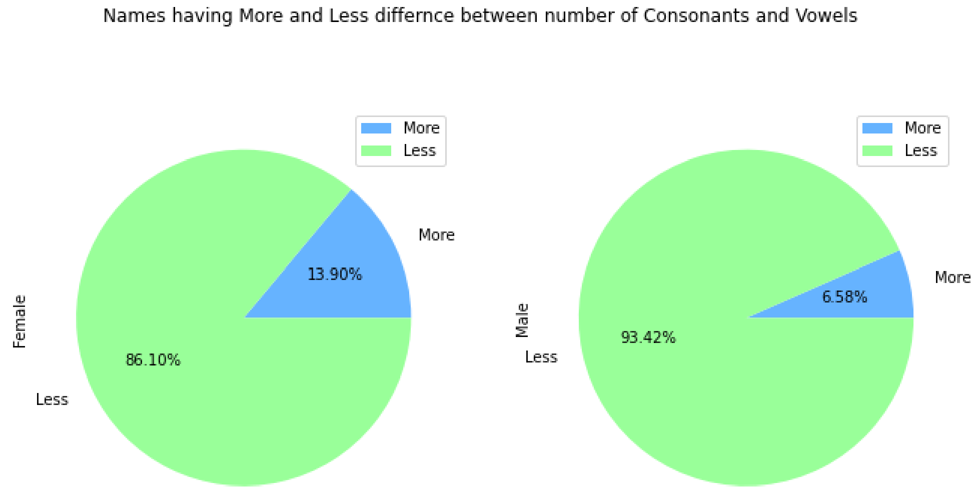
Therefore, instead of creating a feature by putting a threshold, we can use the difference of vowels and consonants itself as a feature.

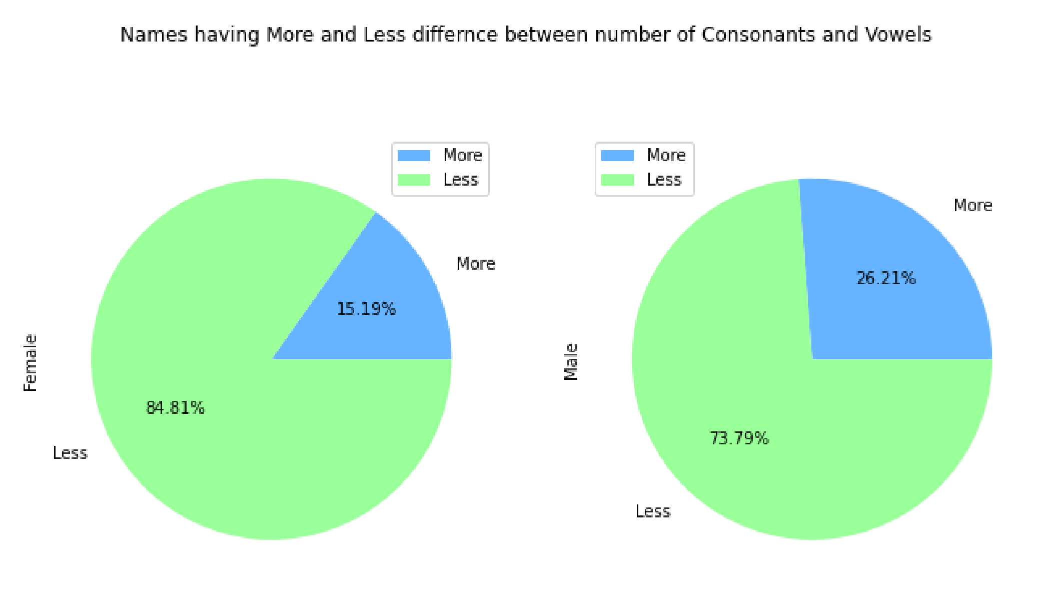

Names having More and Less differnce between number of Consonants and Vowels

So, we thought we should drop the 'MoreDifferenceClass' feature now. But we tried experimenting and using a different threshold randomly to see if there is any difference in the distribution.

Threshold	**Distribution Plot**
>1	 Names having More and Less differnce between number of Consonants and Vowels

>2	 Names having More and Less differnce between number of Consonants and Vowels More Less Female More 13.20% 86.80% Less More Less Male More 20.06% 79.94% Less
>-1	 Names having More and Less differnce between number of Consonants and Vowels More Less Female More 82.12% 17.88% Less More Less Male More 92.24% 7.76% Less
>-2	 Names having More and Less differnce between number of Consonants and Vowels More Less Female More 96.02% 3.98% Less More Less Male More 98.83% 1.17% Less

=0	Names having More and Less differnce between number of Consonants and Vowels More Less Female More 27.57% 72.43% Less Male More 19.04% 80.96% Less
=1	Names having More and Less differnce between number of Consonants and Vowels More Less Female More 26.16% 73.84% Less Male More 26.94% 73.06% Less
=-1	Names having More and Less differnce between number of Consonants and Vowels More Less Female More 13.90% 86.10% Less Male More 6.58% 93.42% Less

=2	

After the experiment, we decided to keep >1 as the threshold and use this feature.

Feature 9: ASCII Values

For the last feature to be extracted from first names, we simply used the ASCII values of the names.

Rejected Features: More features like BeginsWithVowel, LengthofName etc. were extracted but they showed the same results for male and female names hence these were rejected.

4.2.2 Feature Extraction by Dialogues

The following features were also extracted from dialogues, this helped us take into account a semantic and structural view of dialogues which also helps to see whether male dialogues differ from female dialogues.

4.2.2.1 Features related to sentiment

Feature 1: Polarity Score

Sentiment polarity for an element defines the orientation of the expressed sentiment, i.e., it determines if the text expresses the positive, negative or neutral sentiment of the user about the entity in consideration. Polarity lies between [-1,1], -1 defines a negative sentiment and 1 defines a positive sentiment. Negation words reverse the polarity.

We used the TextBlob library for calculating the Polarity Score. TextBlob has semantic labels that help with fine-grained analysis. For example — emoticons, exclamation mark, emojis, etc.

Feature 2: Subjectivity Score

Subjectivity quantifies the amount of personal opinion and factual information contained in the text. The higher subjectivity means that the text contains personal opinion rather than factual information. Subjectivity lies between [0,1].

We used the TextBlob library for calculating the Subjectivity Score. TextBlob has one more parameter: intensity. TextBlob calculates subjectivity by looking at the 'intensity'. Intensity determines if a word modifies the next word.

Feature 3: Sentiment Score

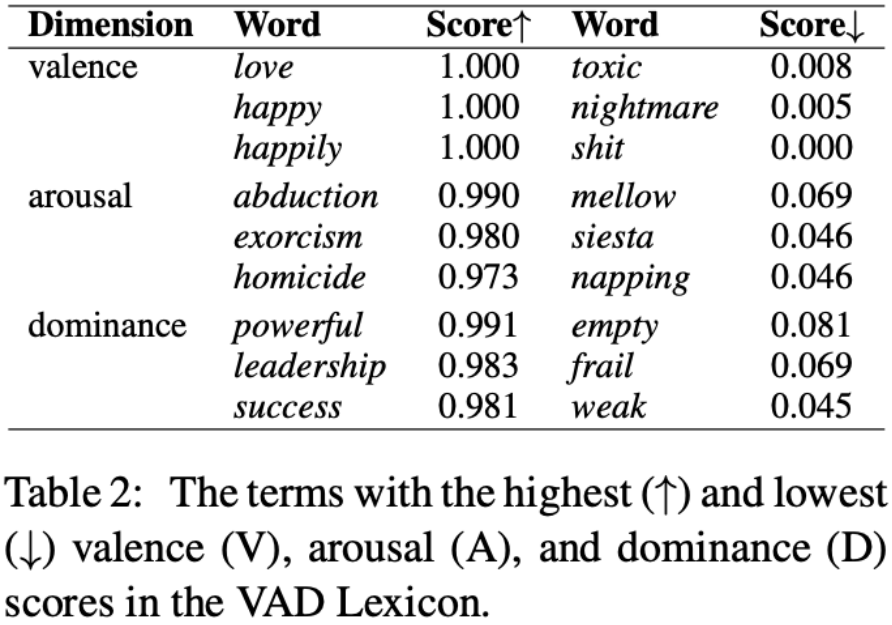
This score was just a classification based on the polarity score into the classes positive, negative or neutral.

Feature 4, 5 and 6: Valence, Arousal and Dominance

Three components of emotions are traditionally distinguished: valence (the pleasantness of a stimulus), arousal (the intensity of emotion provoked by a stimulus), and dominance (the degree of control exerted by a stimulus).

VAD or Valence, Arousal and Dominance value for each dialogue was calculated. We used already scored 20,000 English words[2] to do this task. We removed stopped words from the dialogues and lemmatized each word in the dialogue before matching it to this VAD dictionary. For words that were not given in this list of 20k words, the VAD score was set to 0.

[2] https://www.aclweb.org/anthology/P18-1017.pdf



Dimension	Word	Score↑	Word	Score↓
valence	*love*	1.000	*toxic*	0.008
	happy	1.000	*nightmare*	0.005
	happily	1.000	*shit*	0.000
arousal	*abduction*	0.990	*mellow*	0.069
	exorcism	0.980	*siesta*	0.046
	homicide	0.973	*napping*	0.046
dominance	*powerful*	0.991	*empty*	0.081
	leadership	0.983	*frail*	0.069
	success	0.981	*weak*	0.045

Table 2: The terms with the highest (↑) and lowest (↓) valence (V), arousal (A), and dominance (D) scores in the VAD Lexicon.

4.2.2.2 Structural Features

Feature 1: Average tokens per dialogue

For each dialogue of each speaker, the average number of tokens per dialogue was calculated. This value was averaged to find the average tokens per dialogue for each speaker.

Feature 2: Average token length

For each dialogue, we calculated the average length of words in that dialogue. This value was averaged to find the average token length for each speaker.

Feature 3: Type to Token Ratio (TTR)

Type-token ratio (TTR) computed as t/w, where t is the number of unique terms/vocab, and w is the total number of words.[3]

Feature 4: Measure of Textual Lexical Diversity(MTLD)

The measure of textual lexical diversity is computed as the mean length of sequential words in a text that maintains a minimum threshold TTR score. Iterates over words until TTR scores fall below a threshold, then increase factor counter by 1 and start over.

[3] (Chotlos 1944, Templin 1957)

McCarthy and Jarvis (2010, pg. 385) recommend a factor threshold in the range of [0.660, 0.750].

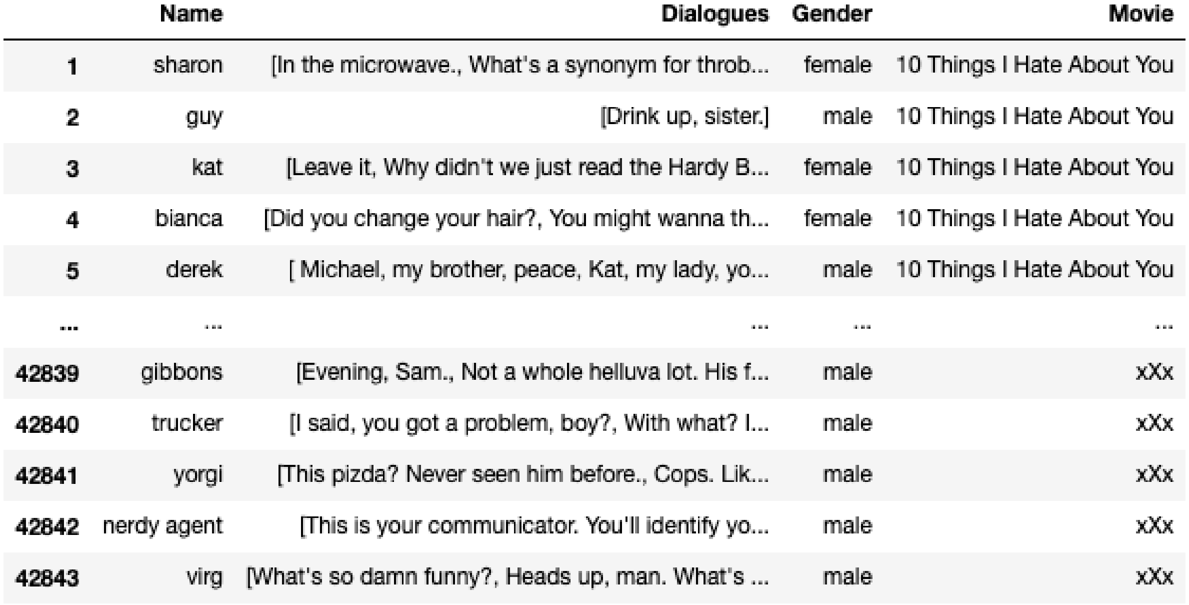
Feature 5: Hypergeometric distribution diversity (HD-D) score.

For each term (t) in the text, compute the probability(p) of getting at least one appearance of t with a random draw of size n < N (text size). The contribution of t to the final HD-D score is p * (1/n). The final HD-D score thus sums over p * (1/n) with p computed for each term t. [4]

4.3 Training and Testing Models using Gender Data

4.3.1 Features

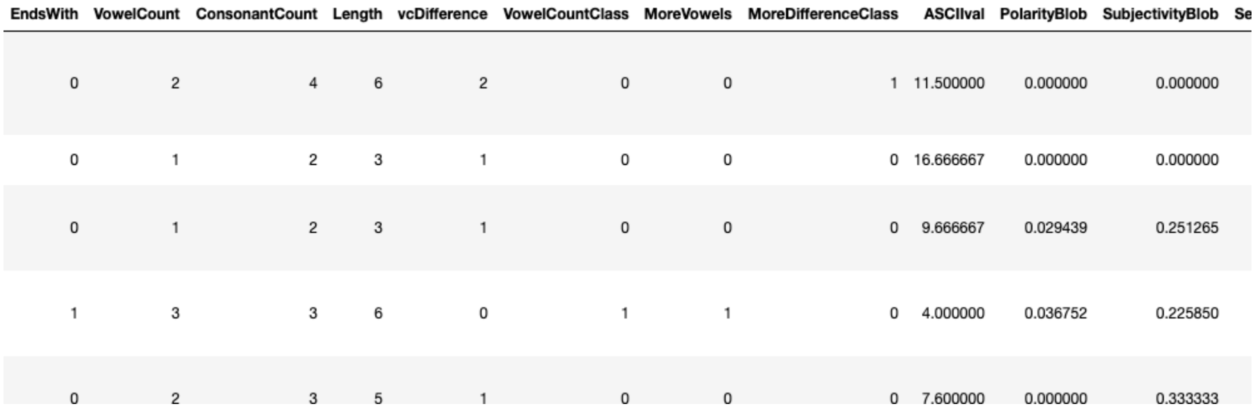
All accepted features from section 4.2 were calculated for our dataset which was compiled at the end of section 4.1 and looks like this:



	Name	Dialogues	Gender	Movie
1	sharon	[In the microwave., What's a synonym for throb...	female	10 Things I Hate About You
2	guy	[Drink up, sister.]	male	10 Things I Hate About You
3	kat	[Leave it, Why didn't we just read the Hardy B...	female	10 Things I Hate About You
4	bianca	[Did you change your hair?, You might wanna th...	female	10 Things I Hate About You
5	derek	[Michael, my brother, peace, Kat, my lady, yo...	male	10 Things I Hate About You
...	...	...	...	...
42839	gibbons	[Evening, Sam., Not a whole helluva lot. His f...	male	xXx
42840	trucker	[I said, you got a problem, boy?, With what? I...	male	xXx
42841	yorgi	[This pizda? Never seen him before., Cops. Lik...	male	xXx
42842	nerdy agent	[This is your communicator. You'll identify yo...	male	xXx
42843	virg	[What's so damn funny?, Heads up, man. What's ...	male	xXx

A snippet of the data after applying all features on it:

[4] Described in McCarthy and Javis 2007, p.g. 465-466. (McCarthy and Jarvis 2007)

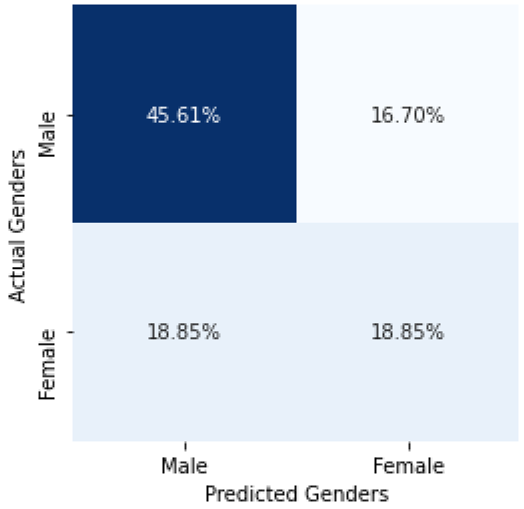


EndsWith	VowelCount	ConsonantCount	Length	vcDifference	VowelCountClass	MoreVowels	MoreDifferenceClass	ASCIIval	PolarityBlob	SubjectivityBlob	Se
0	2	4	6	2	0	0	1	11.500000	0.000000	0.000000	
0	1	2	3	1	0	0	0	16.666667	0.000000	0.000000	
0	1	2	3	1	0	0	0	9.666667	0.029439	0.251265	
1	3	3	6	0	1	1	0	4.000000	0.036752	0.225850	
0	2	3	5	1	0	0	0	7.600000	0.000000	0.333333	

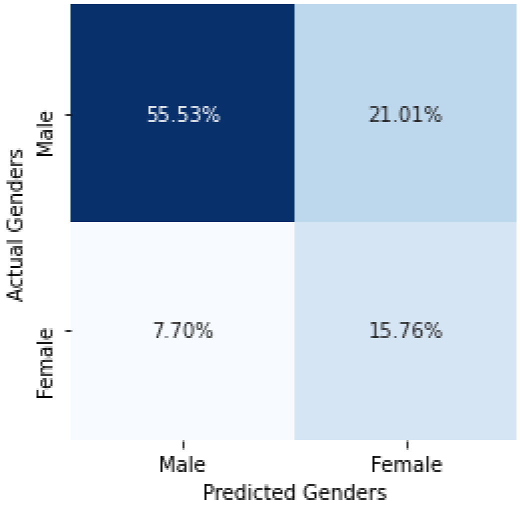
4.3.2 Test and Train Split and Results

The final set of features were divided into training and test sets in the ratio of 80:20.

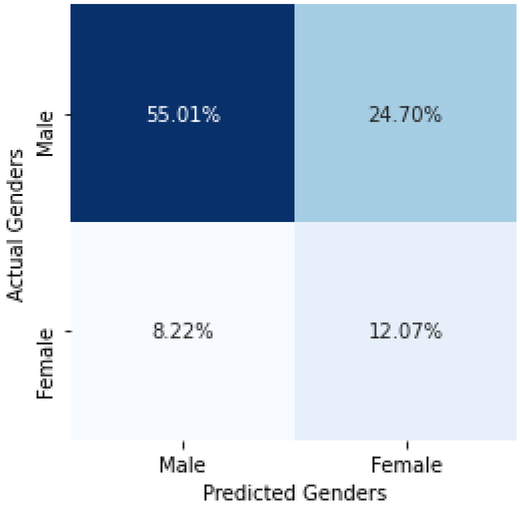
1. **Decision Tree Classifier:** After the model was trained, the importance of features were found for this model. The dialogue features and ASCII value were the most important features for this model. This model gave an accuracy of 64.5%. And the Confusion matrix is given below:



2. **K-means:** Accuracy obtained from this model was the least 50.8%.
3. **Random Forest Classifier:** This model performed the best of the lot and the predicted results had 71.3% accuracy and the confusion matrix is given below:



4. **Logistic Regression:** This model was also trained and tested on the same training and test sets to give an accuracy of 67.07%. And the Confusion matrix is given below:

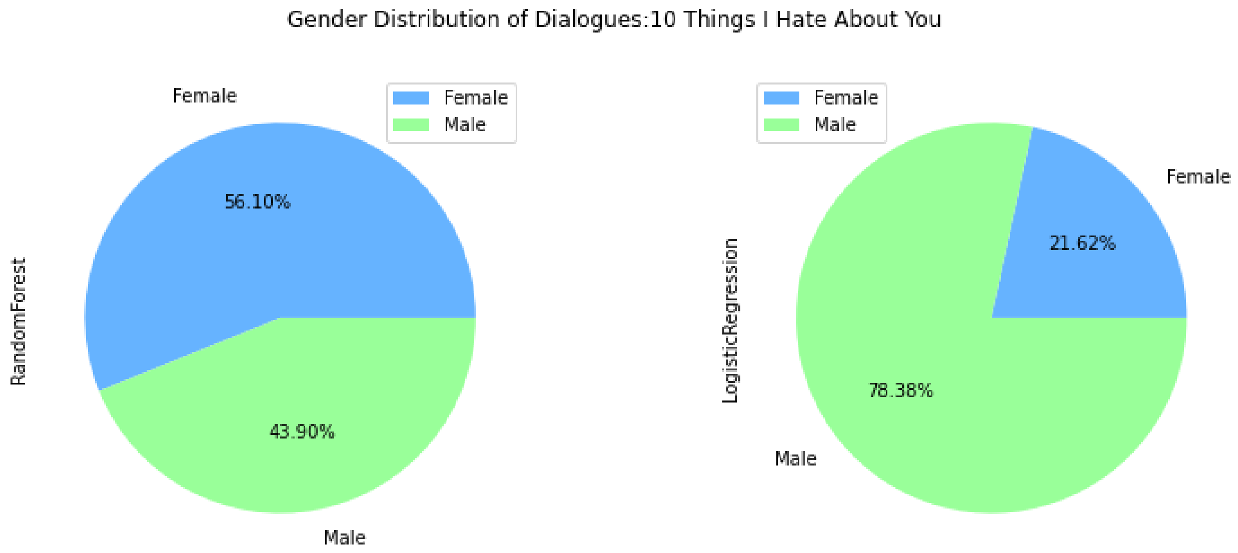


Hence, for our main task, RandomForest and LogisticRegression were used.

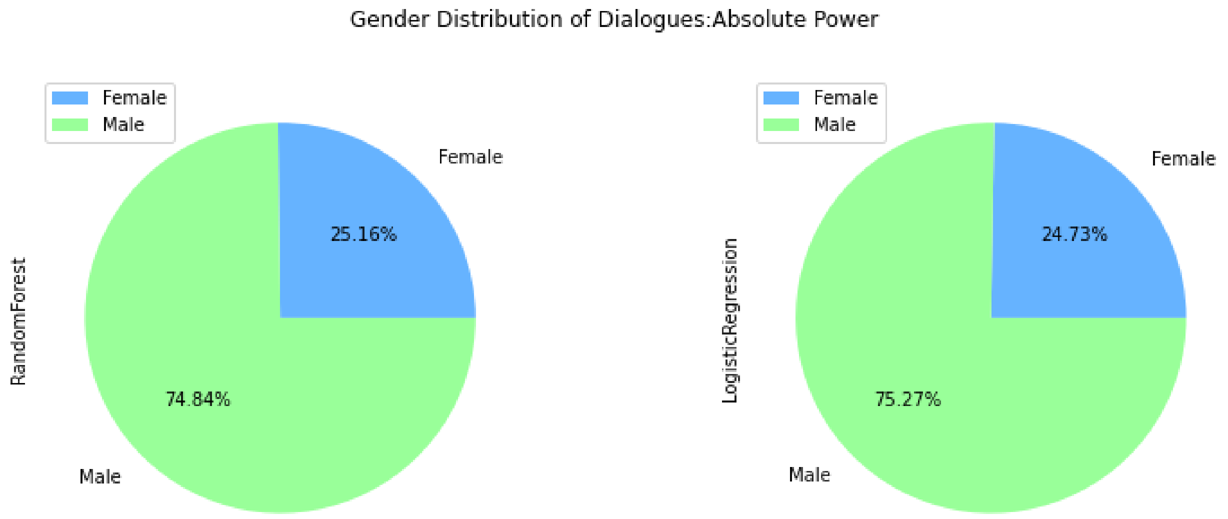
4.4 Gender distribution across dialogues

We used our trained model to predict genders for some movies' speakers and their dialogues and use that to calculate the number of dialogues spoken by male/female characters in a movie. The plot on the left shows distribution wrt prediction made by Random Forest Model and that on the right is Logistic Regression Model:

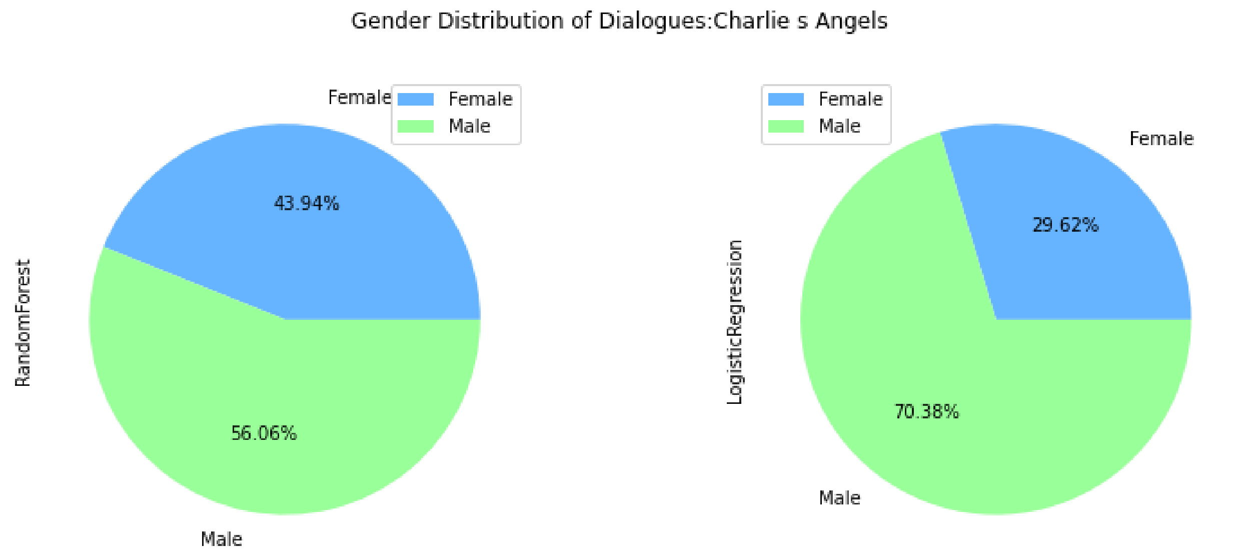
1. 10 Things I Hate About You - 1999 [Pudding: 54% Male | 46% Female]



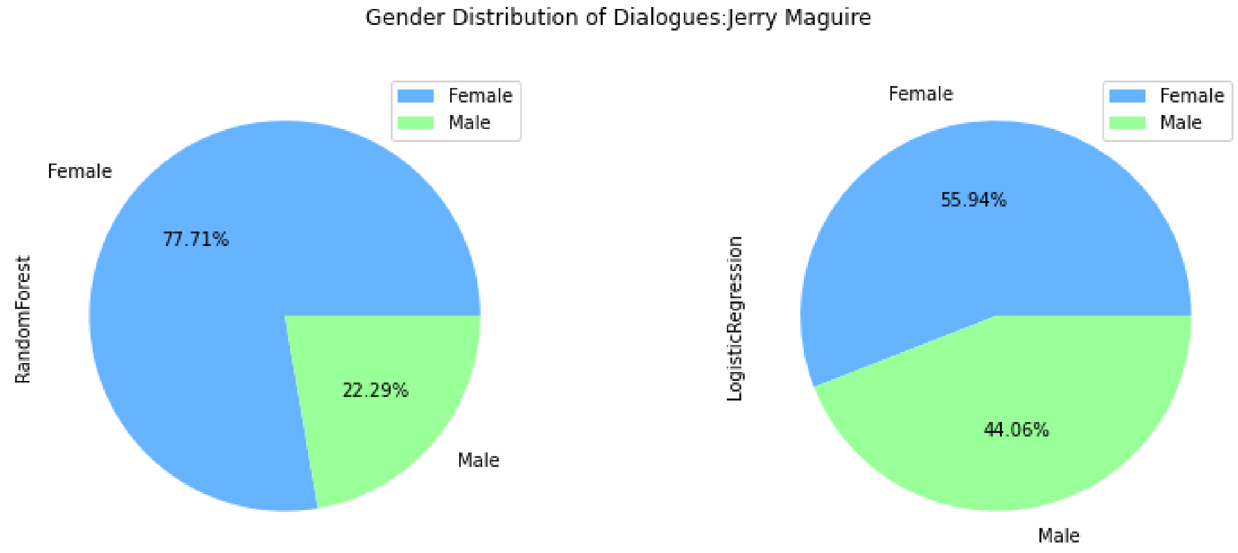
2. Absolute Power - 1997 [Pudding: 75% Male | 25% Female]



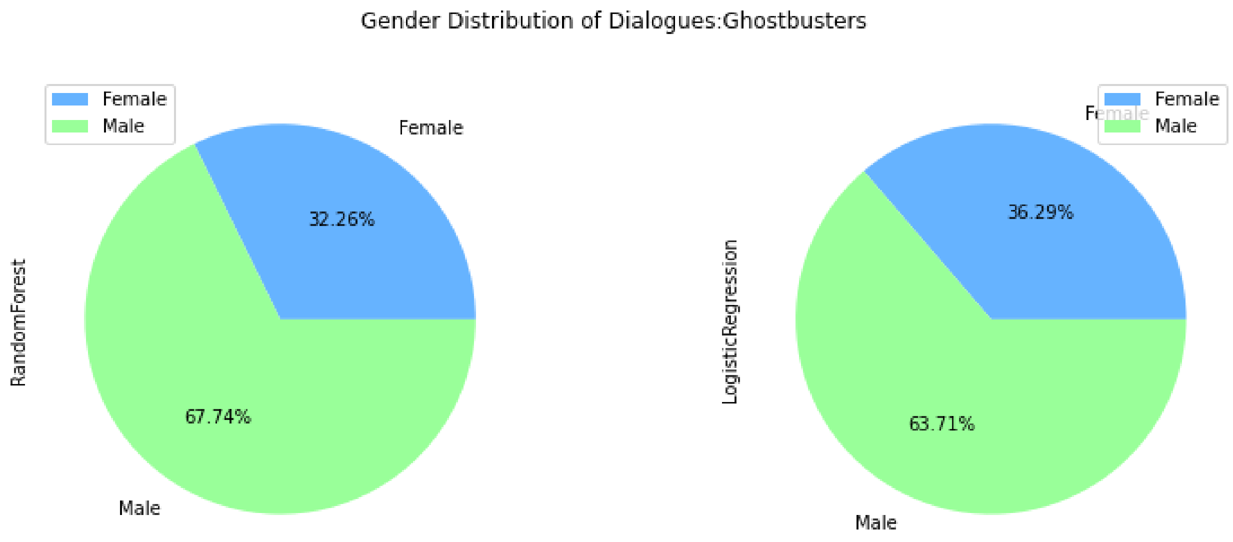
3. Chronicles of Narnia - 2005 [Pudding: 60% Male | 40% Female]



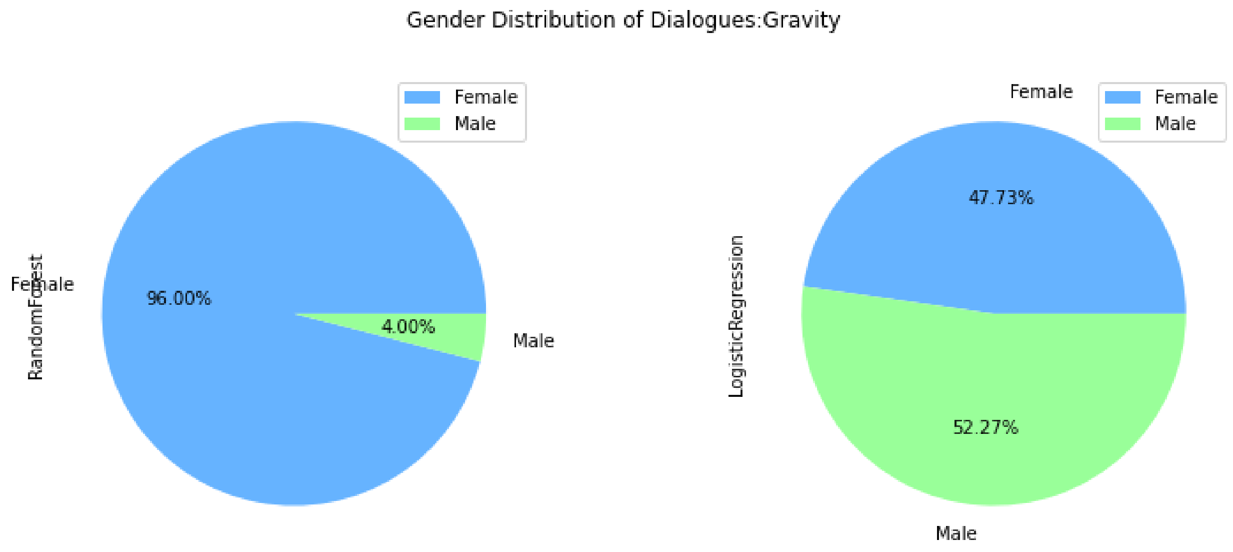
4. Jerry Maguire - 1996 [Pudding: 62% Male | 38% Female]



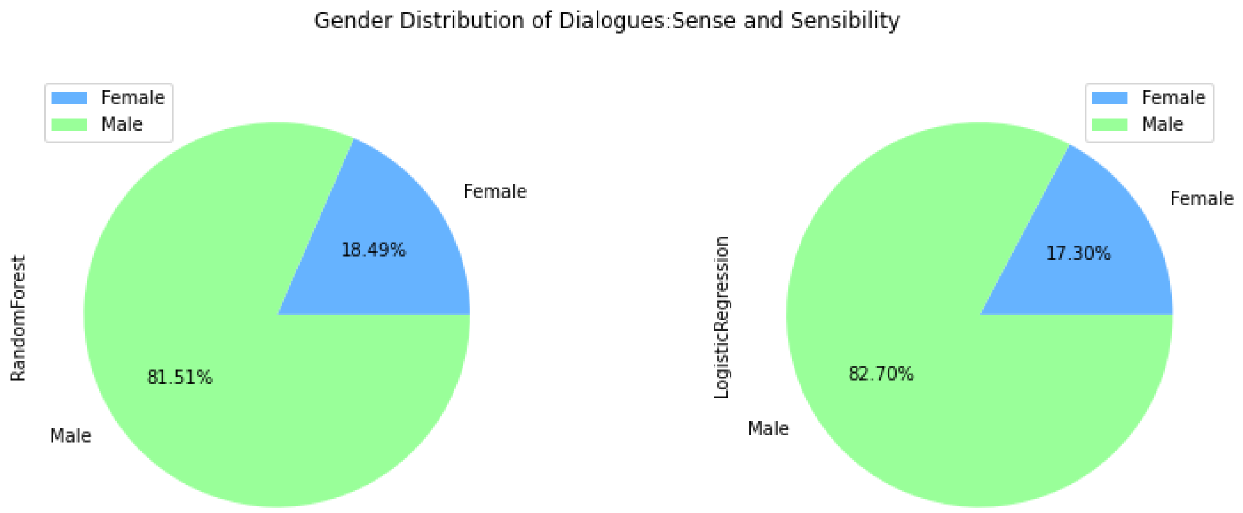
5. Ghostbusters - 1984 [Pudding: 85% Male | 15% Female]



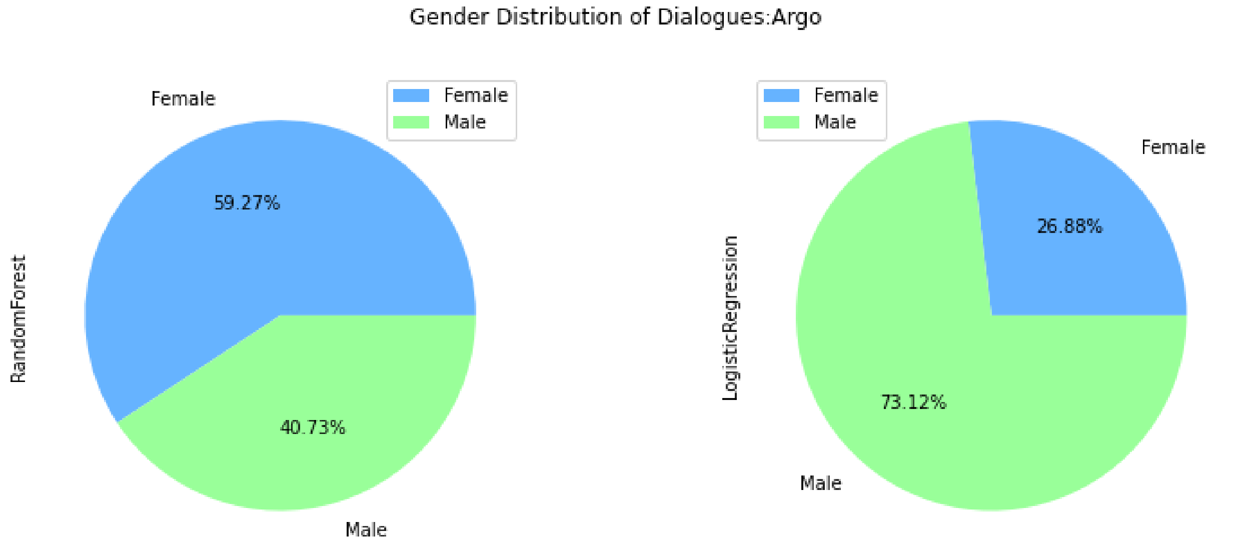
6. Gravity - 2013 [Pudding: 54% Male | 46% Female]



7. Sense and Sensibility - 1995 [Pudding: 29% Male | 71% Female]



8. Argo - 2012 [Pudding: 96% Male | 4% Female]



We can see that for some movies both models predict similar, a bit different or completely different results. On observing results for over ~30 randomly selected movies which are common to Pudding, we observe that the results given by RandomForest are proportional to Pudding - since they present the number of words, unlike our model which gives the number of dialogues. And when the RandomForest model fails, Logistic regression works for example for movies like Gravity.

It cannot be avoided that for some movies - whose count much less among the randomly picked movies - both models stray away from the Pudding distribution. But to confirm if they had really failed on the number of dialogues we performed a manual evaluation for such movies and it was found that they had really failed. An example of such a distribution is seen in Sense and Sensibility.

5 Gender Distribution of characters

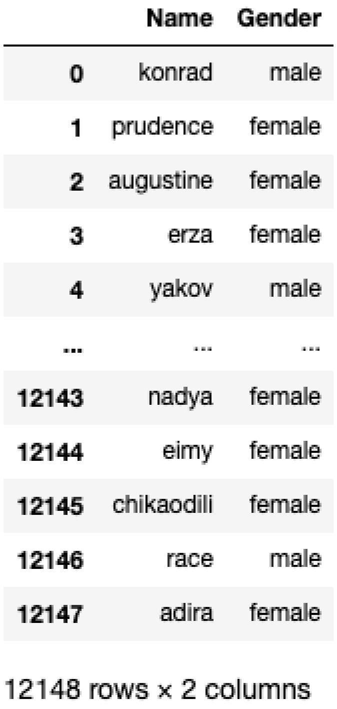
This milestone is not limited to our scripts dataset. The aim here is to provide the gender distribution of characters of any movie given its title and year (to distinguish movies with the same name).

The following steps were observed to complete this milestone:

	Tasks
1	Building a model to predict gender from first names
2	Procuring Cast of the movie
3	Preprocessing of the acquired data i.e cast
4	Getting gender distribution of characters of the movie and visualising the result

5.1 Model to predict gender from only first names

We collected around 12,000 common names and their gender from multiple resources to build our model.

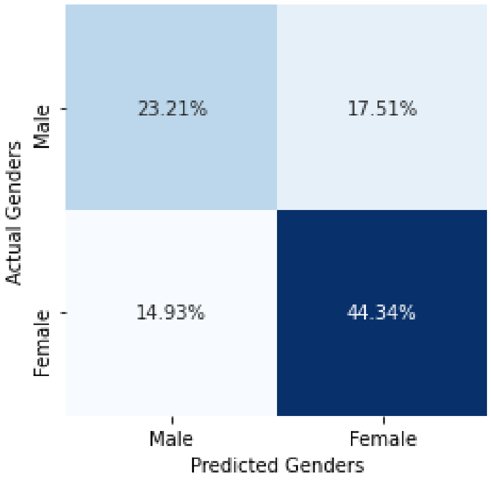


	Name	Gender
0	konrad	male
1	prudence	female
2	augustine	female
3	erza	female
4	yakov	male
...	...	...
12143	nadya	female
12144	eimy	female
12145	chikaodili	female
12146	race	male
12147	adira	female

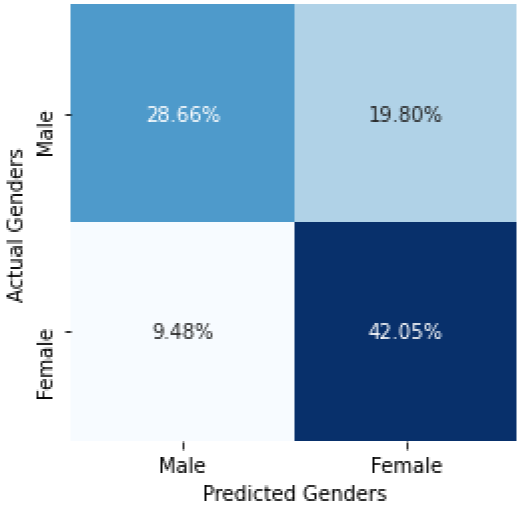
12148 rows × 2 columns

Many models were trained and tested using the features described in section 4.2.1. The final set of features were divided into training and test sets in the ratio of 80:20.

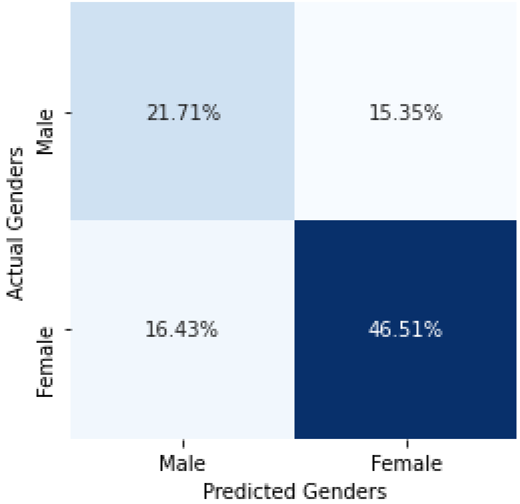
5. **Decision Tree Classifier:** After the model was trained, the importance of features were found for this model. The Vowel Ending and ASCII value were the most important features for this model. This model gave an accuracy of 67.5%. And the Confusion matrix is given below:



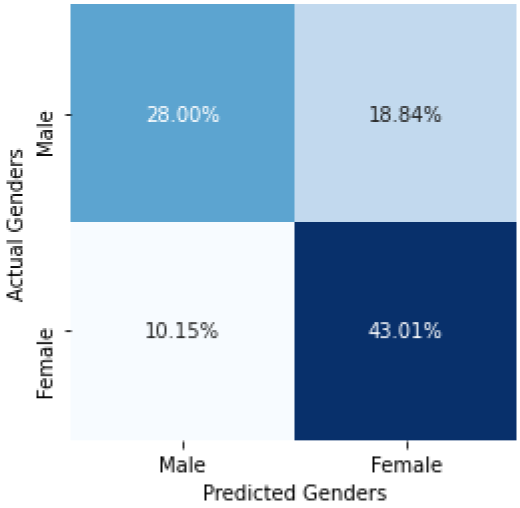
6. **Support Vector Machine(SVM)**: SVM was also trained and tested on the same training and test sets to give an accuracy of 70.7%. And the Confusion matrix is given below:



7. **K-means:** Accuracy obtained from this model was the least 60.3%.
8. **Random Forest Classifier:** This model performed third best with an accuracy of 68.21% and the confusion matrix is given below:



9. **Logistic Regression:** This model performed the best of the lot and the predicted results had 71% accuracy, which is quite close to SVM. The confusion matrix is given below:



Hence, for our main task, SVM and Logistic Regression were used.

5.2 Procuring cast of the movie given its title and year of release

To obtain the cast of a given movie, we used the library IMDbPy. However, to easily and uniquely identify a movie, we would require its IMDb ID. So, we start this task by finding the IMDb ID of any given movie and its year. This was done in three steps:

1. If the given movie is already present in our dataset of Characters wherein we have saved its IMDb ID, it can be simply extracted from there.
2. Else if the given movie is present in an IMDb dataset acquired from the web of 58k movies, its IMDb ID is collected from there.
3. Else the movie name is searched in the IMDbPy library using their search() function which lists all similar movies sorted in order of movies that most match the searched name, and if a movie is found with the same year, its IMDb ID is copied from there. This case happens very rarely.

Then the cast of the movie is found using its IMDb ID from the get_movie() function of the IMDbPy library. This returns an object containing the names of actors and their roles in this particular movie.

Since the returned movie object is not in the desired format, it is processed.

5.3 Preprocessing of the acquired data i.e cast

The object containing the list of actors and characters is processed in the following ways:

1. We check for multiple roles. It was observed in many films that the same actor has performed multiple roles or that the same character is being played by multiple actors. So, we separate each actor and their role to get a one-to-one mapping of them.
2. The movie object was converted to a string object and all unnecessary details and symbols were removed to obtain only a list of actors mapped to their Character.
3. It was observed that it is difficult to obtain the gender of common nouns like ‘doctor’, ‘professor’, ‘soldier1’, ‘soldier2’ etc. which are the names of characters in a lot of movies. And since our models have been trained on first names it will be meaningless to use them to predict the gender of these words, when even humans cannot predict the same. To handle this issue, we made an assumption:

   Each actor plays a character who has the same gender as the actor.

   Using this assumption, we decided to use the names of actors instead of the names of characters they played. Since we are looking at all characters of a movie, using either will work as long as the assumption holds. So, we picked only the names of actors from the cast.
4. After getting the names of the actors, we had one last and most important thing to do. Since our model is trained only on first names because last names can be common and for features that use the property: whether a name ends with a vowel, we required only first names and no last names. So, we created a rule-based method that extracted first names from a given list of names. Some of the rules were as follows:
   a. First, check if the name has any salutations and strip them.

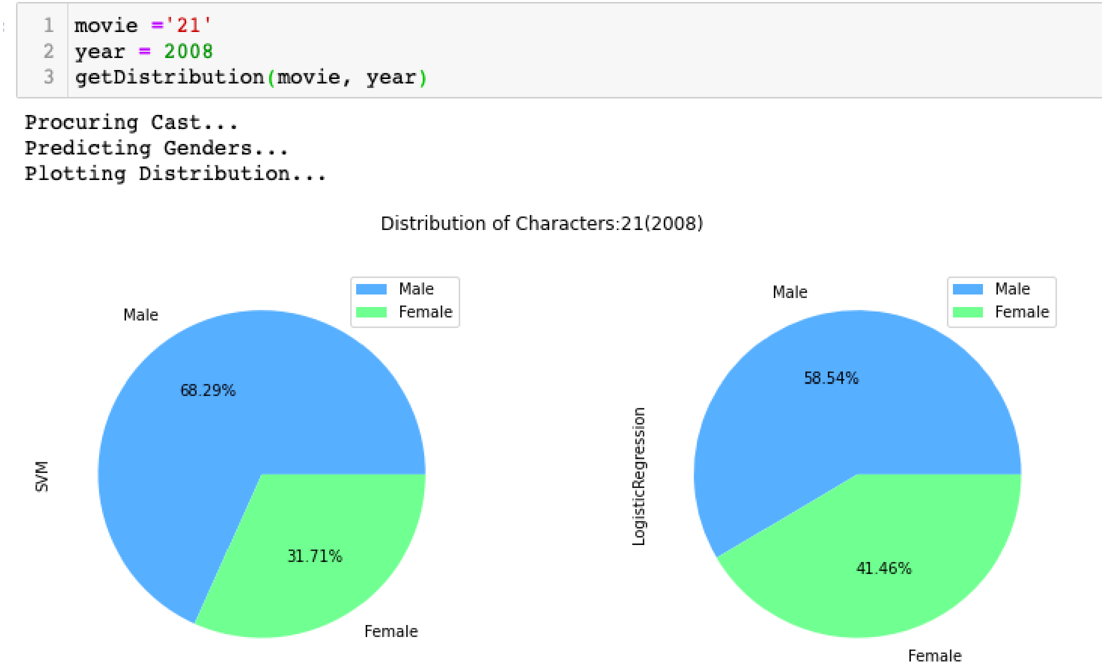
b. Then, we checked if the name had only one word. If so, this would imply in most cases that, this is the first name of the actor.
c. If the name was of length two, then, we used the spaCy library to get Part-Of-Speech tags of the names. And we checked if the first word was a Common Noun or a whitespace character, if so we chose the second word as the first name, otherwise the first word was chosen.
d. If the name was of length greater than 3, we chose the first word as the first name.

5.4 Getting gender distribution of characters of the movie and visualising the result

Once, we had the first names of the characters, all feature values were calculated for those names, and were fed to the already trained models to predict the genders of the given names. Finally, the predicted gender labels were used to find the gender distribution of characters of the given movie and a pie plot is used to show the result.

Here are the results of some movies: The left plot shows the results using the SVM Model, while the right shows results from the Logistic Regression model. The first image shows that to get the results we must enter the movie name and its year of release then run the file and once the script runs, in less than a minute the results will be displayed.

1. 21 - 2008

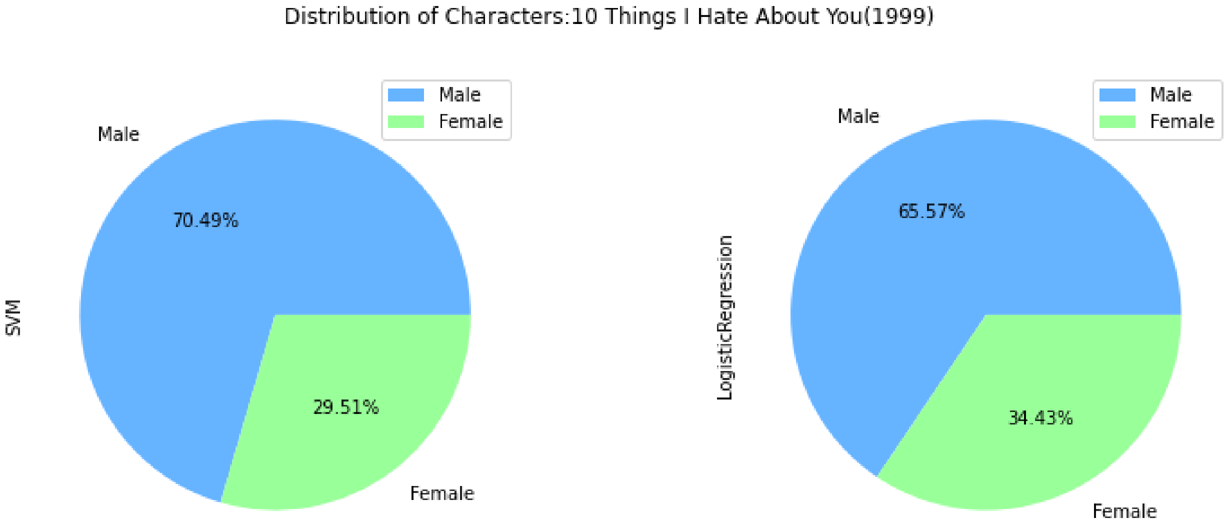


```
movie ='21'
year = 2008
getDistribution(movie, year)
```

```
Procuring Cast...
Predicting Genders...
Plotting Distribution...
```

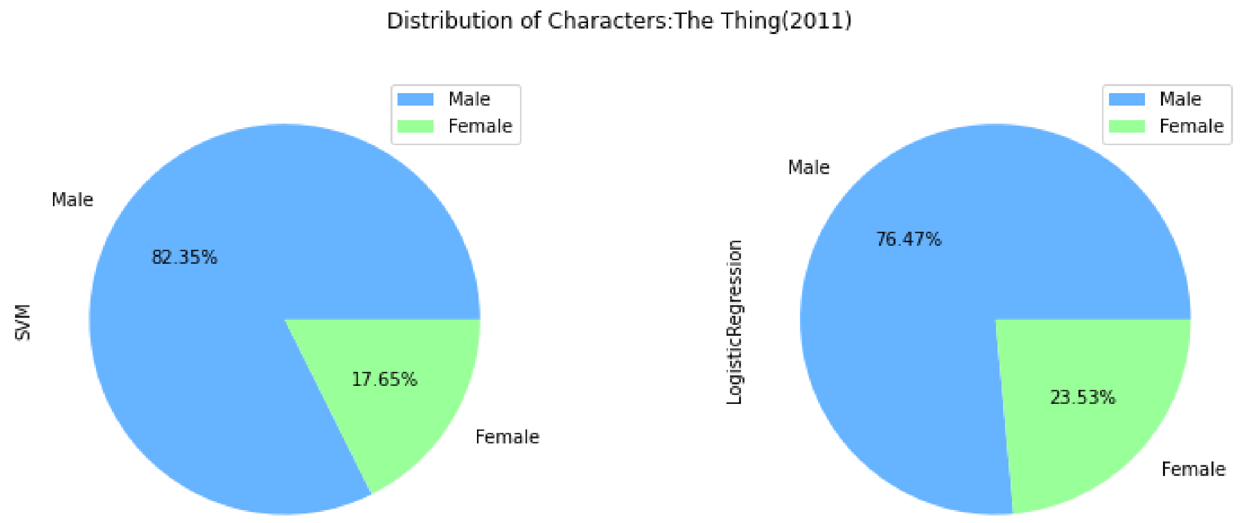
Distribution of Characters:21(2008)

2. 10 Things I Hate About You - 1999

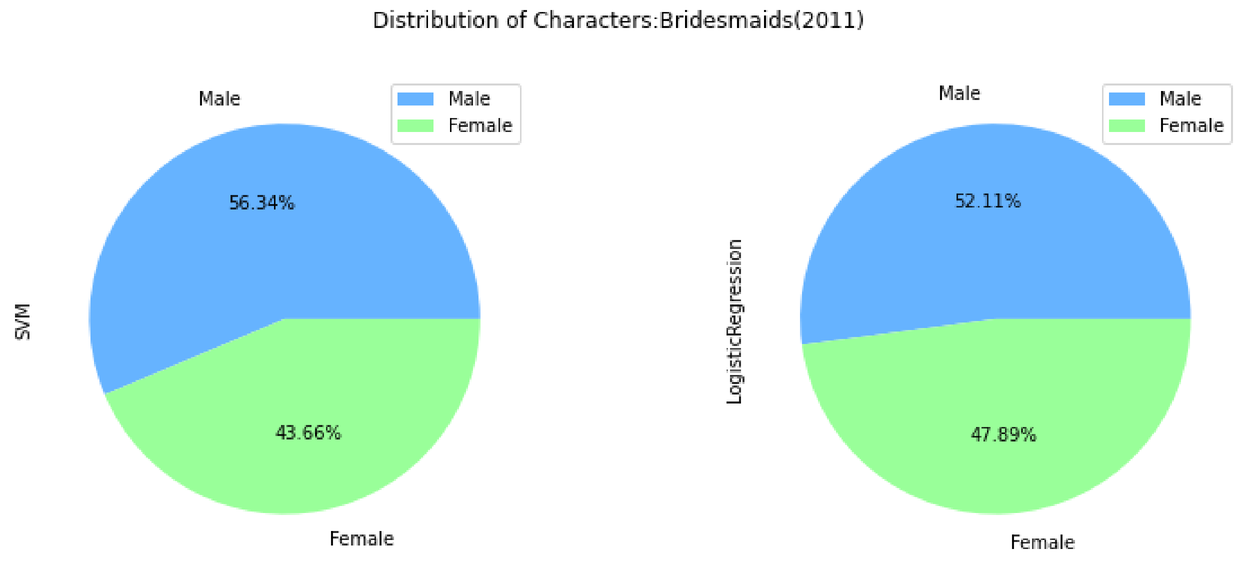


Distribution of Characters:10 Things I Hate About You(1999)

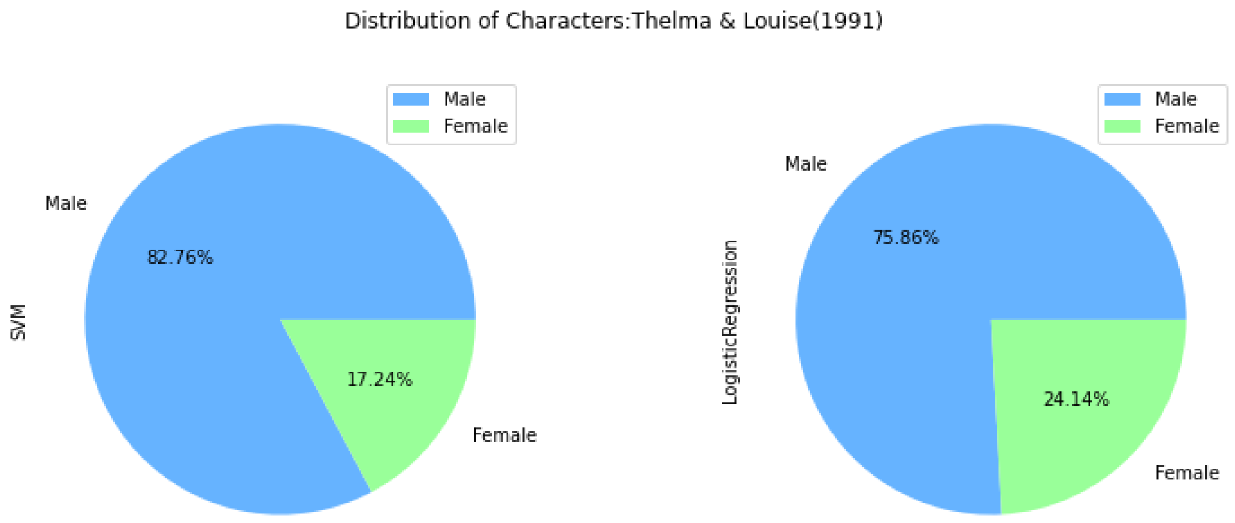
3. The Thing - 2011



4. Bridesmaids - 2011



5. Thelma & Louise - 1991



6 Gender Distribution of named characters

This milestone is also not limited to our scripts dataset. The aim here is to provide the gender distribution of named characters of any movie given its title and year (to distinguish movies with the same name).

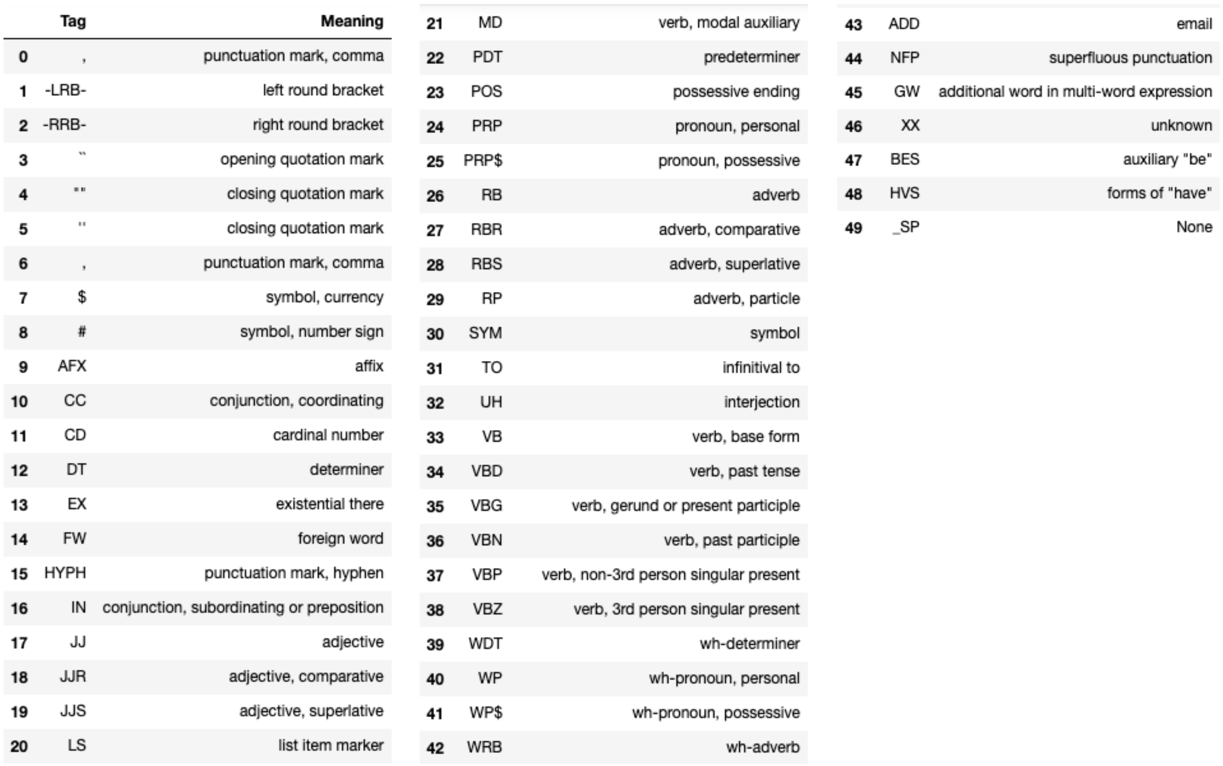
The following steps were observed to complete this milestone:

	Tasks
1	Building a model to predict gender from first names
2	Procuring Cast of the movie
3	Preprocessing of the acquired data i.e cast to get named characters
4	Getting gender distribution of characters of the movie and visualising the result

All tasks are similar to the previous milestone. Therefore, to avoid redundancy, only the new things will be discussed under this section which is from Task 3 onwards.

6.1 Preprocessing of the acquired dataset

1. Because we have to segregate only the named characters from all characters we cannot use the names of actors, but we must use the character names and remove those characters which have no proper names. Hence, from the cast object, we extracted only the character names this time.
2. Before we extract first names, we had to filter the character names to get only named characters. To do this task, we created a rule-based method:
    a. First, we checked if the character name was empty because while processing our scripts the accuracy was not 100%. If it was an empty string, we straightaway reject it.
    b. Then, we checked if the character's name has a number or a hashtag in it, as it indicated multiple background characters like 'Doctor1', 'Doctor#2'. Hence, such a name was rejected.
    c. Then, we checked if the character's name had a possessive relation of 's in it, as it indicated the character is a mere relation of a named character. For example, Jay's mom, April's teacher etc. Hence, such a name was also rejected.
    d. Then, we checked, the POS tag of each word in the character's name:
        i. If any word belonged to the following list of 50 tags, it was rejected.



	Tag	Meaning
0	,	punctuation mark, comma
1	-LRB-	left round bracket
2	-RRB-	right round bracket
3	``	opening quotation mark
4	""	closing quotation mark
5	''	closing quotation mark
6	,	punctuation mark, comma
7	$	symbol, currency
8	#	symbol, number sign
9	AFX	affix
10	CC	conjunction, coordinating
11	CD	cardinal number
12	DT	determiner
13	EX	existential there
14	FW	foreign word
15	HYPH	punctuation mark, hyphen
16	IN	conjunction, subordinating or preposition
17	JJ	adjective
18	JJR	adjective, comparative
19	JJS	adjective, superlative
20	LS	list item marker
21	MD	verb, modal auxiliary
22	PDT	predeterminer
23	POS	possessive ending
24	PRP	pronoun, personal
25	PRP$	pronoun, possessive
26	RB	adverb
27	RBR	adverb, comparative
28	RBS	adverb, superlative
29	RP	adverb, particle
30	SYM	symbol
31	TO	infinitival to
32	UH	interjection
33	VB	verb, base form
34	VBD	verb, past tense
35	VBG	verb, gerund or present participle
36	VBN	verb, past participle
37	VBP	verb, non-3rd person singular present
38	VBZ	verb, 3rd person singular present
39	WDT	wh-determiner
40	WP	wh-pronoun, personal
41	WP$	wh-pronoun, possessive
42	WRB	wh-adverb
43	ADD	email
44	NFP	superfluous punctuation
45	GW	additional word in multi-word expression
46	XX	unknown
47	BES	auxiliary "be"
48	HVS	forms of "have"
49	_SP	None

   ii. Else if all words in the character's name were common nouns, it was also rejected

  e. We compiled lists of nationality, jobs, kinship, animals and people common nouns. Then, we checked if either the entire character name or any part of it was present in the compiled lists. If so, it was rejected.
  f. If a name passed all these tests, it was accepted as a proper name.

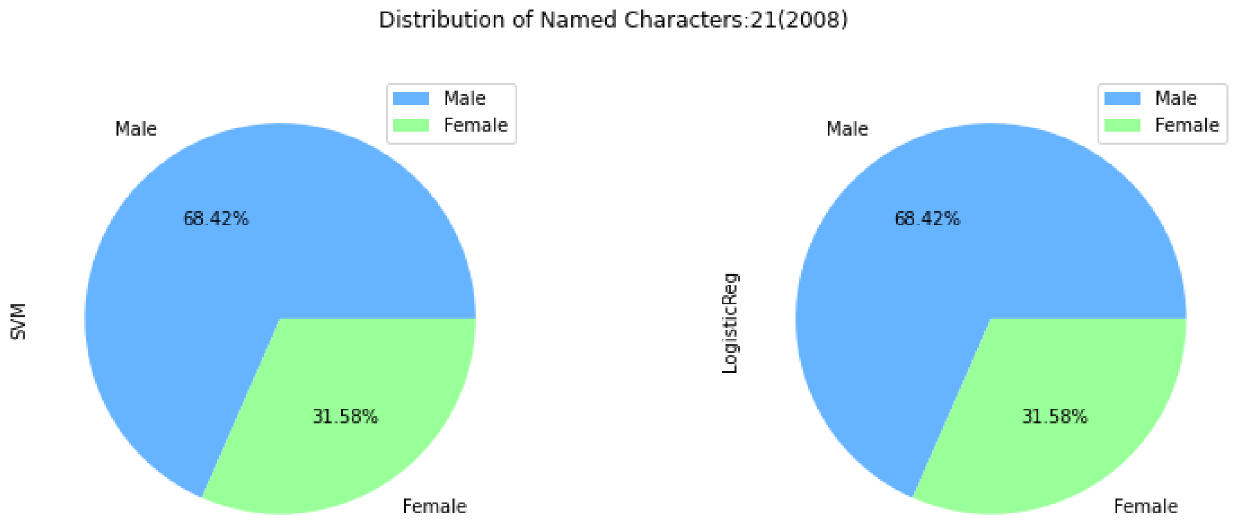
3. After this, the named character's names were processed to extract first names.

6.2 Getting gender distribution of named characters of the movie and visualising the result

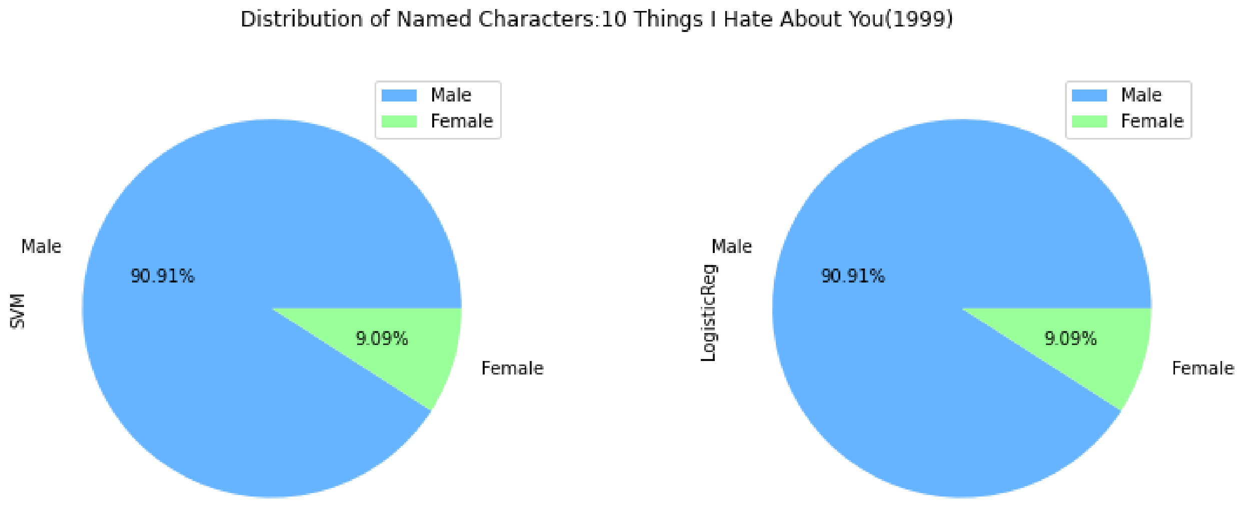
Once, we had the first names of the named characters, all feature values were calculated for those names, and were fed to the already trained models to predict the genders of the given names. Finally, the predicted gender labels were used to find the gender distribution of named characters of the given movie and a pie plot is used to show the result.

Here are the results of some movies: The left plot shows the results using the SVM Model, while the right shows results from the Logistic Regression model.

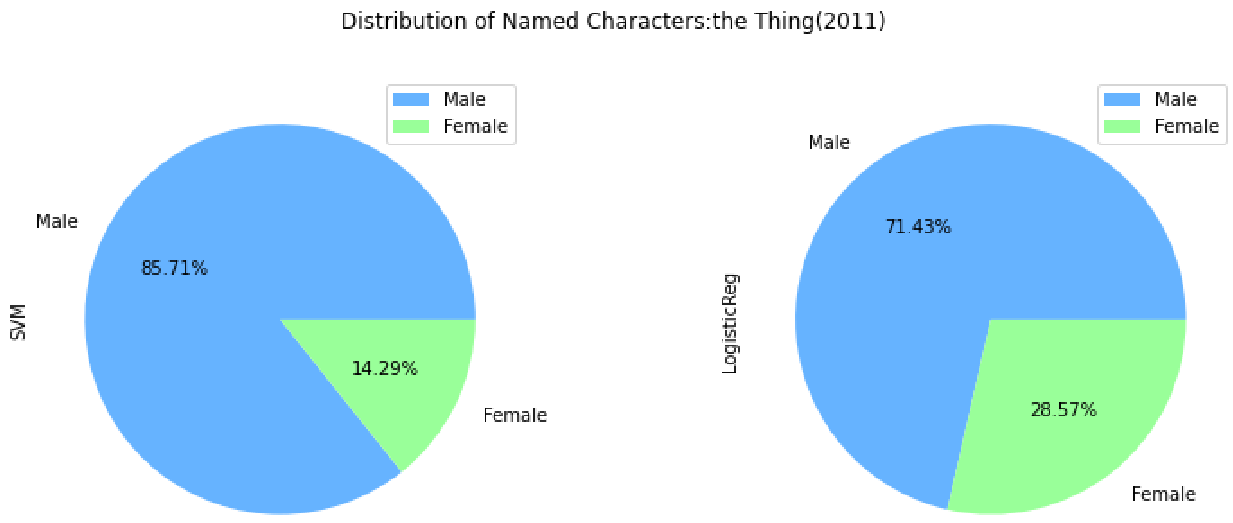
1. 21 - 2008



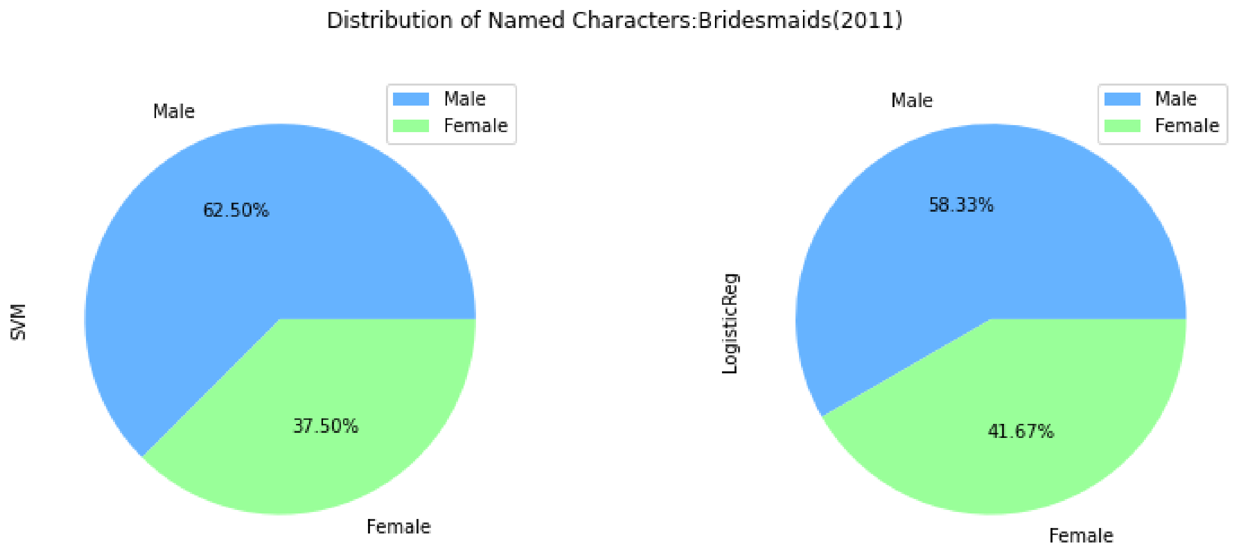
2. 10 Things I hate about you - 1999



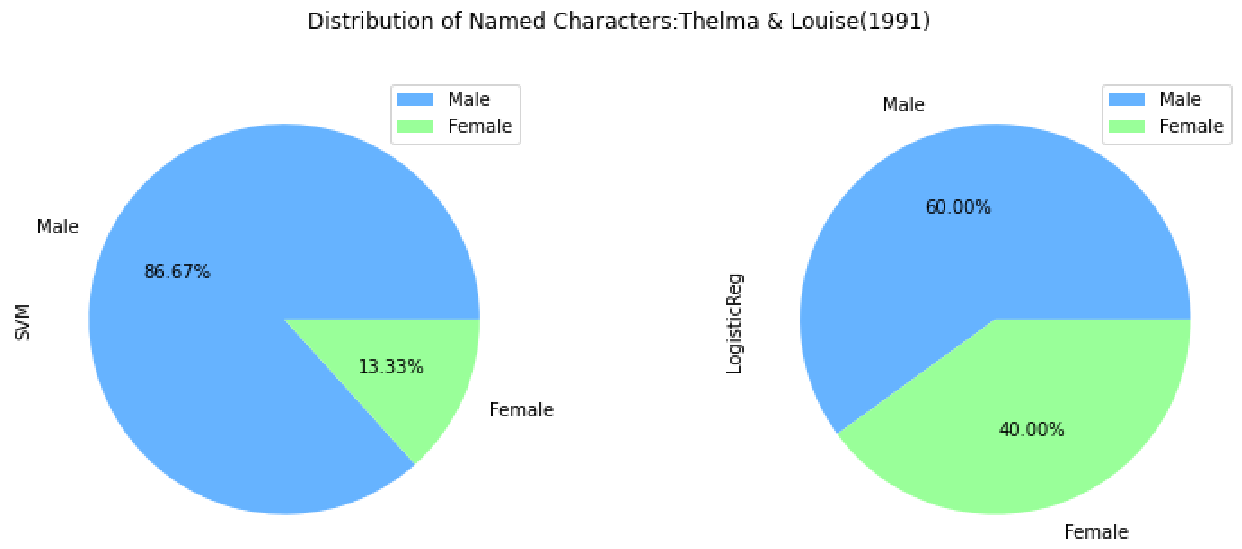
3. The Thing - 2011


Distribution of Named Characters:the Thing(2011)

4. Bridesmaids - 2011


Distribution of Named Characters:Bridesmaids(2011)

5. Thelma & Louise - 1991



Distribution of Named Characters:Thelma & Louise(1991)

7 Gender distribution of character with most dialogues in movies

<Ayushi>

8 Gender distribution of Speaking Roles

<Ayushi>

9 Links to files and data

Code	
Data	

10 Future Work

We have achieved a precision close to <Ayushi> while processing the scripts to generate speaker to dialogue mapping. Since the format of scripts varies a lot from script to script, a rule-based method with better precision than ours can be of help to much future research as most of the research today handles these scripts manually.

The models for gender distribution of dialogues in films can be improved by taking into account more features of greater importance. Even though we have included many features, their importance is not that high. So a proper study can be taken up to understand the importance of these features and how they can be improved.

11 References

11.1 List of Research papers read

1. (PDF) Gender-Distinguishing Features in Film Dialogue
2. Analyzing Gender Stereotyping in Bollywood Movies | Request PDF
3. [PDF] Linguistic analysis of differences in portrayal of movie characters
4. Detecting Gender Stereotypes: Lexicon vs. Supervised Learning Methods
5. When does a compliment become sexist? Analysis and classification of ambivalent sexism using twitter data
6. The Gender Gap Tracker: Using Natural Language Processing to measure gender bias in media
7. (PDF) Gender Prediction From Social Media Comments With Artificial Intelligence
8. Obtaining Reliable Human Ratings of Valence, Arousal, and Dominance for 20,000 English Words

11.2 Other references

1. Saif | VAD Lexicon
2. lexicalrichness · PyPI
3.