What is I learned about machine learning?_

Machine Learning Fundamentals

Damian Łukasik

04.11.2018

codecademy

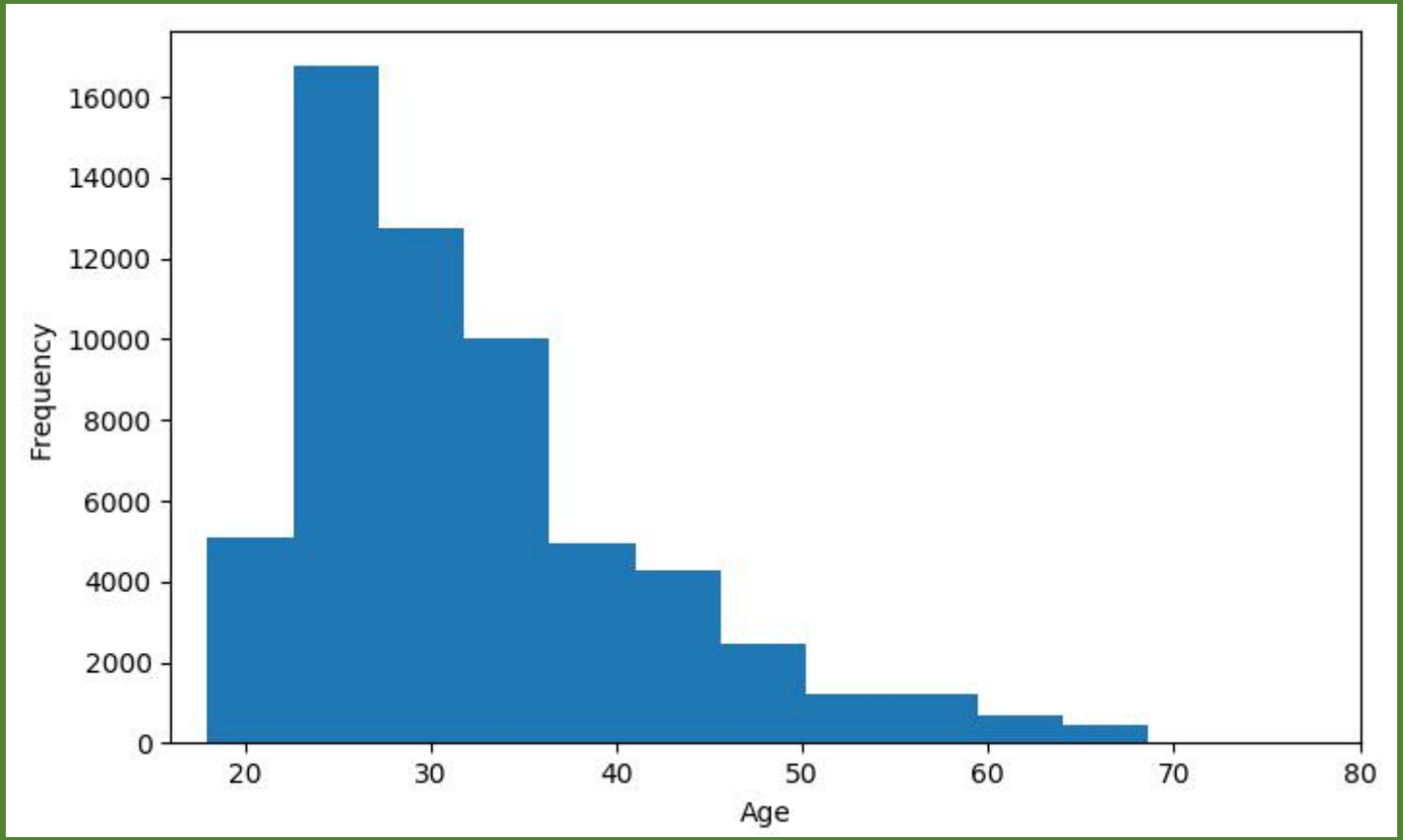
What contain the presentation?

Exploration of the Dataset

- **Graphs containing exploration of the dataset**
- **A statements of my questions and How I arrived there?**
- **New columns and how I did it**
- **The comparison between two classification approaches**
- **The comparison between two regression approaches**
- **An overall conclusion, with a preliminary answer to my initial questions**

codecademy

Graphs containing exploration of the dataset

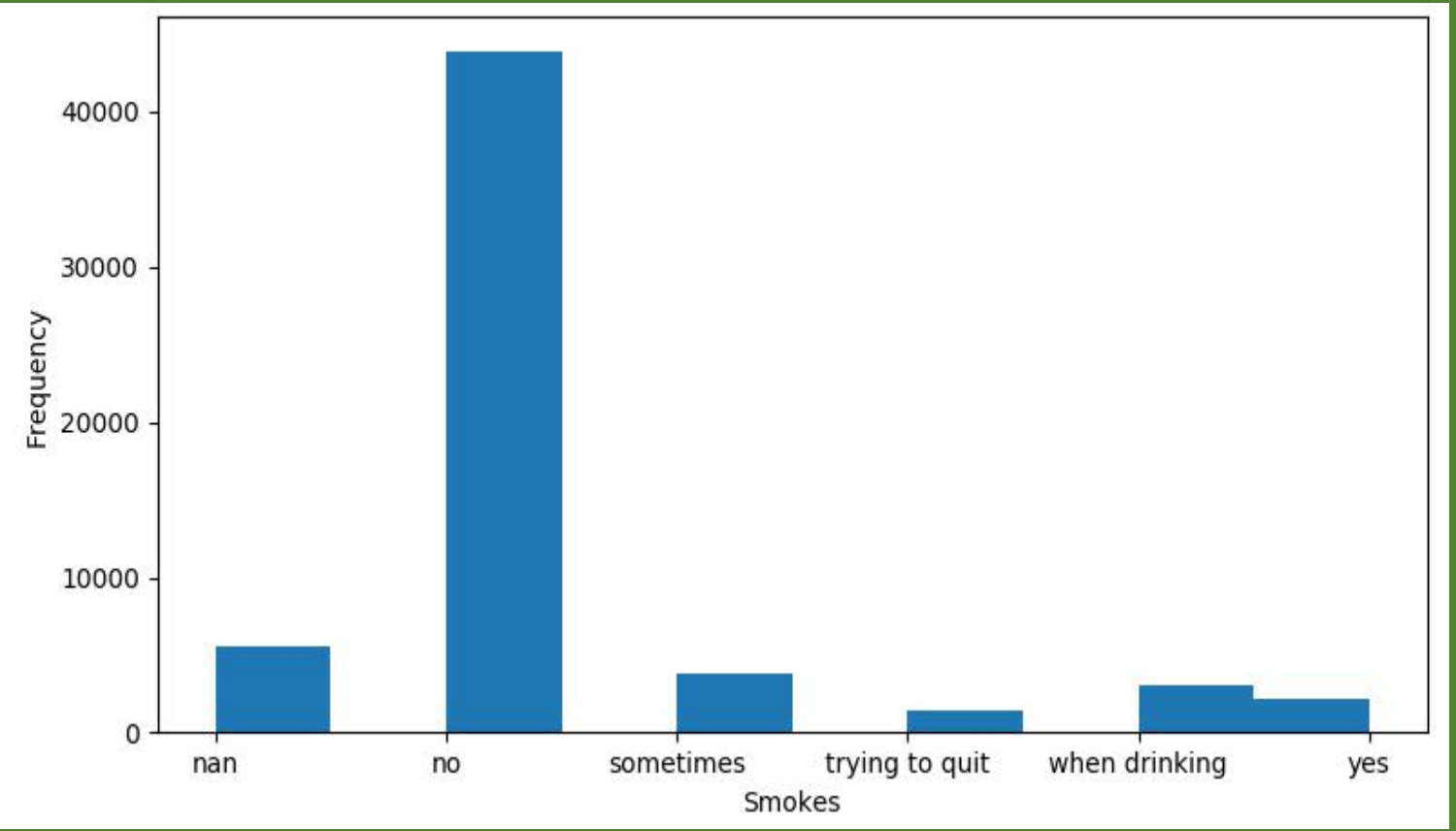


This graph present:

- Frequency of ages in dataset

codecademy

Graphs containing exploration of the dataset

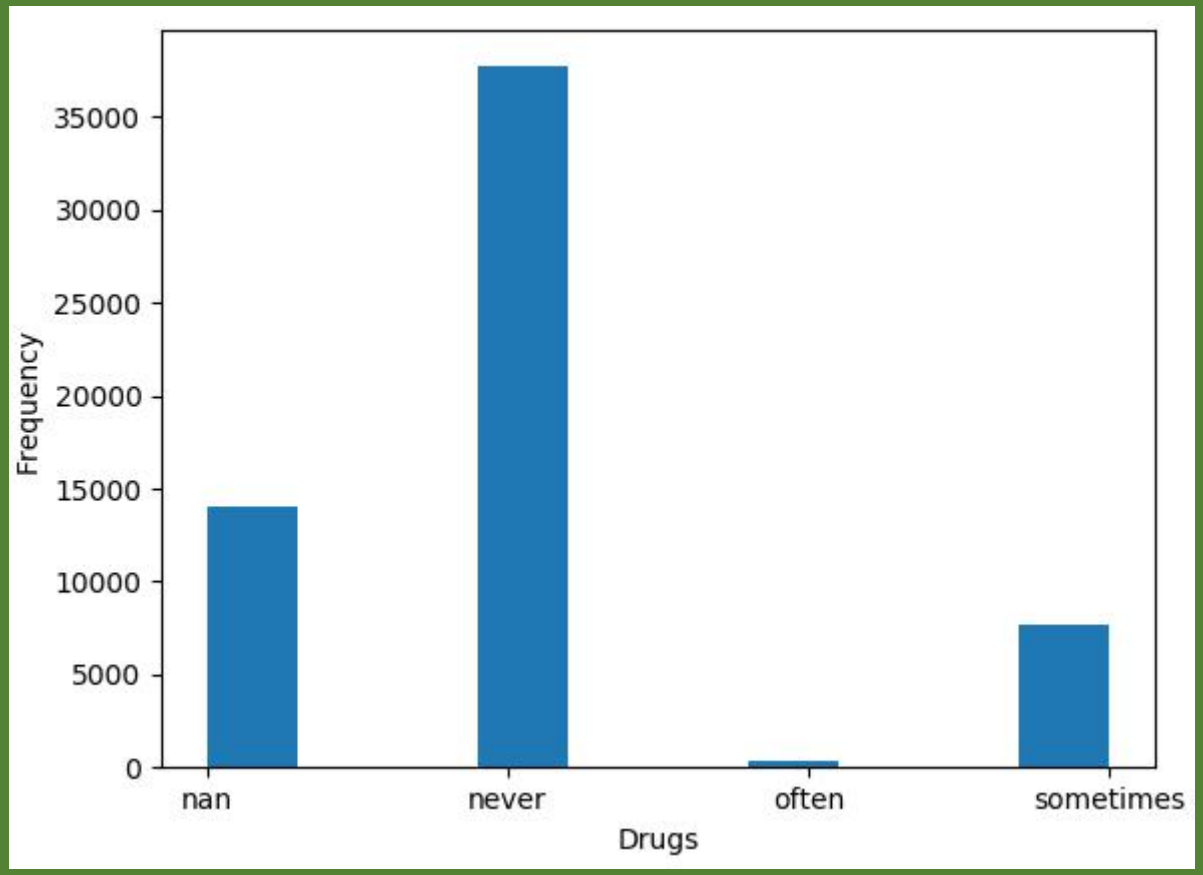


This graph present:

- Frequency of smokes in dataset

codecademy

Graphs containing exploration of the dataset

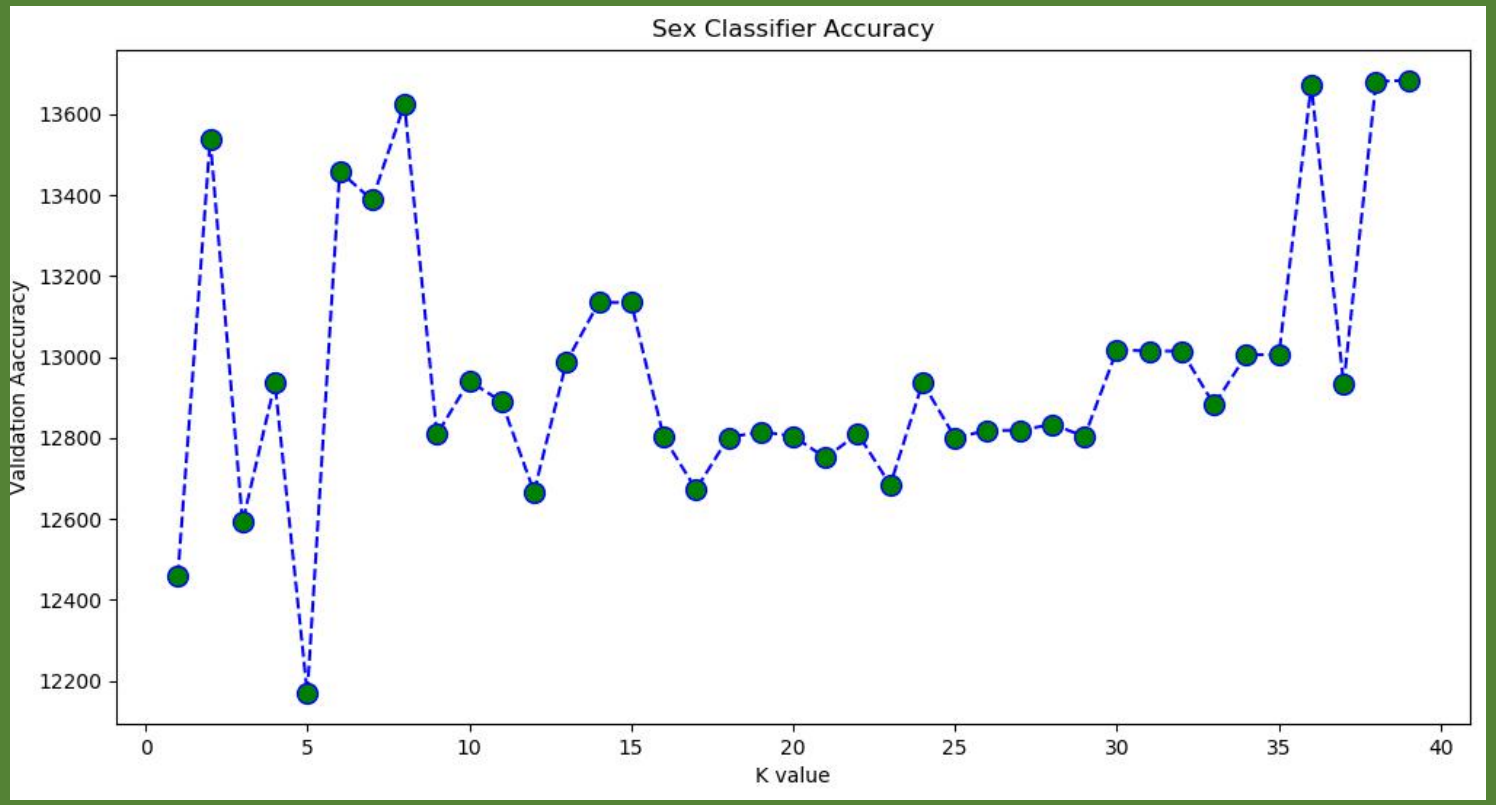


This graph present:

- Frequency of drugs in dataset

codecademy

Graphs containing exploration of the dataset



This graph present:

- Predict Sex by base education level and income

- X-axis is number k

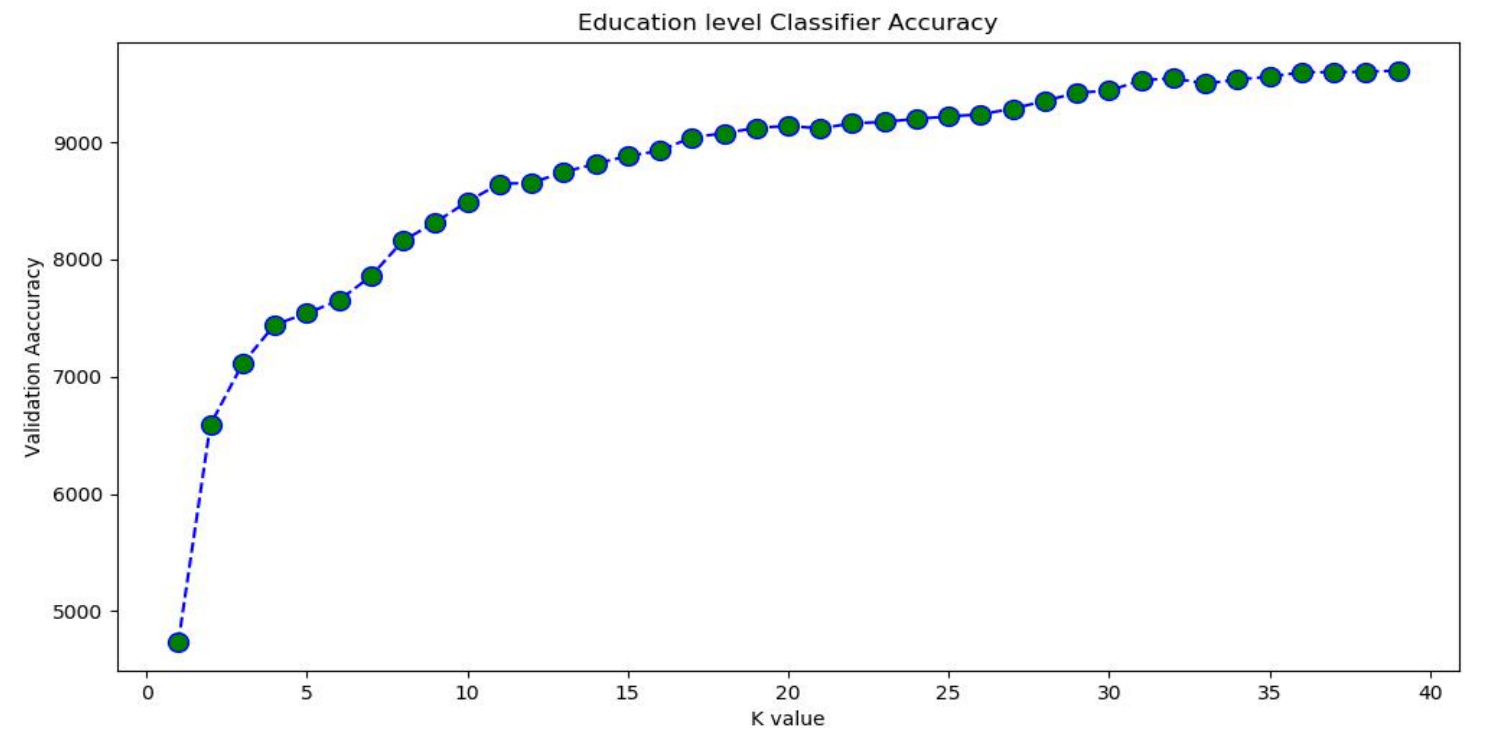
- I used method K-Nearest Neighbors

- I solved Accuracy (number of correct rows) by

$$\frac{\text{number good predict}}{\text{all rows}}$$

codecademy

Graphs containing exploration of the dataset



This graph present:

- Predict Education level by base essay text word counts

- X-axis is number k

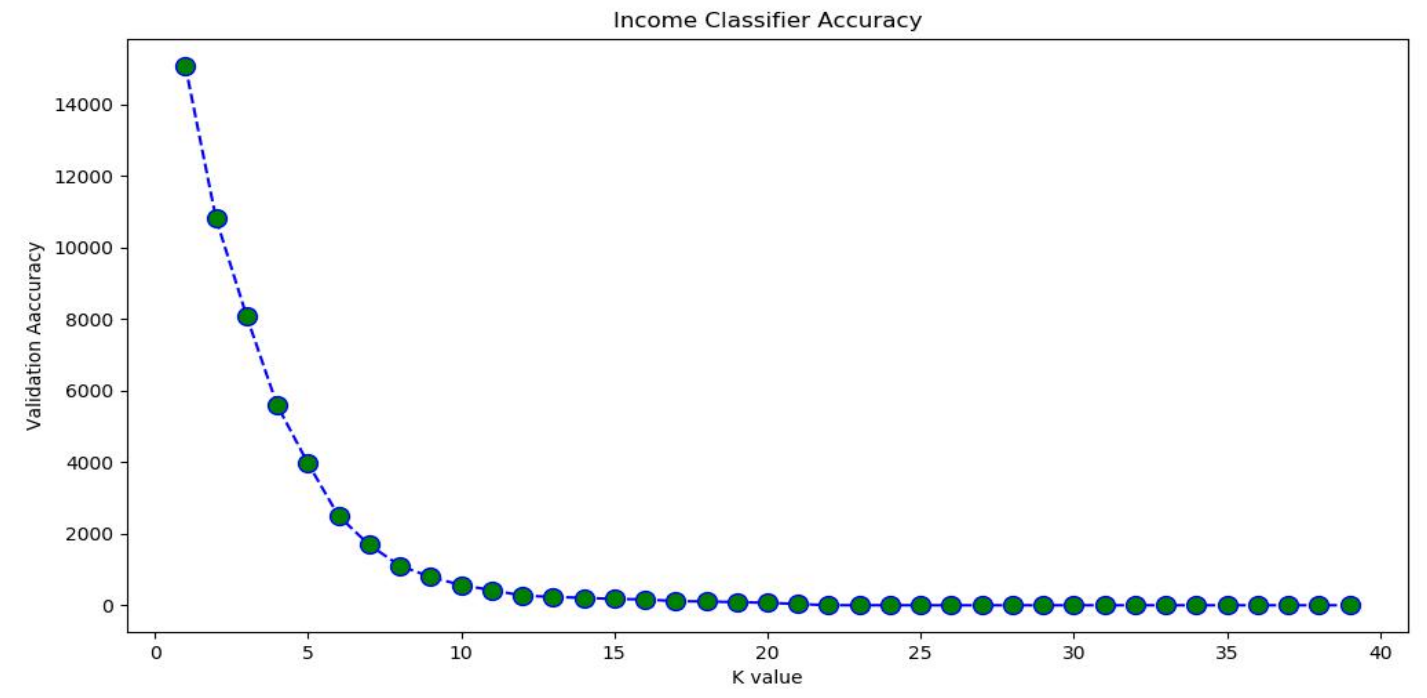
- I used method K-Nearest Neighbors

- I solved Accuracy (number of correct rows) by

$$\frac{\text{number good predict}}{\text{all rows}}$$

codecademy

Graphs containing exploration of the dataset



This graph present:

- Predict Income by base length of essays and average word length

- X-axis is number k

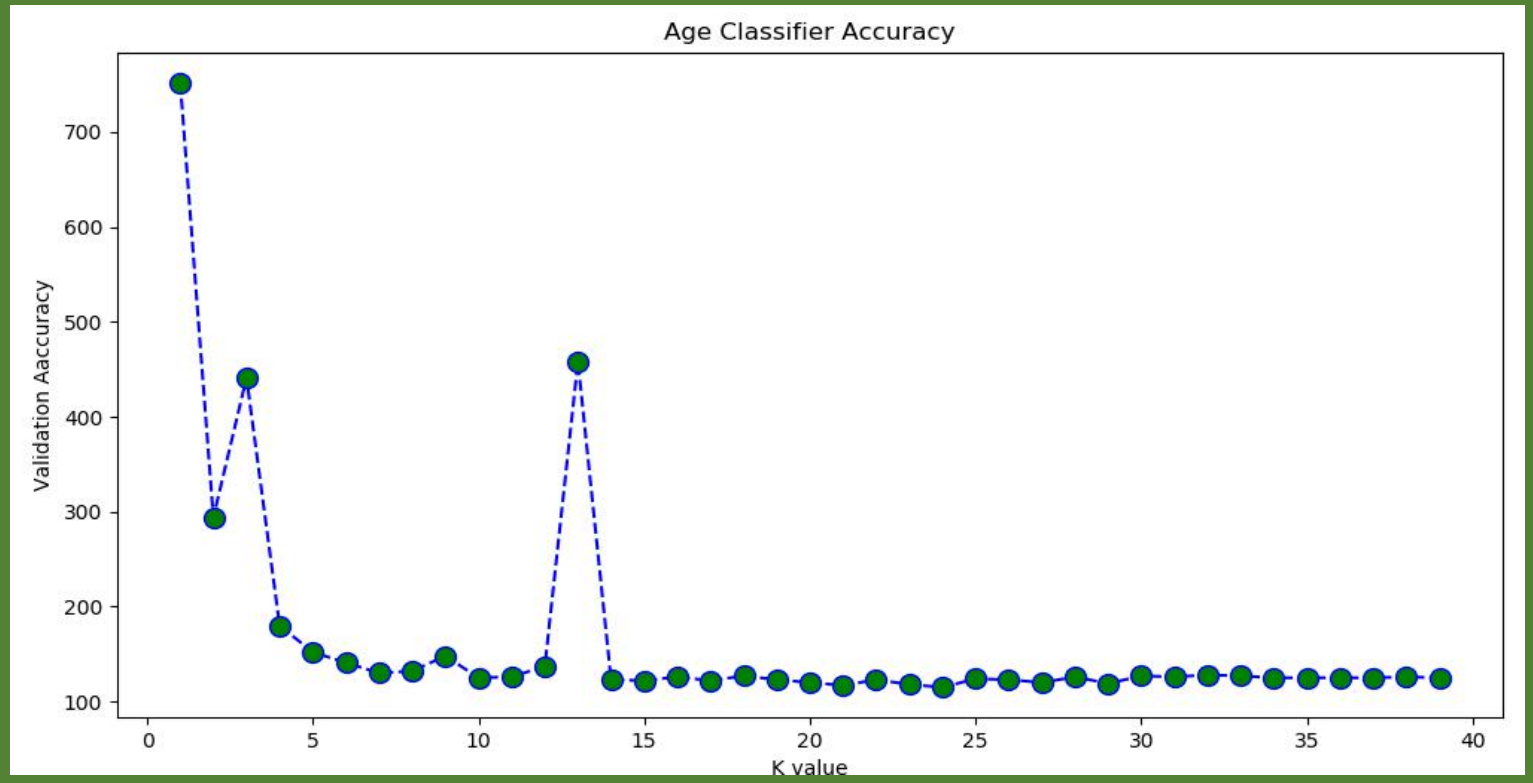
- I used method K-Nearest Neighbors Regression

- I solved Accuracy (number of correct rows) by

$$\frac{\text{number good predict}}{\text{all rows}}$$

codecademy

Graphs containing exploration of the dataset



This graph present:

- Predict Age by base the frequency of "I" or "me" in essays

- X-axis is number k

- I used method K-Nearest Neighbors Regression

- I solved Accuracy (number of correct rows) by

$$\frac{\text{number good predict}}{\text{all rows}}$$

codecademy

A statements of my questions and How I arrived there?

- I created dataset to my methods

```
#function to divide data
def ShareData(arr,size,normalization=True):
  size_training = round(len(arr)*size)
  data_training = []
  data_test = []
  for i in range(len(arr)):
    if i<size_training:
      data_training.append(arr[i])
    else:
      data_test.append(arr[i])
  if normalization==True:
    min_max_scaler = preprocessing.MinMaxScaler()
    if(isinstance(data_training[0], list)==True):
      data_training = min_max_scaler.fit_transform(data_training)
    if(isinstance(data_test[0], list)==True):
      data_test = min_max_scaler.fit_transform(data_test)
  return data_training, data_test
```

I created dataset which containts:

- training_labels (60%)
- test_labels (40%)

- training_points (60%)
- test_points (40%)

codecademy

A statements of my questions and How I arrived there?

K-Nearest Neighbors

#1. Can we predict sex with education level and income?

```
for k in range(1, 40):
  start_time = time.time()
  classifier = KNeighborsClassifier(n_neighbors = k)
  classifier.fit(training_points_first_question,training_labels_first_question)
  guesses_first_question = classifier.predict(test_points_first_question)
  stop_time = (time.time() - start_time)
  knn_first_question.append([guesses_first_question,classifier,[1,"K-Nearest
Neighbors"],test_labels_first_question,stop_time])
results.append(knn_first_question)
```

codecademy

A statements of my questions and How I arrived there?

K-Nearest Neighbors

#2. Can we predict education level with essay text word counts?

```
for k in range(1, 40):
  start_time = time.time()
  classifier = KNeighborsClassifier(n_neighbors = k)
  classifier.fit(training_points_second_question,training_labels_second_question)
  guesses_second_question = classifier.predict(test_points_second_question)
  stop_time = (time.time() - start_time)
  knn_second_question.append([guesses_second_question,classifier,[2,"K-Nearest
Neighbors"],test_labels_second_question,stop_time])
results.append(knn_second_question)
```

codecademy

A statements of my questions and How I arrived there?

Support Vector Machines

#1. Can we predict sex with education level and income?

```
start_time = time.time()
classifier = SVC(kernel = 'linear')
classifier.fit(training_points_first_question,training_labels_first_question)
guesses_first_question = classifier.predict(test_points_first_question)
stop_time = (time.time() - start_time)
results.append([guesses_first_question,classifier,[1,"Support Vector
Machines"],test_labels_first_question,stop_time])
```

codecademy

A statements of my questions and How I arrived there?

Support Vector Machines

#2. Can we predict education level with essay text word counts?

```
start_time = time.time()
classifier = SVC(kernel = 'linear')
classifier.fit(training_points_second_question,training_labels_second_question)
guesses_second_question = classifier.predict(test_points_second_question)
stop_time = (time.time() - start_time)
results.append([guesses_second_question,classifier,[2,"Support Vector
Machines"],test_labels_second_question,stop_time])
```

codecademy

A statements of my questions and How I arrived there?

Naive Bayes

#1. Can we predict sex with education level and income?

```
start_time = time.time()
classifier = MultinomialNB()
classifier.fit(training_points_first_question,training_labels_first_question)
guesses_first_question = classifier.predict(test_points_first_question)
stop_time = (time.time() - start_time)
results.append([guesses_first_question,classifier,[1,"Naive
Bayes"],test_labels_first_question,stop_time])
```

codecademy

A statements of my questions and How I arrived there?

Naive Bayes

#2. Can we predict education level with essay text word counts?

```
start_time = time.time()
classifier = MultinomialNB()
classifier.fit(training_points_second_question,training_labels_second_question)
guesses_second_question = classifier.predict(test_points_second_question)
stop_time = (time.time() - start_time)
results.append([guesses_second_question,classifier,[2,"Naive
Bayes"],test_labels_second_question,stop_time])
```

codecademy

A statements of my questions and How I arrived there?

K-Nearest Neighbors Regression with parameter weights = distance

 # 3. Predict income with length of essays and average word length?

```
for k in range(1, 40):
  start_time = time.time()
  regressor = KNeighborsRegressor(n_neighbors = k, weights = "distance")
  regressor.fit(training_points_third_question,training_labels_third_question)
  guesses_third_question = regressor.predict(test_points_third_question)
  stop_time = (time.time() - start_time)
  knnr_first_question.append([guesses_third_question,classifier,[3,"K-Nearest
Neighbors Regression"],test_labels_third_question,stop_time])
results.append(knnr_first_question)
```

codecademy

A statements of my questions and How I arrived there?

K-Nearest Neighbors Regression with parameter weights = distance

 # 4. Predict age with the frequency of "I" or "me" in essays?

```
for k in range(1, 40):
  start_time = time.time()
  regressor = KNeighborsRegressor(n_neighbors = k, weights = "distance")
  regressor.fit(training_points_four_question,training_labels_four_question)
  guesses_four_question = regressor.predict(test_points_four_question)
  stop_time = (time.time() - start_time)
  knnr_second_question.append([guesses_four_question,classifier,[4,"K-Nearest
Neighbors Regression"],test_labels_four_question,stop_time])
results.append(knnr_second_question)
```

codecademy

A statements of my questions and How I arrived there?

K-Nearest Neighbors Regression with parameter weights = uniform

 # 3. Predict income with length of essays and average word length?

```
for k in range(1, 40):
  start_time = time.time()
  regressor = KNeighborsRegressor(n_neighbors = k, weights = "uniform")
  regressor.fit(training_points_third_question,training_labels_third_question)
  guesses_third_question = regressor.predict(test_points_third_question)
  stop_time = (time.time() - start_time)
  knnr_first_question.append([guesses_third_question,classifier,[3,"K-Nearest
Neighbors Regression"],test_labels_third_question,stop_time])
results.append(knnr_first_question)
```

codecademy

A statements of my questions and How I arrived there?

K-Nearest Neighbors Regression with parameter weights = uniform

 # 4. Predict age with the frequency of "I" or "me" in essays?

```
for k in range(1, 40):
  start_time = time.time()
  regressor = KNeighborsRegressor(n_neighbors = k, weights = "uniform")
  regressor.fit(training_points_four_question,training_labels_four_question)
  guesses_four_question = regressor.predict(test_points_four_question)
  stop_time = (time.time() - start_time)
  knnr_second_question.append([guesses_four_question,classifier,[4,"K-Nearest
Neighbors Regression"],test_labels_four_question,stop_time])
results.append(knnr_second_question)
```

codecademy

A statements of my questions and How I arrived there?

- I created models which had same dataset in input, but I got different answers, different the time to run, accuracy, precision and recall

5. We also learned about K-Nearest Neighbors Regression. Which form of regression works better to answer your question?

- K-Nearest Neighbors Regression with parameter weights = distance works better, because model was more accuracy, but slower

codecademy

New columns and how I did it

Column with the frequency of the words "I" or "me" appearing in the essays.

```
def frequency_words(words,find_words):
  words = words.split()
  count_words = 0
  for word in words:
    for find in find_words:
      if find==word:
        count_words+=1
  if len(words)==0:
    return 0
  return (count_words/len(words))


all_data["frequency_words"] = all_essays.apply(lambda x: frequency_words(x,['I','me']))
print(all_data["frequency_words"])
```

This column contains frequency of the words "I" or "me" appearing in the essays. My function counts number of occurrence of a words and divide this number by number all words

codecademy

New columns and how I did it

Column with average word length

```
def average_word_length(words):
  words = words.split()
  div = len(words)
  if(div==0):
    return 0
  return sum(len(word) for word in words) / div


all_data["avg_word_length"] = all_essays.apply(lambda x: average_word_length(x) )
print(all_data["avg_word_length"])
```

This column contains average words length in row. I created function which sum all words lengths and divide by words number.

codecademy

The comparison between two classification approaches

K-Nearest Neighbors

simplicity

```
from sklearn.neighbors import KNeighborsClassifier

for k in range(1, 40):
  start_time = time.time()
  classifier = KNeighborsClassifier(n_neighbors = k)
  classifier.fit(training_points_first_question,
   training_labels_first_question)
  guesses_first_question =
   classifier.predict(test_points_first_question)
  stop_time = (time.time() - start_time)
  knn_first_question.append([guesses_first_question,classifier,
   [1,"K-Nearest Neighbors"],test_labels_first_question,stop_time])
results.append(knn_first_question)
```

Support Vector Machines

simplicity

```
from sklearn.svm import SVC

start_time = time.time()
classifier = SVC(kernel = 'linear')
classifier.fit(training_points_first_question,
 training_labels_first_question)
guesses_first_question =
classifier.predict(test_points_first_question)
stop_time = (time.time() - start_time)
results.append([guesses_first_question,classifier,
 [1,"Support Vector Machines"],test_labels_first_question,stop_time])
```

codecademy

The comparison between two classification approaches

K-Nearest Neighbors

time to run the model
accuracy, precision, and/or recall

```
K-Nearest Neighbors - Question 1
Accuracy: 0.5613061973475686
[[10319  3558]
 [ 6961  3140]]
             precision    recall  f1-score   support

          0       0.60      0.74      0.66     13877
          1       0.47      0.31      0.37     10101

avg / total       0.54      0.56      0.54     23978

time to run the model: 2.298283100128174
```

Support Vector Machines

time to run the model
accuracy, precision, and/or recall

```
Support Vector Machines - Question 1
Accuracy: 0.5787388439402786
[[13877     0]
 [10101     0]]
             precision    recall  f1-score   support

          0       0.58      1.00      0.73     13877
          1       0.00      0.00      0.00     10101

avg / total       0.33      0.58      0.42     23978

time to run the model: 16.791550159454346
```

codecademy

The comparison between two classification approaches

K-Nearest Neighbors	Support Vector Machines
- I musted to creat 40 times model, where each model had different parameter n_neighbors - both models had this same daatset in input - time to run the model is faster than second model	- I musted to creat only one model - Dataset in input includes 60% original dataset, 40% rest of the dataset was designated to test models - accuracy the model is better than second model

codecademy

The comparison between two regression approaches

K-Nearest Neighbors Regression

with parameter weights = distance

simplicity

```
from sklearn.neighbors import KNeighborsRegressor

for k in range(1, 40):
  start_time = time.time()
  regressor = KNeighborsRegressor(n_neighbors = k, weights =
"distance")

regressor.fit(training_points_third_question,training_labels_third_
question)
  guesses_third_question =
regressor.predict(test_points_third_question)
  stop_time = (time.time() - start_time)
  knnr_first_question.append([guesses_third_question,classifier,
[3,"K-Nearest Neighbors
Regression"],test_labels_third_question,stop_time])
results.append(knnr_first_question)
```

with parameter weights = uniform

simplicity

```
from sklearn.neighbors import KNeighborsRegressor

for k in range(1, 40):
  start_time = time.time()
  regressor = KNeighborsRegressor(n_neighbors = k, weights = "uniform")

regressor.fit(training_points_third_question,training_labels_third_que
stion)
  guesses_third_question =
regressor.predict(test_points_third_question)
  stop_time = (time.time() - start_time)
  knnr_first_question.append([guesses_third_question,classifier,
[3,"K-Nearest Neighbors
Regression"],test_labels_third_question,stop_time])
results.append(knnr_first_question)
```

codecademy

The comparison between two regression approaches

K-Nearest Neighbors Regression

with parameter weights = distance

time to run the model
accuracy, precision, and/or recall

```
K-Nearest Neighbors Regression with parameter weights = distance - Question 3
Accuracy: 267.06666666666666
time to run the model: 0.288820743560791

K-Nearest Neighbors Regression with parameter weights = distance - Question 4
Accuracy: 32.93846153846154
time to run the model: 1.8118846416473389
```

- This model worked slower
- This model is more accuracy

with parameter weights = uniform

time to run the model
accuracy, precision, and/or recall

```
K-Nearest Neighbors Regression with parameter weights = uniform - Question 3
Accuracy: 267.12307692307695
time to run the model: 0.21185564994812012

K-Nearest Neighbors Regression with parameter weights = uniform - Question 4
Accuracy: 17.682051282051283
time to run the model: 1.7771146297454834
```

- This model worked faster
- This model is less accuracy

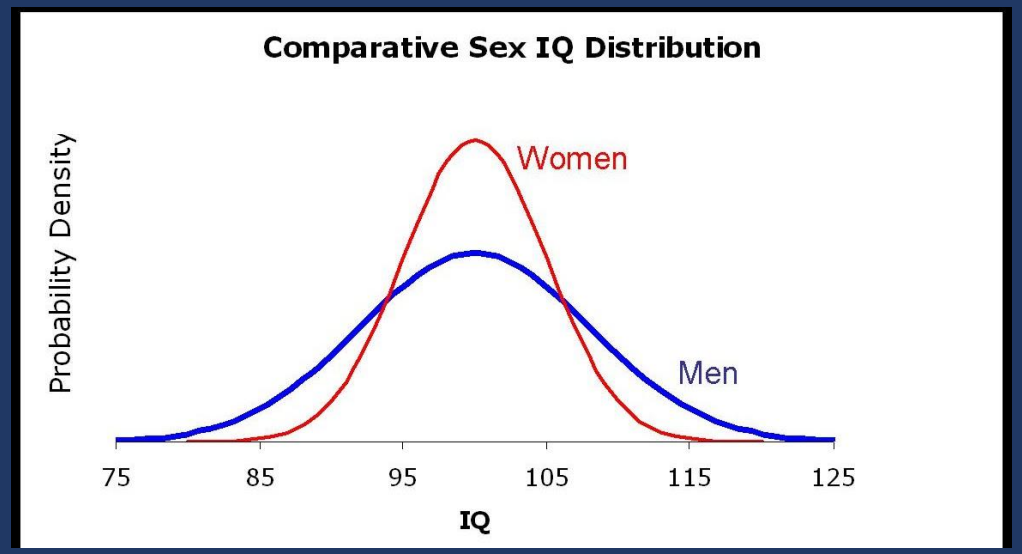
codecademy

An overall conclusion

Can we predict sex with education level and income?

- I think that we can with large propability predict sex with education level and income. Good example visualise this phenomenon is this graph that show comparative between women's and men's IQ. I think that this don't discriminate people only show fact which we can observes in world. I belive that women and men complement each other. Higher IQ means higher income and higher education level.

- Models can be used for statistical surveys



Can we predict education level with essay text word counts?

- I thnik that no, because accuracy level is too low. I consider probality that a lot of people can write a long answers which can be of little value and they don't need to have higher education to write long answers.

What other data can better to answer my question?

- I think that I need information about sex, income, age and job

codecademy

:)

I very apologise about my mistakes in English. I try to better write.

The End

codecademy