DATE-A-SCIENTIST

Machine Learning Fundamentals

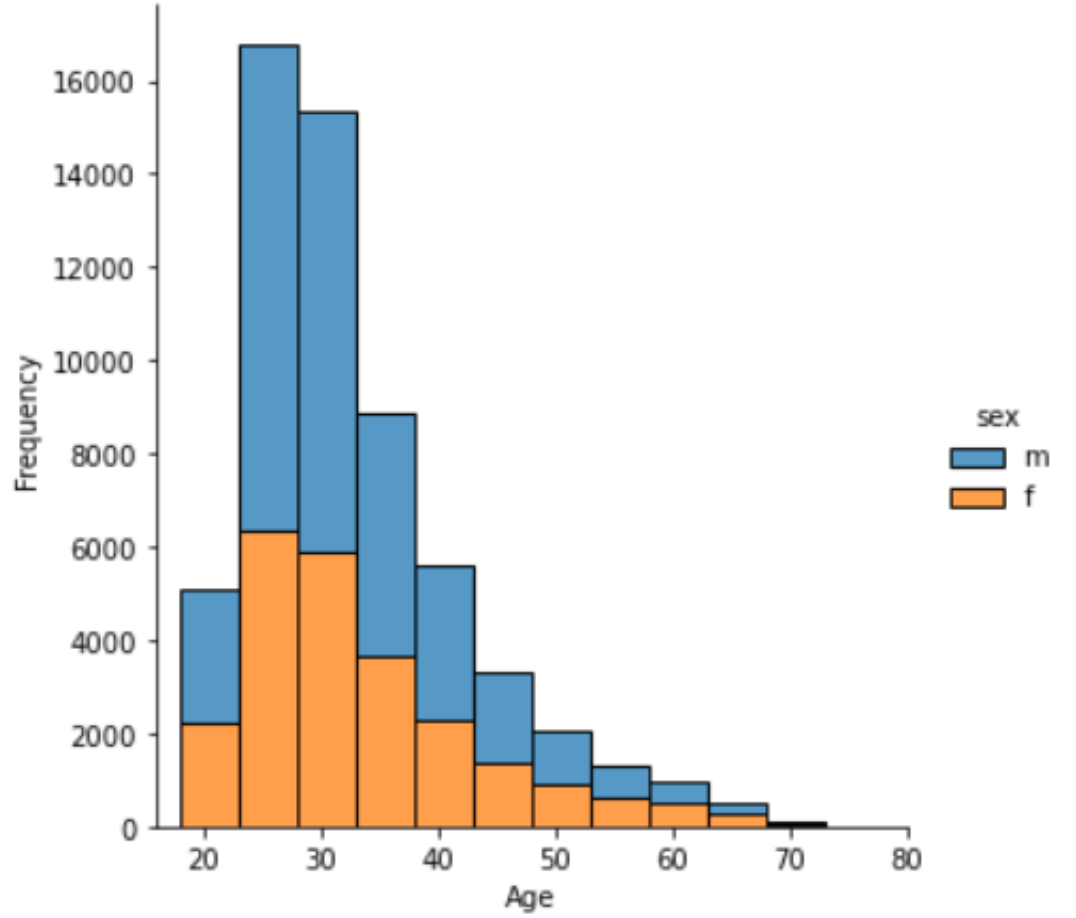
Ali Pakzad

TABLE OF CONTENTS

EXPLORATION OF THE DATASET

Most of the users are between 25 to 35 years old.

The distribution for both genders are the same, but there are more male users than female users.

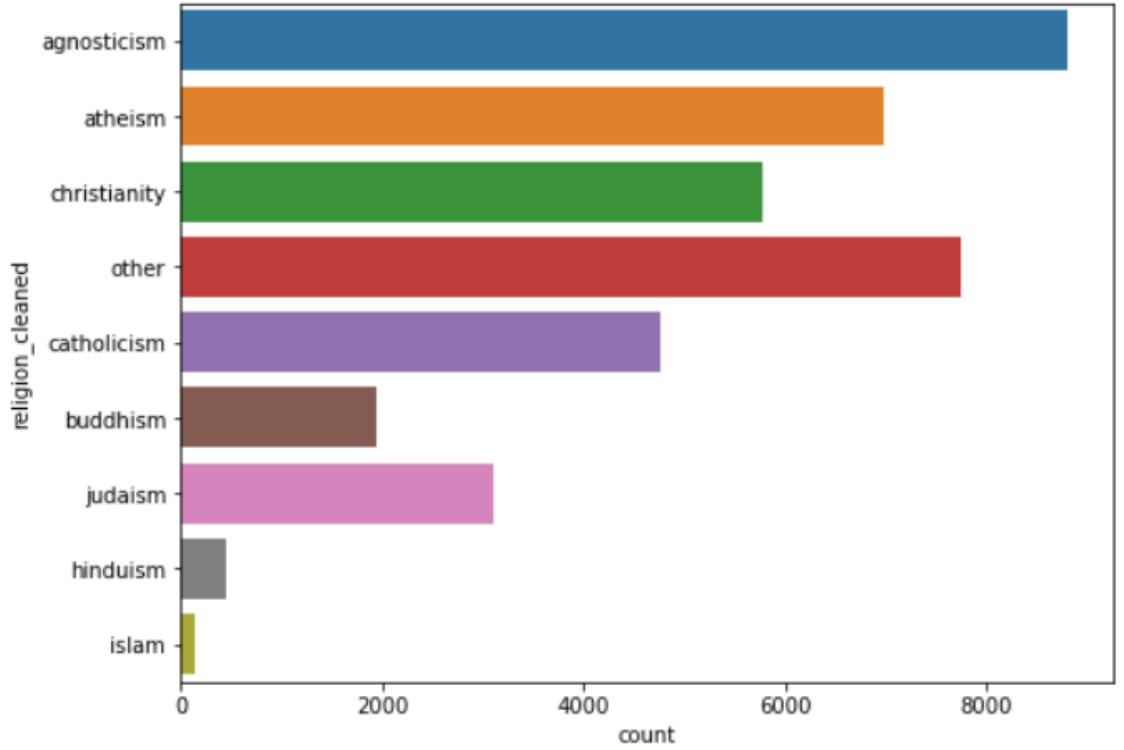


EXPLORATION OF THE DATASET

The plot shows that most of the users are not religious.

The majority of OKCupid's users are agnostic or atheist or did not mention their belief(other).

Between religious users, most of them are Christian.



QUESTIONS TO ANSWER

- Question 1: Can we predict Zodiac signs using 'body_type', 'diet', 'orientation', 'pets', 'religion_cleaned', 'sex', 'job', 'smokes_code', 'drinks_code' and 'drugs_code' features?
- Question 2: Can we use lifestyle information ('diet', 'smokes_code', 'drinks_code', 'drugs_code'), sex and age to predict body type?
- Question 3: Can we predict income of users using "job", "sex" and "education"?
- Question 4: Can we predict the sex of the users based on age and body type?

QUESTIONS TO ANSWER

- Question 5: Can we predict sex of the user with education level and income?
- Question 6: Can we predict education level with essay text word counts?
- Question 7: Can we predict income with length of essays and average word length?
- Question 8: Predict age with the frequency of “I” or “me” in essays?

AUGMENTING THE DATASET

- I created new columns using the mapping:
- smokes_code
- drinks_code
- drugs_code

```
drinks_codes = {
    "not at all": 0,
    "rarely": 1,
    "socially": 2,
    "often": 3,
    "very often": 4,
    "desperately": 5
}

drugs_codes = {
    "never": 0,
    "sometimes": 1,
    "often": 2
}

smokes_codes = {
    "no": 0,
    "when drinking": 1,
    "sometimes": 2,
    "yes": 3,
    "trying to quit": 3
}
```

```
#convert drinks column to numeric value
dataframe["drinks_code"] = dataframe['drinks'].map(drinks_codes)

#convert drugs column to numeric value

dataframe["drugs_code"] = dataframe['drugs'].map(drugs_codes)

#convert smokes ordinal categorical values into numeric value

dataframe["smokes_code"] = dataframe['smokes'].map(smokes_codes)
```

AUGMENTING THE DATASET

```
all_data = df

essay_cols = ["essay0","essay1","essay2","essay3","essay4","essay5","essay6","essay7","essay8","essay9"]

# Removing the NaNs
all_essays = all_data[essay_cols].replace(np.nan, '', regex=True)

# Combining the essays
all_essays = all_essays[essay_cols].apply(lambda x: ' '.join(x), axis=1)

#clean the dirty text from hyperlinks, punctuation and html tags
all_data["all_essays_cleaned_text"] = all_essays.apply(cleanText)

#compute the length of each essay and save them on a new column
all_data["essay_len"] = all_data["all_essays_cleaned_text"].apply(lambda x: len(x))

#count the number of words in each essay
all_data["word_count"] =  all_data["all_essays_cleaned_text"].apply(lambda x: len(x.split()))

#compute the avrage length of each word
#all_data["avg_word_len"] = all_data['essay_len'] / all_data['word_count']

all_data["avg_word_len"] = all_data.apply(lambda row: 0 if row.word_count==0 else (row.essay_len/row.word_count), axis = 1)

#count the number of "i" or "me" occurances in the essay text of each user
all_data["i_or_me_count"] = all_data["all_essays_cleaned_text"].apply(lambda x: x.split().count('i') + x.split().count('me'))
```

all_data["essay_len"] : *The length of each essay and save them on a new column*
all_data["word_count"] : *Number of words in each essay*
all_data["avg_word_len"] : *The avrage length of each word*
all_data["i_or_me_count"] : *The number of "i" or "me" occurances in the essay text of each user*

REGRESSION APPROACHES

- Question 8: Predict age with the frequency of "I" or "me" in essays?

```
mean_squared_error :  87.16646053154385
mean_absolute_error :  7.160277076802092

RMSE of linear regression model is: 9.336298010000744

R2 value of our model is: 0.004232094036801692

The runtime of linear regression model is: 0.00398 seconds
```

```
mean_squared_error :  88.18806428136006
mean_absolute_error :  7.15697021823742

RMSE of knn model is: 9.390850029755564
R2 value of our model is: -0.007438452415069907

The runtime of KNeighbors Regressor model is: 0.46192 secon
```

- By comparing these two regression models, we can see that linear regression is faster than KNN regressor and its R2 value is bigger too. Linear regression model outperforms KNeighborsRegressor a little.

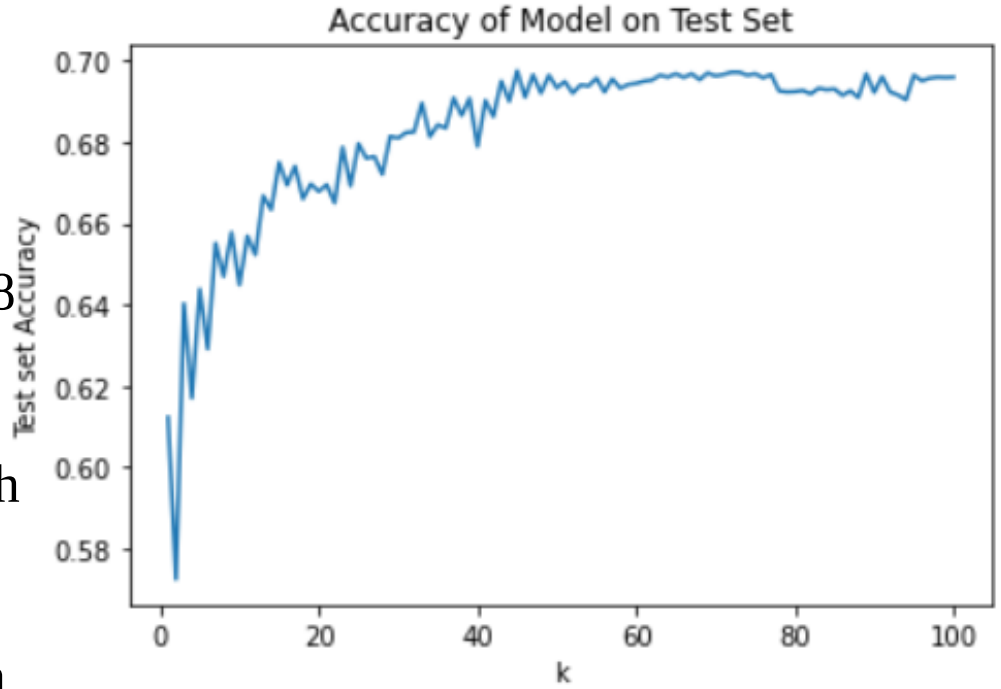
- We cannot predict age with the frequency of "I" or "me" in essays.(Because of low R2 value)

CLASSIFICATION APPROACHES: (K-NEAREST NEIGHBORS)

- **Question 4: Can we predict the sex of the users based on age and body type?**

  - Best accuracy was found when k = 45
  - The runtime of K Nearest Neighbor model is: 24.99098 seconds
  - The accuracy of model on training data is: 85.0% which is very good

An accurate model can be built to predict the gender of an user based on their age and body type



Best K for best accuracy is: 45

CLASSIFICATION APPROACHES: (MULTINOMIAL NAIVE BAYES)

	precision	recall	f1-score	support
f	0.84	0.78	0.81	4259
m	0.87	0.90	0.88	6671
accuracy			0.86	10930
macro avg	0.85	0.84	0.85	10930
weighted avg	0.85	0.86	0.85	10930

The runtime of K Nearest Neighbor model is: 24.99098 seconds

The accuracy of model on training data is: 85.0%

	precision	recall	f1-score	support
f	0.74	0.34	0.47	4259
m	0.69	0.92	0.79	6671
accuracy			0.70	10930
macro avg	0.71	0.63	0.63	10930
weighted avg	0.71	0.70	0.66	10930

The runtime of Multinomial Naive Bayes model is: 0.34425 seconds

The accuracy of model on training data is: 69.0%

The runtime of Multinomial Naive Bayes model is: 0.34425 seconds

The accuracy of model on training data is: 69.0%

KNN algorithm has better accuracy but Multinomial Naive Bayes model is very faster

CONCLUSIONS

- Most of the users are between 25 to 35 years old.There are more male users than female users.
- We cannot predict Zodiac signs using 'body_type', 'diet', 'orientation', 'pets', 'religion_cleaned', 'sex', 'job', 'smokes_code', 'drinks_code' and 'drugs_code' features.
- We cannot predict body type by using lifestyle information ('diet', 'smokes_code', 'drinks_code', 'drugs_code'), sex and age.
- We cannot predict the income of the user by making models based on his job, sex and education.
- An accurate model can be built to predict the gender of an user based on their age and body type. I got 86% accuracy with SVM, Logistic Regression, Decision Tree and KNN models.
- We can build good models with high accuracy(72%) to predict sex of an user with education level and income.
- We cannot use essay text word count to predict education level.
- we cannot predict income with length of essays and average word length.
- We cannot predict age with the frequency of “I” or “me” in essays.

GOOD LUCK AND HAVE FUN!

The End