Reviewable project **Date-A-Scientist**

Machine Learning Fundamentals

Itziar Salaberria

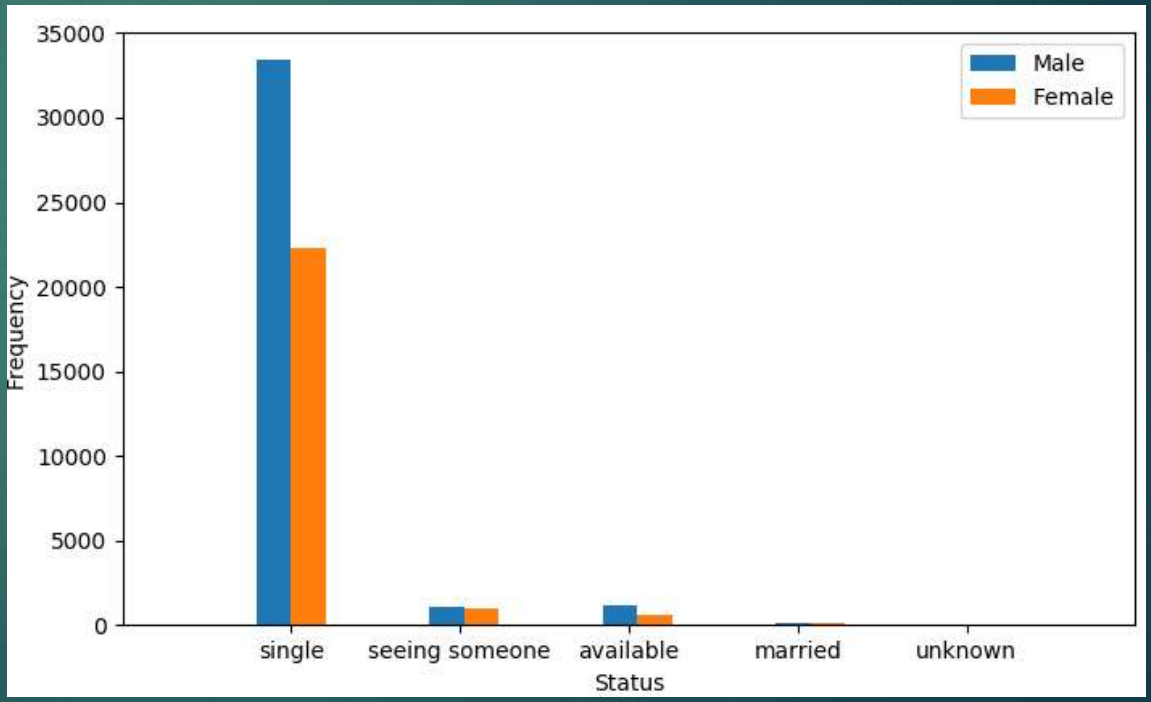
12/11/2018

Table of Contents

Exploration of the Dataset

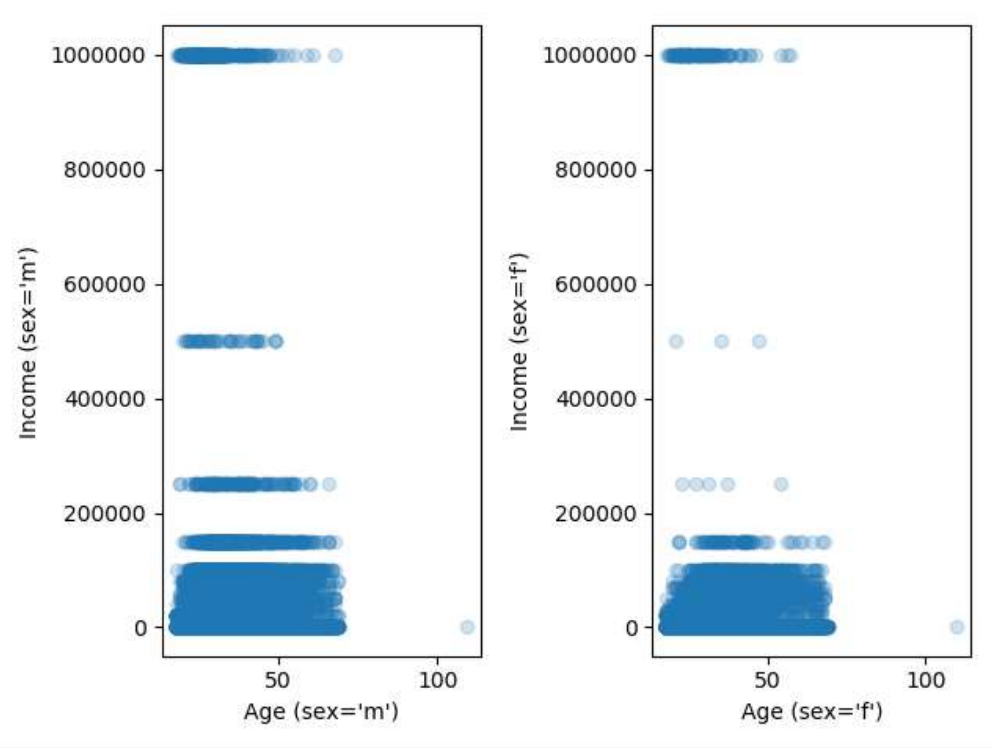
- I have checked status count for “male” and “female” sex values:

```
Male
single               33378
available             1209
seeing someone        1061
married                175
unknown                  6
Name: status, dtype: int64
Female
single               22319
seeing someone        1003
available              656
married                135
unknown                  4
Name: status, dtype: int64
```



Exploration of the Dataset

- I have checked the relation between income and age taking into account sex




```
# Relation income and age
sex_mapping = {"m": 0, "f": 1}

print(df[["age", "income"]])

plt.subplot(1, 2, 1)
plt.scatter(df[df.sex=='m'].age,
df[df.sex=='m'].income, alpha=0.2)

plt.xlabel("Age (sex='m')")
plt.ylabel("Income (sex='m')")

plt.subplot(1, 2, 2)
plt.scatter(df[df.sex=='f'].age, df[df.sex=='f'].income,
alpha=0.2)

plt.xlabel("Age (sex='f')")
plt.ylabel("Income (sex='f')")


plt.show()
```

Question to Answer

- I wonder if it is possible to **predict user's sex** from the information in their profile. I will test it taking into account these features:
  - *age*
  - *income*
  - *height*
  - *essay_word_count (new column)*
  - *status_code (new column)*

Augmenting the Dataset

- I create "essay_word_count" column because I think that maybe women give more details in their descriptions than men. I have calculated like this:

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

# Removing the NaNs
all_essays = df[essay_cols].replace(np.nan, '', regex=True)
# Combining the essays
all_essays = all_essays[essay_cols].apply(lambda x: ' '.join(x), axis=1)


df["essay_word_count"] = all_essays.apply(lambda x: len(x.split()))
```

Augmenting the Dataset

- I also added a new column, which is “status_code” because I think that it can be influence in prediction results:

```
status_mapping = {"single": 0, "seeing someone": 1, "available": 2, "married": 3, "unknown": 4}
df["status_code"] = df.status.map(status_mapping)
```

Augmenting the Dataset

- Features used in the predictions:
  - feature_data = df[['age', 'income', 'essay_word_count', 'status_code','height', 'sex_code']]
  - print(feature_data.head())

	age	income	essay_word_count	status_code	height	sex_code
1	35	80000	278	0	70.0	0
3	23	20000	79	0	71.0	0
11	28	40000	851	1	72.0	0
13	30	30000	0	0	66.0	1
14	29	50000	463	0	62.0	1

Classification Approaches
Support Vector Machines

- Accuracy score: 0.7648848326814428
- Recall score: 0.1211129296235679 3
- Precision score: 0.9487179487179487

- Time to run the model: 2.169583350999999

Classification Approaches

K-Means

- Accuracy score: 0.2894393741851369
- Recall score: 0.9574468085106383
- Precision score: 0.266362807657247

- Time to run the model: 0.07478530299999964

Classification Approaches
Conclusion

- I have applied two classification approaches: *Support Vector Machines* and *K-Means*
- *Support Vector Machines* has obtained much better accuracy score
- Although *K-Means* has obtained better recall score, *Support Vector Machine* has demonstrated more precision, indicating that the percentage of items of the classifier found were actually relevant.
- So, in order to get a better classifier for the data of the question that has been selected in this capstone, *Support Vector Machine* has offered better results.

Regression Approaches

Multiple Linear Regression

- Train score: 0.3406492540297771
- Test score: 0.340498529128181

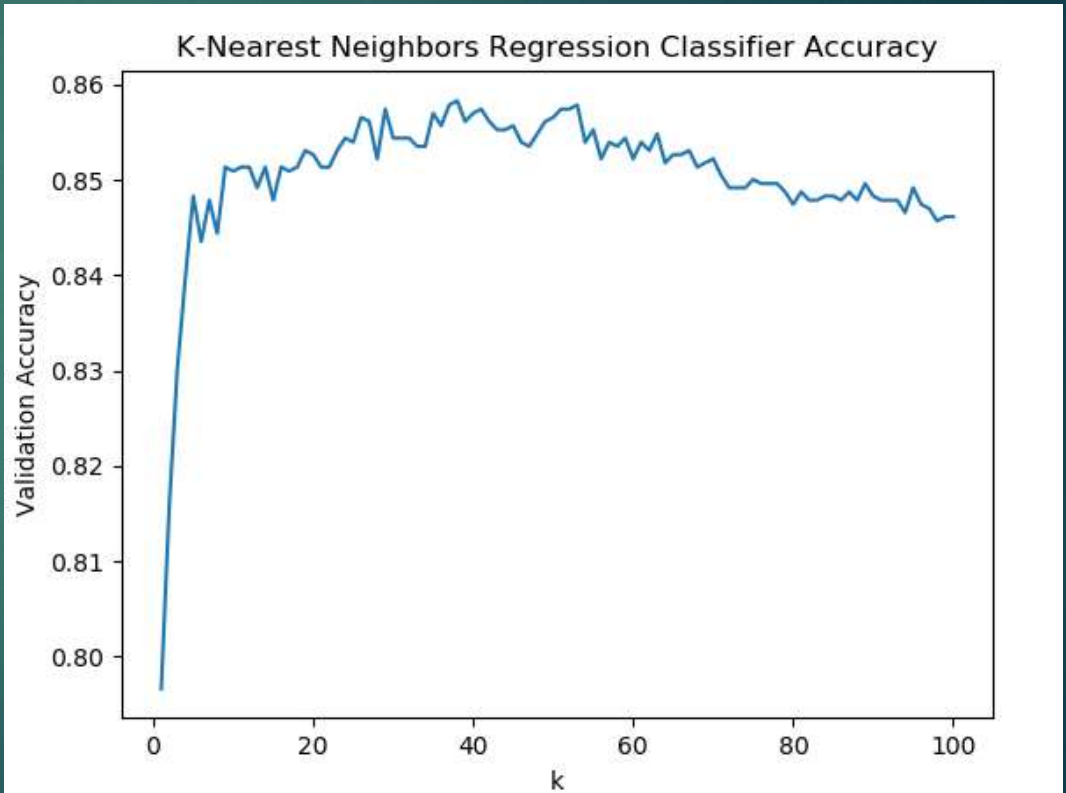
- Accuracy score: 0.8509343763581052
- Recall score: 0.8509343763581052
- Precision score: 0.8509343763581052

- Time to run the model: 0.00454755899999701

Regression Approaches

K-Nearest Neighbors Regression

- One graph have been produce to show the classification accuracy versus K
- I have been able to find best K, which have been 38



Regression Approaches

K-Nearest Neighbors Regression

- Best K: 38
- Score: 0.85832246849196
- Accuracy score: 0.85832246849196
- Recall score: 0.6202945990180033
- Precision score: 0.801268498942917 5

- Time to run the model: 0.1563300560000016
- Time to run the model and selecting best K: 17.820373577

Regression Approaches
Conclusion

- I have applied two regression approaches: *Multiple Linear Regression* and *K-Nearest Neighbors Regression*.
- I have obtained 85% score in applying both techniques
- *Multiple Linear Regression* has obtained better recall score. So, it has had more success finding relevant items
- Although it has been obtained good precision score in both techniques, *Multiple Linear Regression* has got the best score.
- Moreover, *K-Nearest* needs to check K values in order to select "the best" one, which need more runtime in its performance.

Conclusions/Next steps

- Taking into account the obtained results, I can conclude that it is possible to predict user's sex from the information of their profile in a sufficiently precise way
- Next steps could be to include more features in the dataset and explore more results.
- It would be interesenting to add features abaut personality (sociable, sympathetic, introverted, egoistical, etc.) in the dataset