Machine Learning Fundamentals Capstone Project

Donald Olsen
01.07.2019

Hypothesis

Can an individual’s self described body type, diet, education and job predict their gender?

The Experiment

Materials

Found around the ...!

- OKCupid User Profiles
- Python 3.7
- Scikit-Learn 0.20.2
- Pandas 0.23.4
- Matplotlib 3.0.2
- A lot of patiences

Explore the data:

1. Import modules
2. Load the profiles into a DataFrame
3. Print some exploratory data:
    a. DataFrame Describe method
    b. Column Value Counts

	age	height	income
count	59946.000000	59943.000000	59946.000000
mean	32.340290	68.295281	20033.222534
std	9.452779	3.994803	97346.192104
min	18.000000	1.000000	-1.000000
25%	26.000000	66.000000	-1.000000
50%	30.000000	68.000000	-1.000000
75%	37.000000	71.000000	-1.000000
max	110.000000	95.000000	1000000.000000

```
average           14652
fit               12711
athletic          11819
thin               4711
curvy              3924
a little extra     2629
skinny             1777
full figured       1009
overweight          444
jacked              421
used up             355
rather not say      198
Name: body_type, dtype: int64
```

Explore some more:

1. Create a new dataframe by copying the specific columns of interest.

2. Drop the not a number values, i.e. NaNs.

3. Drop some outlier ages that were questionable, 109 and 110, figure 1.

4. Drop the ‘rather not say’ body type as it is not informative, figure 2.

5. Plot the ‘body_type’ to the ‘diet’.

6. Plot the ‘age’ to the ‘body_type’.

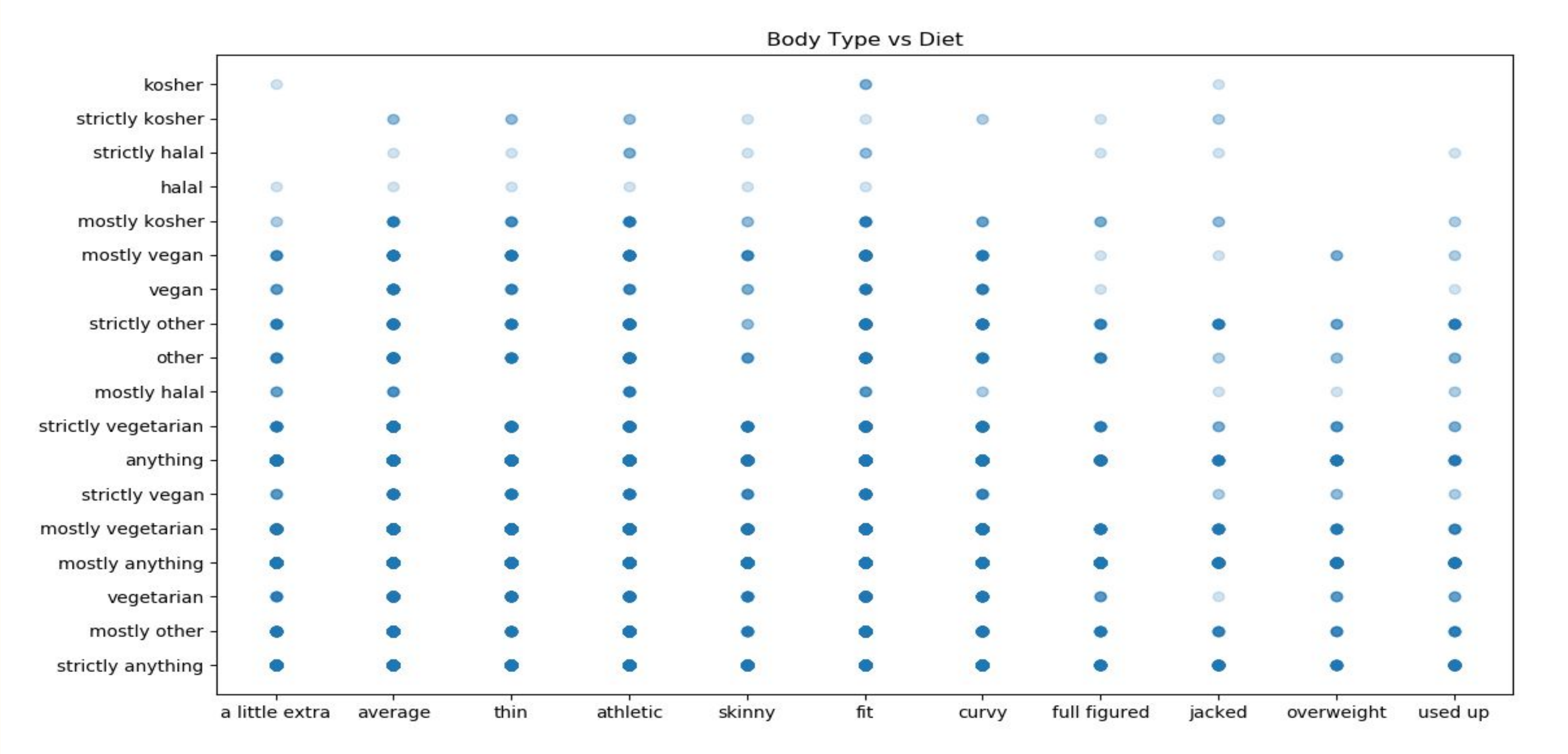
Figure 1:

```
67      66
68      59
69      31
110      1
109      1
Name: age, dtype: int64
```

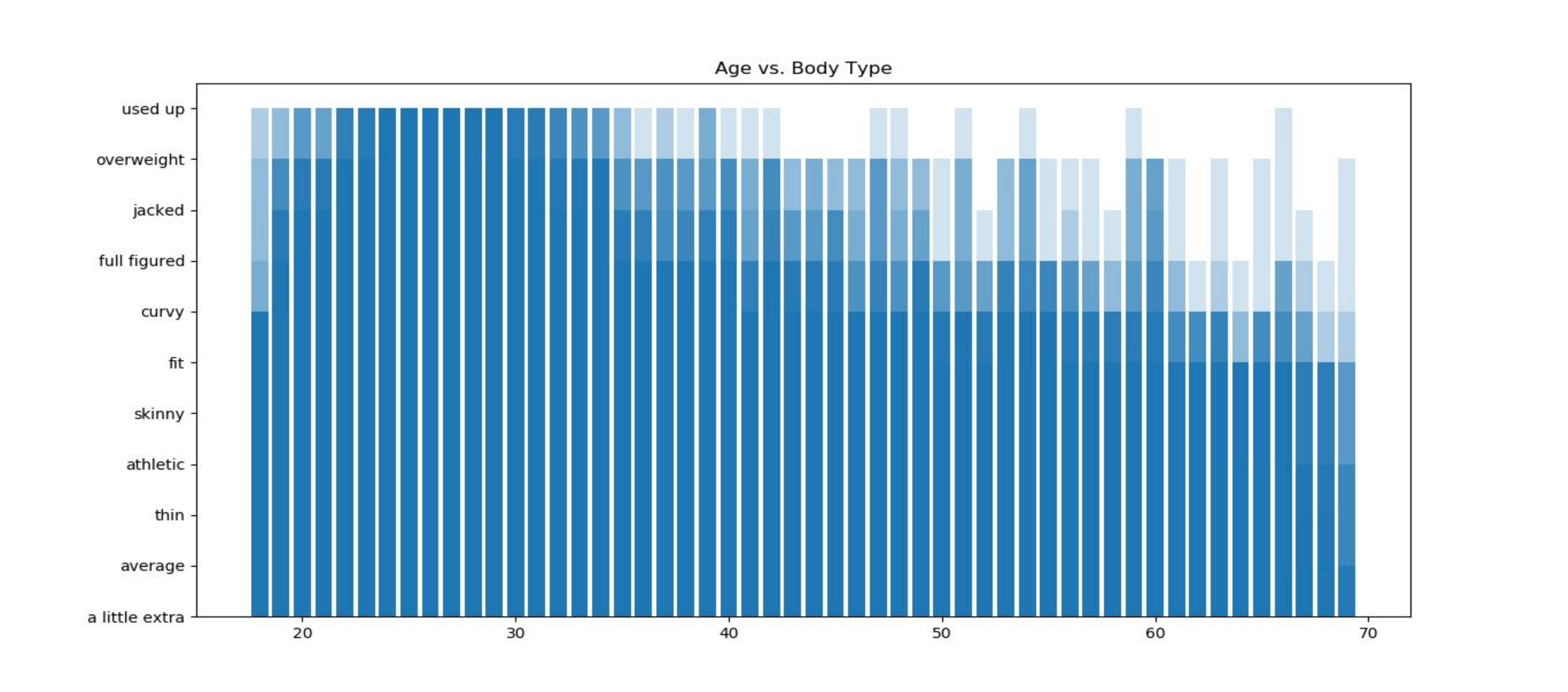
Figure 2:

```
average            14652
fit                12711
athletic           11819
thin                4711
curvy               3924
a little extra      2629
skinny              1777
full figured        1009
overweight           444
jacked               421
used up              355
rather not say       198
Name: body_type, dtype: int64
```

Plot of the ‘body_type’ to the ‘diet’



Plot ‘age’ to ‘body_type’



Data Transformation:

Transform the data using the Pandas get_dummies() method to encode the columns to a one hot encoding, using the following lines of code:

```
rev_cols = ['age', 'body_type', 'diet', 'job', 'education']
profiles_ohe = profiles[rev_cols].copy()
X = pd.get_dummies(profiles_ohe,  columns=rev_cols, prefix=rev_cols)
```

	body_type	body_type_a little extra	body_type_average	body_type_thin
0	a little extra	1	0	0
1	average	0	1	0
3	thin	0	0	1
5	average	0	1	0
7	average	0	1	0

Convert ‘sex’ data to numerical data

Convert the ‘sex’ column data from ‘f’ and ‘m’ to numerical values of 0 and 1, respectively, using the following line of code:

```
profiles['sex_code'] = profiles['sex'].astype("category").cat.codes
```

user	sex	sex_code
0	f	0
1	m	1

Expectations

Tell the audience what you expect to happen... ???

Hypothesis support

I think this is what’s going to happen because…

I am not sure what will happen since I believe most people have a distorted view of their body.

Variables that may affect the outcome…

- Limited data sample.
- Distorted answers due to self-perceptions.
- Deliberate false answers.

The Results

Decision Tree Classifier Results

```
[CART] CV Mean: 0.65504, Std: 0.00699
[CART] Train Score: 0.90449
[CART] Test Score: 0.66026
[CART] Metrics Accuracy: 0.66026
[CART] Metrics Report:
              precision    recall  f1-score   support

      female       0.57      0.61      0.59      2307
        male       0.73      0.69      0.71      3518

   micro avg       0.66      0.66      0.66      5825
   macro avg       0.65      0.65      0.65      5825
weighted avg       0.67      0.66      0.66      5825

CART Model: Elapsed Time (s): 10.721
```

K-Neighbors Classifier Results

```
[KNN] CV Mean: 0.66929, Std: 0.01019
[KNN] Train Score: 0.90655
[KNN] Test Score: 0.67451
[KNN] Metrics Accuracy: 0.67451
[KNN] Metrics Report:
              precision    recall  f1-score   support

      female       0.60      0.54      0.57      2307
        male       0.72      0.76      0.74      3518

   micro avg       0.67      0.67      0.67      5825
   macro avg       0.66      0.65      0.65      5825
weighted avg       0.67      0.67      0.67      5825

KNN Model: Elapsed Time (s): 564.407
```

Logistic Regression Results

```
[LR] CV Mean: 0.71688, Std: 0.00701
[LR] Train Score: 0.72009
[LR] Test Score: 0.72275
[LR] Metrics Accuracy: 0.72275
[LR] Metrics Report:
              precision    recall  f1-score   support

      female       0.64      0.69      0.66      2307
        male       0.79      0.74      0.76      3518

   micro avg       0.72      0.72      0.72      5825
   macro avg       0.71      0.72      0.71      5825
weighted avg       0.73      0.72      0.72      5825

LR Model: Elapsed Time (s): 4.783
```

SGD Classifier Results

```
[SGD] CV Mean: 0.70246, Std: 0.02756
[SGD] Train Score: 0.72576
[SGD] Test Score: 0.71828
[SGD] Metrics Accuracy: 0.71828
[SGD] Metrics Report:
              precision    recall  f1-score   support

      female       0.67      0.56      0.61      2307
        male       0.74      0.82      0.78      3518

   micro avg       0.72      0.72      0.72      5825
   macro avg       0.71      0.69      0.70      5825
weighted avg       0.71      0.72      0.71      5825

SGD Model: Elapsed Time (s): 5.211
```

GaussianNB Results

```
[GNB] CV Mean: 0.51799, Std: 0.04731
[GNB] Train Score: 0.51895
[GNB] Test Score: 0.52841
[GNB] Metrics Accuracy: 0.52841
[GNB] Metrics Report:
              precision    recall  f1-score   support

      female       0.45      0.93      0.61      2307
        male       0.85      0.27      0.41      3518

   micro avg       0.53      0.53      0.53      5825
   macro avg       0.65      0.60      0.51      5825
weighted avg       0.69      0.53      0.49      5825

GNB Model: Elapsed Time (s): 4.353
```

MultinomialNB Results

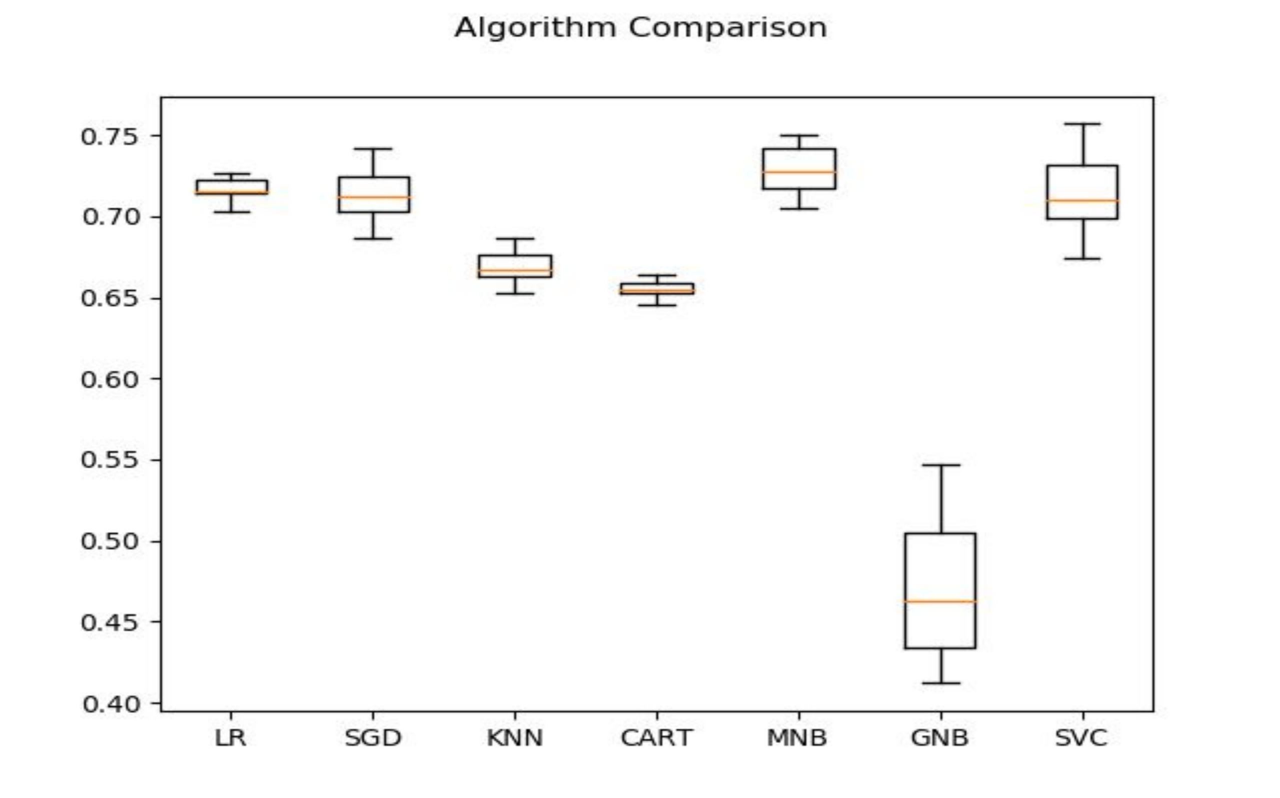
```
[MNB] CV Mean: 0.72873, Std: 0.01519
[MNB] Train Score: 0.73185
[MNB] Test Score: 0.72824
[MNB] Metrics Accuracy: 0.72824
[MNB] Metrics Report:
              precision    recall  f1-score   support

      female       0.73      0.50      0.59      2307
        male       0.73      0.88      0.80      3518

   micro avg       0.73      0.73      0.73      5825
   macro avg       0.73      0.69      0.69      5825
weighted avg       0.73      0.73      0.72      5825

MNB Model: Elapsed Time (s): 1.165
```

Algorithm Comparison Results Plot



Conclusion:

It took about 10 minutes to run the six models with MultinomialNB being the fastest and KNeighborsClassifier being the slowest:

1. MultinomialNB: 1.165 seconds;
2. GaussianNB: 4.353 seconds;
3. LogisticRegression: 4.78 seconds;
4. SGDClassifier: 5.21 seconds;
5. DecisionTreeClassifier: 10.721 seconds;
6. KNeighborsClassifier: 564.407 seconds;

Conclusion, cont'd:

GaussianNB returned the lowest accuracy score for predicting the females (45%) and the highest in predicting the males (85%). MultinomialNB returned the highest overall accuracy score with the same score for both female and male at 73%.

1. GaussianNB: female: 45%, male: 85%, overall: 65%;
2. DecisionTreeClassifier: female: 57%, male: 73%, overall: 65%;
3. KNeighborsClassifier: female: 60%, male: 72%, overall: 66%;
4. LogisticRegression: female: 64%, male: 79%, overall: 71%;
5. SGDClassifier: female: 67%, male: 74%, overall: 72%;
6. MultinomialNB: female: 73%, male: 73%, overall: 73%;

Conclusion, cont’d:

The results of the different models indicate that it is less accurate in the prediction of the females than the males. The reason for this may have to do with males answer specific body type questions compared to women. Specifically, to they have a tendency to answer the body type as athletic or fit, regardless of how accurate it is since they are trying to find a partner. Or, are women more honest about their body type.

The End