Date-a-Scientist Portfolio Project

‘Marcellus Hunt

Agenda

Project Description

Data Analysis

ML Problem/Discovery

ML Model

Results/Conclusion




Project Description

- In this Project, I will analyze data from OKCupid, which is a dating app that uses multiple choice and short answers to match users.

Data Analysis

Subtitle

Loading and Checking Data

I loaded the data and individually examined the following fields that appeared in data.

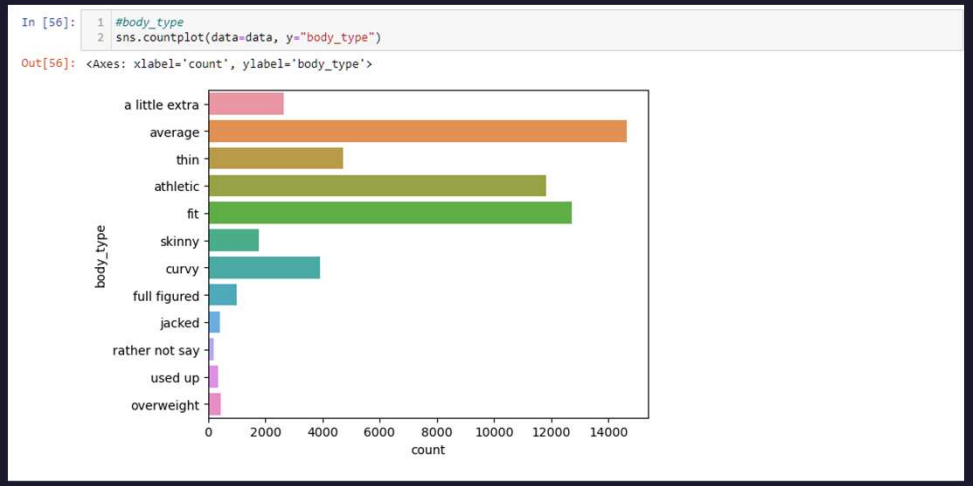
- Age and Height are the only continuous variables
- Essays are open ended answers
- The rest were discrete fields

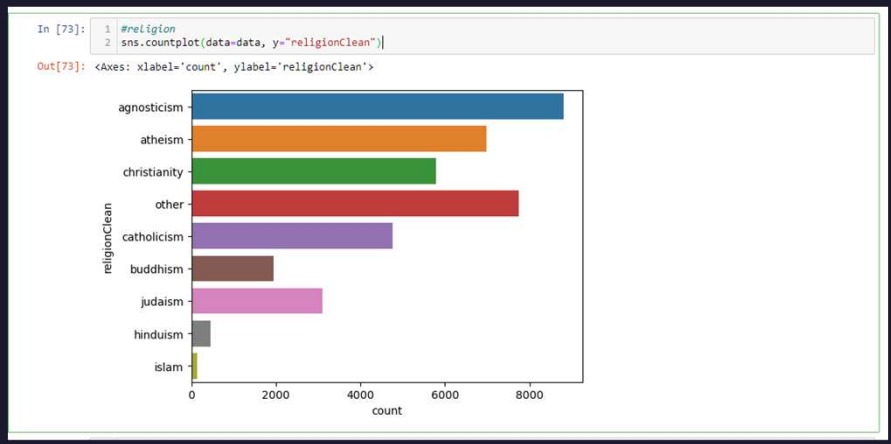
```
In [5]: 1 #EDA
        2 print(data.info())

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 59946 entries, 0 to 59945
Data columns (total 31 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   age          59946 non-null  int64
 1   body_type    54650 non-null  object
 2   diet         35551 non-null  object
 3   drinks       56961 non-null  object
 4   drugs        45866 non-null  object
 5   education    53318 non-null  object
 6   essay0       54458 non-null  object
 7   essay1       52374 non-null  object
 8   essay2       50308 non-null  object
 9   essay3       48470 non-null  object
 10  essay4       49409 non-null  object
 11  essay5       49096 non-null  object
 12  essay6       46175 non-null  object
 13  essay7       47495 non-null  object
 14  essay8       40721 non-null  object
 15  essay9       47343 non-null  object
 16  ethnicity    54266 non-null  object
 17  height       59943 non-null  float64
 18  income       59946 non-null  int64
 19  job          51748 non-null  object
 20  last_online  59946 non-null  object
 21  location     59946 non-null  object
 22  offspring    24385 non-null  object
 23  orientation  59946 non-null  object
 24  pets         40025 non-null  object
 25  religion     39720 non-null  object
 26  sex          59946 non-null  object
 27  sign         48890 non-null  object
 28  smokes       54434 non-null  object
 29  speaks       59896 non-null  object
 30  status       59946 non-null  object
dtypes: float64(1), int64(2), object(28)
memory usage: 14.2+ MB
None
```

Exploring the fields

I performed analysis on the following fields as shown in the example to the right.

I examined these to get a glimpse an insight into the distribution of these fields



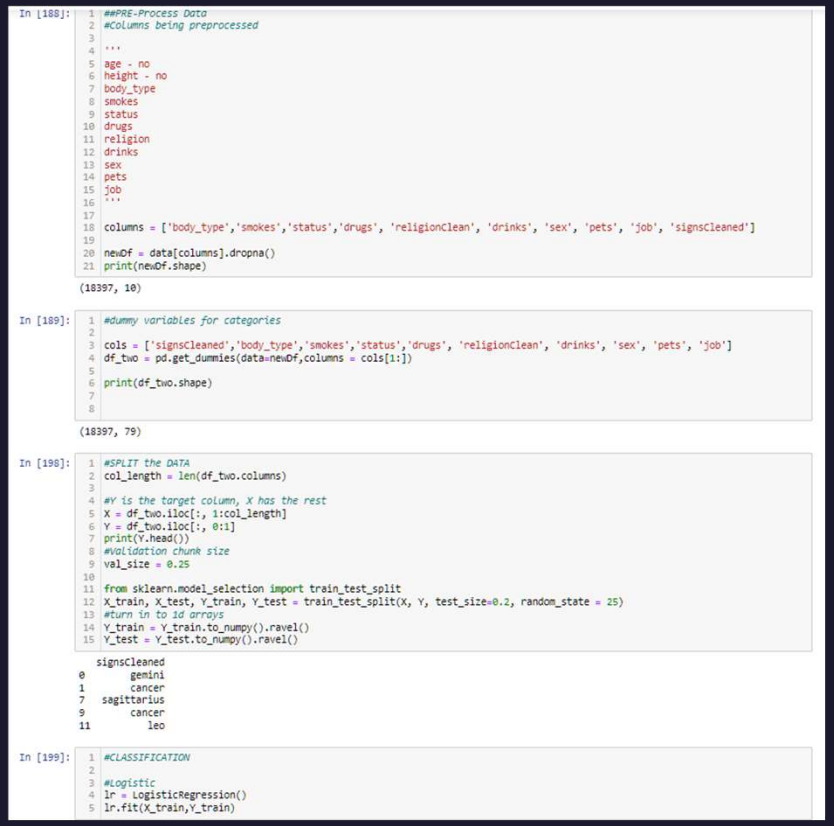


ML Problem/Discovery

- I am now wanting to test if Zodiac signs are even valid.
- Many lean on this for characterizing individuals. I will explore if they can be predicted accurately.
- I will also be looking to prove that The Most Honorable Elijah Muhammad's statement that there is 'no science to behind astrology' is true.

Model

- I will be using three models to find best performance:
  - Logistic Regression
  - K Nearest Neighbor Classifier
  - Decision Trees
- First I used Logistic Regression Model.
- To do this I needed to use One Hot encoding for the categorical fields.
- Age and Height variables aren't useful for this type of classification so those will be excluded from being used in model



```
In [188]:
##PRE-Process Data
#Columns being preprocessed

'''
age - no
height - no
body_type
smokes
status
drugs
religion
drinks
sex
pets
job
'''

columns = ['body_type','smokes','status','drugs', 'religionClean', 'drinks', 'sex', 'pets', 'job', 'signsCleaned']

newDf = data[columns].dropna()
print(newDf.shape)
```

```
(18397, 10)
```

```
In [189]:
#dummy variables for categories

cols = ['signsCleaned','body_type','smokes','status','drugs', 'religionClean', 'drinks', 'sex', 'pets', 'job']
df_two = pd.get_dummies(data=newDf,columns = cols[1:])

print(df_two.shape)
```

```
(18397, 79)
```

```
In [198]:
#SPLIT the DATA
col_length = len(df_two.columns)

#Y is the target column, X has the rest
X = df_two.iloc[:, 1:col_length]
Y = df_two.iloc[:, 0:1]
print(Y.head())
#validation chunk size
val_size = 0.25

from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state = 25)
#turn in to 1d arrays
Y_train = Y_train.to_numpy().ravel()
Y_test = Y_test.to_numpy().ravel()
```

```
   signsCleaned
0        gemini
1        cancer
7   sagittarius
9        cancer
11           leo
```

```
In [199]:
#CLASSIFICATION

#Logistic
lr = LogisticRegression()
lr.fit(X_train,Y_train)
```

Model cont.

- The score for Logistic regression scored at a very low 7.6% on its training set
- KNN Classifer scored a better 33% on its training set.

```
In [44]:
#KNN Classifier
knn_model = KNeighborsClassifier(n_neighbors = 5).fit(X_train, Y_train)
knn_predictions = knn_model.predict(X_train)

In [45]:
print(classification_report(Y_train, knn_predictions))
```

	precision	recall	f1-score	support
aquarius	0.25	0.64	0.36	1128
aries	0.28	0.49	0.35	1185
cancer	0.30	0.42	0.35	1223
capricorn	0.31	0.34	0.32	1109
gemini	0.37	0.34	0.35	1339
leo	0.39	0.31	0.34	1338
libra	0.43	0.28	0.34	1253
pisces	0.40	0.26	0.31	1152
sagittarius	0.40	0.25	0.31	1197
scorpio	0.43	0.25	0.31	1247
taurus	0.40	0.23	0.29	1238
virgo	0.42	0.23	0.30	1308
accuracy			0.33	14717
macro avg	0.36	0.34	0.33	14717
weighted avg	0.37	0.33	0.33	14717

```
#CLASSIFICATION

#Logistic
lr = LogisticRegression()
lr.fit(X_train,Y_train)
```

```
C:\Users\celly\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:458: ConvergenceWarning: lbfgs failed to converge
(status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
LogisticRegression()
```

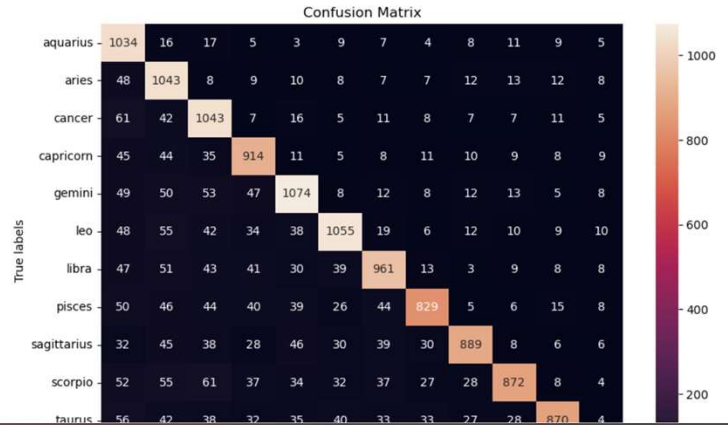
```
# Score the model on the training data
print(lr.score(X_train,Y_train))

# Score the model on the test data
print(lr.score(X_test,Y_test))

# Print the model coefficients
#print(lr.coef_)
```

```
0.11605626146633145
0.0766304347826087
```

Model Cont.

- Decision Tree Classifier improved to a 40% in accuracy!
- But as shown, the Decision tree model overfitted the data as it has a depth of 54 branches.
- After the K cross fold validation and setting max depth to 20, the best accuracy was 8%
- Also the Test set for the K Nearest Neighbors scored an 8% as well.

```
sns.heatmap(cart_cm, annot=True, ax = ax,fmt="d");

# labels, title and ticks
ax.set_xlabel('Predicted labels');
ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
ax.yaxis.set_tick_params(rotation=360)
ax.xaxis.set_tick_params(rotation=90)

ax.xaxis.set_ticklabels(cart_labels);
ax.yaxis.set_ticklabels(cart_labels);
```



Out[49]: 51

In [50]:
```
#To make a point, a five fold cross validation is created with the same data.
#The results are worse than the KNN and about the Logistic Regression algorithms. the baseline was ~9%

from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score

kfold = KFold(n_splits=5, shuffle=True, random_state=0)
results = cross_val_score(cart_model, X_train, Y_train, cv=kfold, scoring='accuracy')

print(results)
print("Baseline: %.2f%% (%.2f%%)" % (results.mean()*100, results.std()*100))
```

```
[0.08559783 0.0923913  0.0795107  0.08460754 0.08392796]
Baseline: 8.52% (0.42%)
```

In [51]:
```
#This situation I'm going to set max depth to prevent overfitting.
cart_model20 = DecisionTreeClassifier(max_depth = 20).fit(X_train, Y_train)
cart_predictions20 = cart_model20.predict(X_train)
```

In [52]:
```
print(classification_report(Y_train, cart_predictions20))
#We now get an accuracy of 39% which is once again low
```

```
              precision    recall  f1-score   support

    aquarius       0.27      0.48      0.35      1128
       aries       0.49      0.44      0.46      1185
      cancer       0.25      0.52      0.34      1223
   capricorn       0.50      0.37      0.43      1109
      gemini       0.43      0.40      0.41      1339
         leo       0.54      0.38      0.45      1338
       libra       0.59      0.32      0.42      1253
      pisces       0.45      0.36      0.40      1152
 sagittarius       0.48      0.37      0.42      1197
     scorpio       0.39      0.37      0.38      1247
      taurus       0.38      0.39      0.39      1238
       virgo       0.40      0.34      0.36      1308
```

Conclusion/Results

In our analysis of trying to predict zodiac signs, it turns out that based on my experiment that zodiac signs can NOT be predicted based off important information.

As the Most Honorable Elijah Muhammad has taught. There's NO science to Zodiac signs!

Thank You

Marcellus Hunt