What is I learned about machine learning?_

Machine Learning Fundamentals

Damian Łukasik

04.11.2018

codecademy

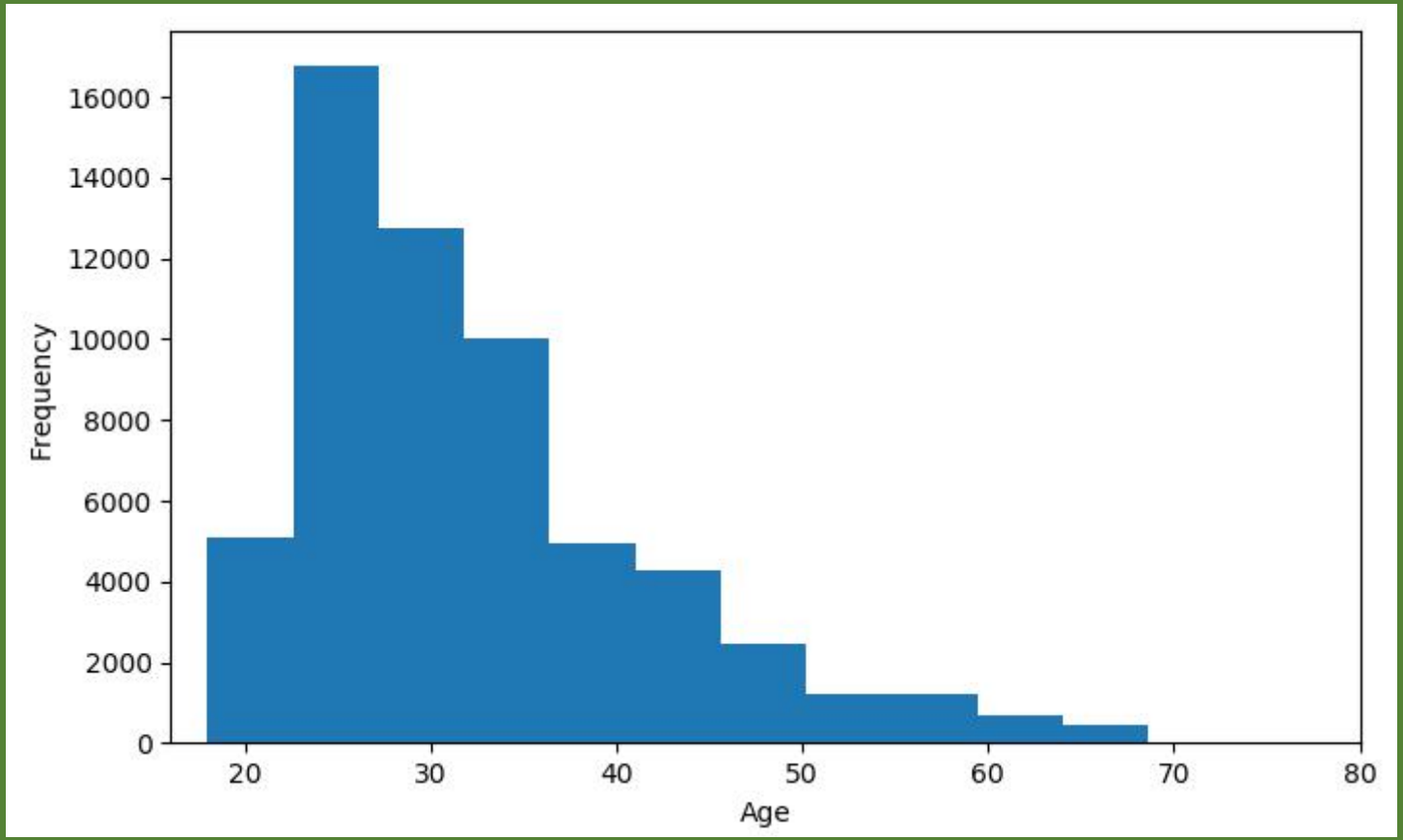
What contain the presentation?

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

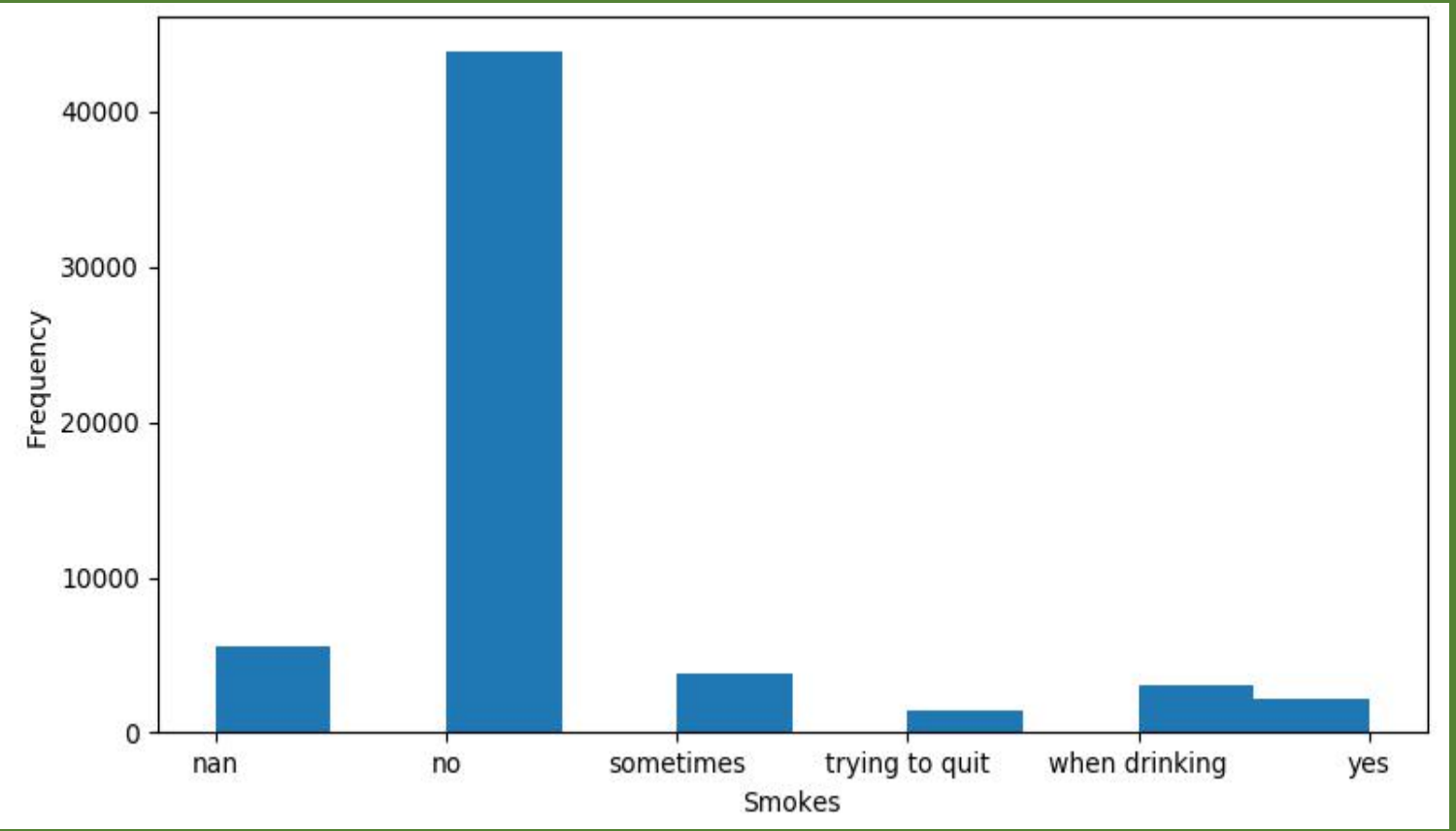


This graph present:

- Frequency of ages in dataset

codecademy

Graphs containing exploration of the dataset

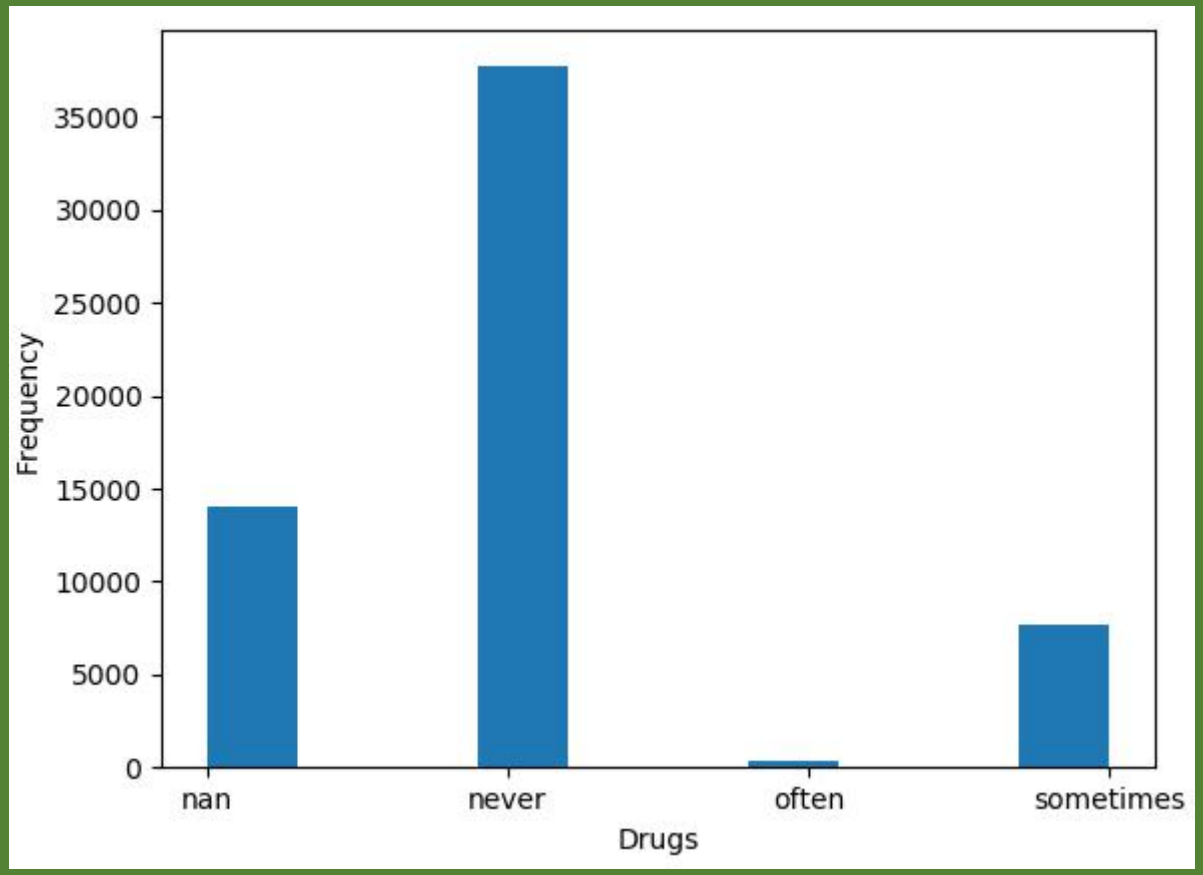


This graph present:

- Frequency of smokes in dataset

codecademy

Graphs containing exploration of the dataset

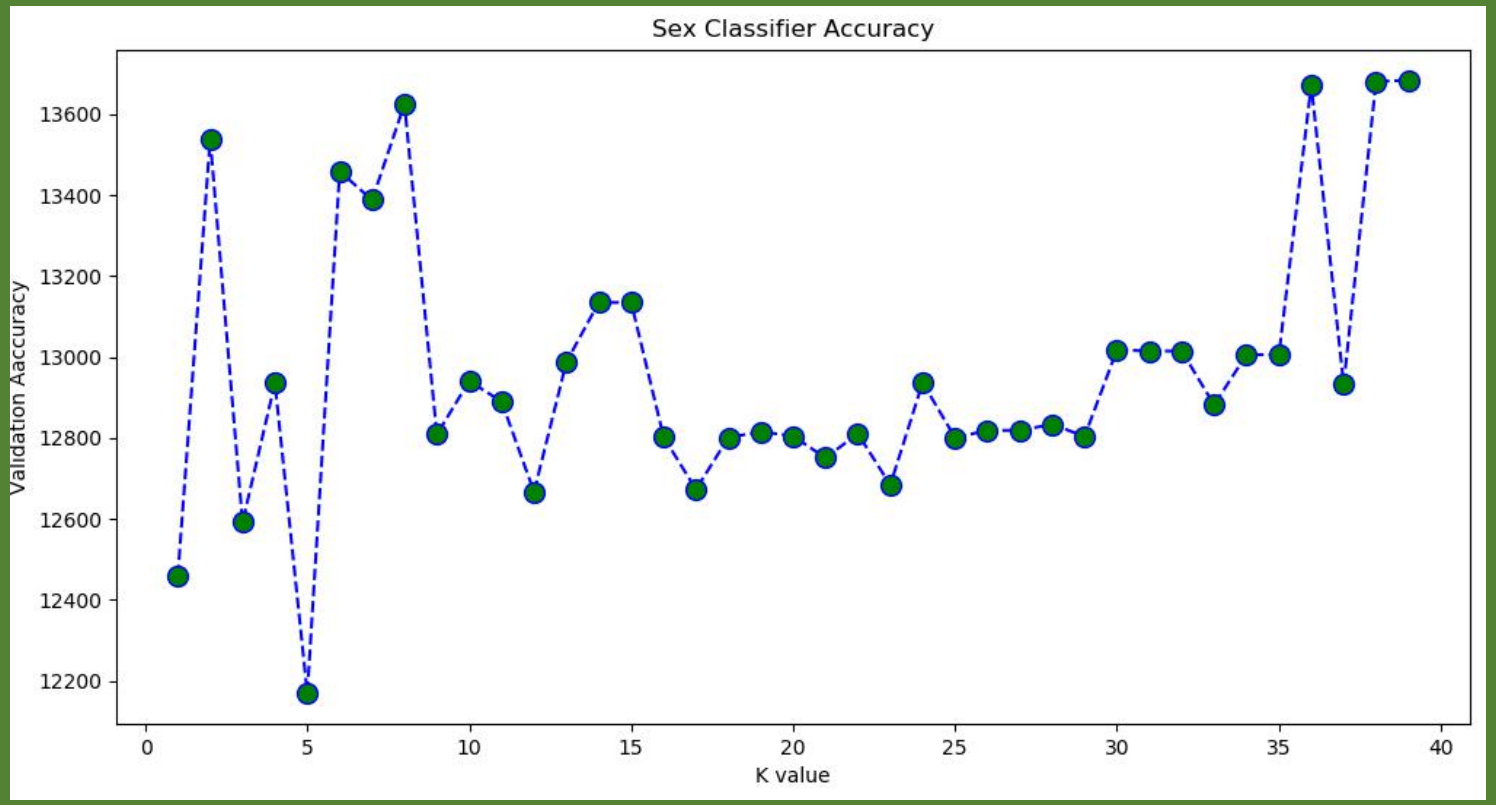


This graph present:

- 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

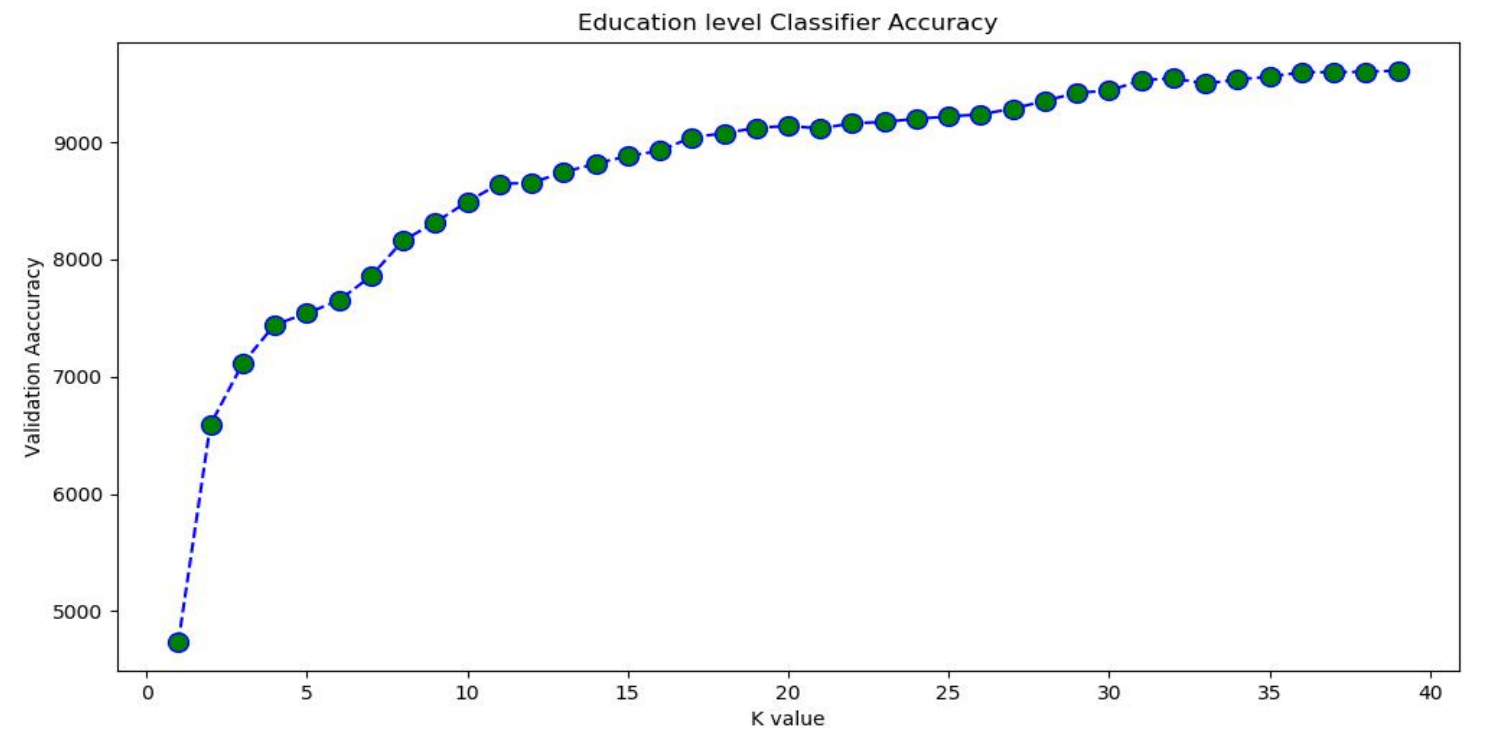
- I used method K-Nearest Neighbors

- 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

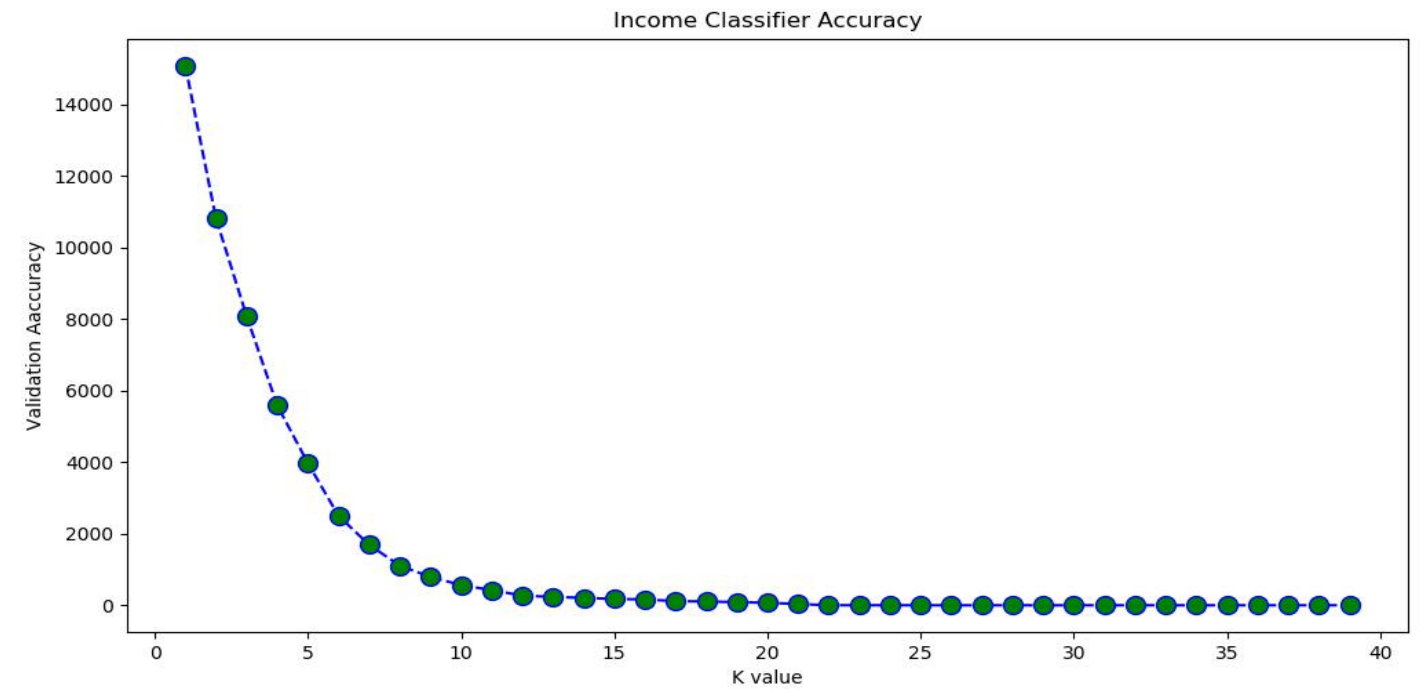
- I used method K-Nearest Neighbors

- 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

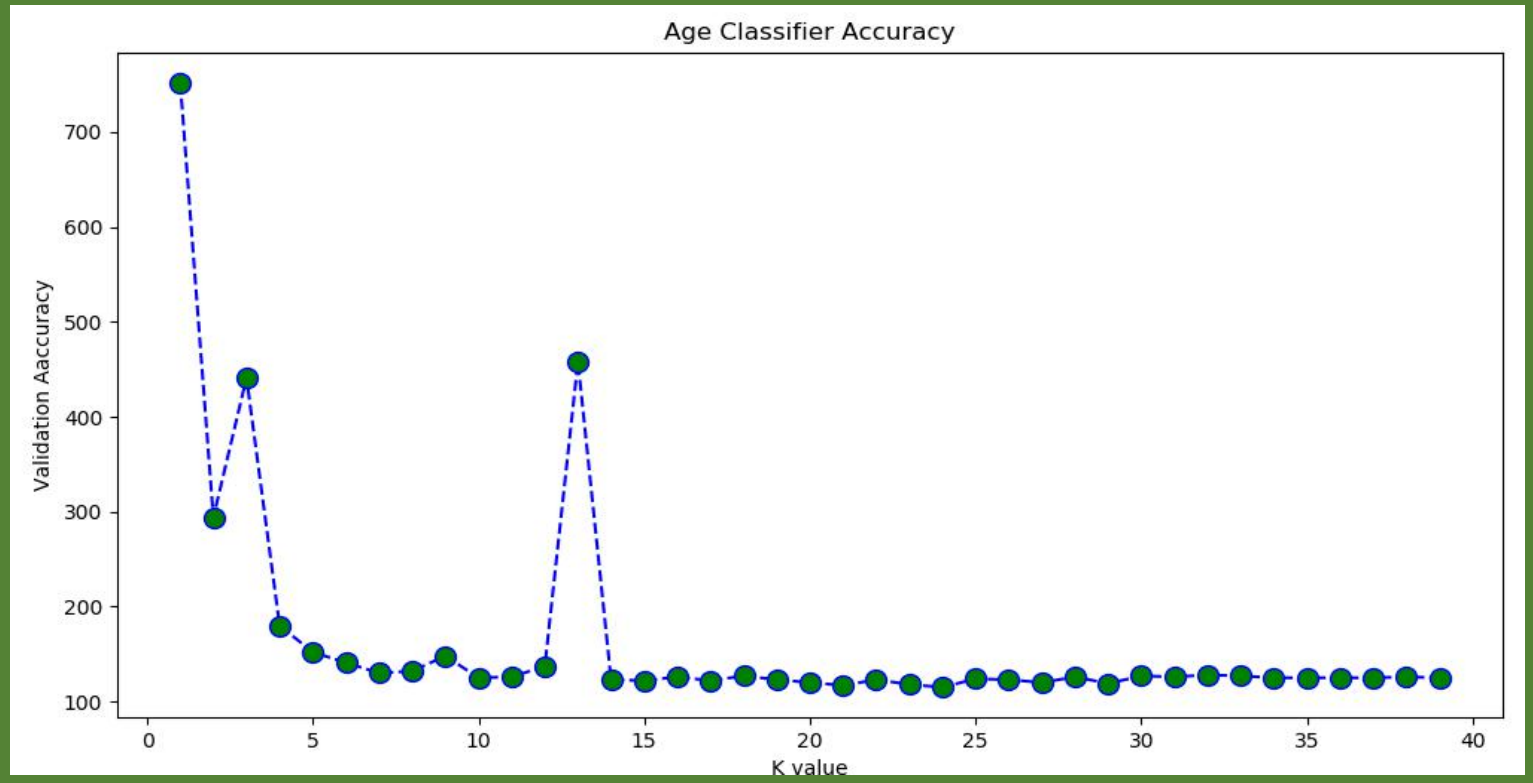
- I used method K-Nearest Neighbors Regression

- 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

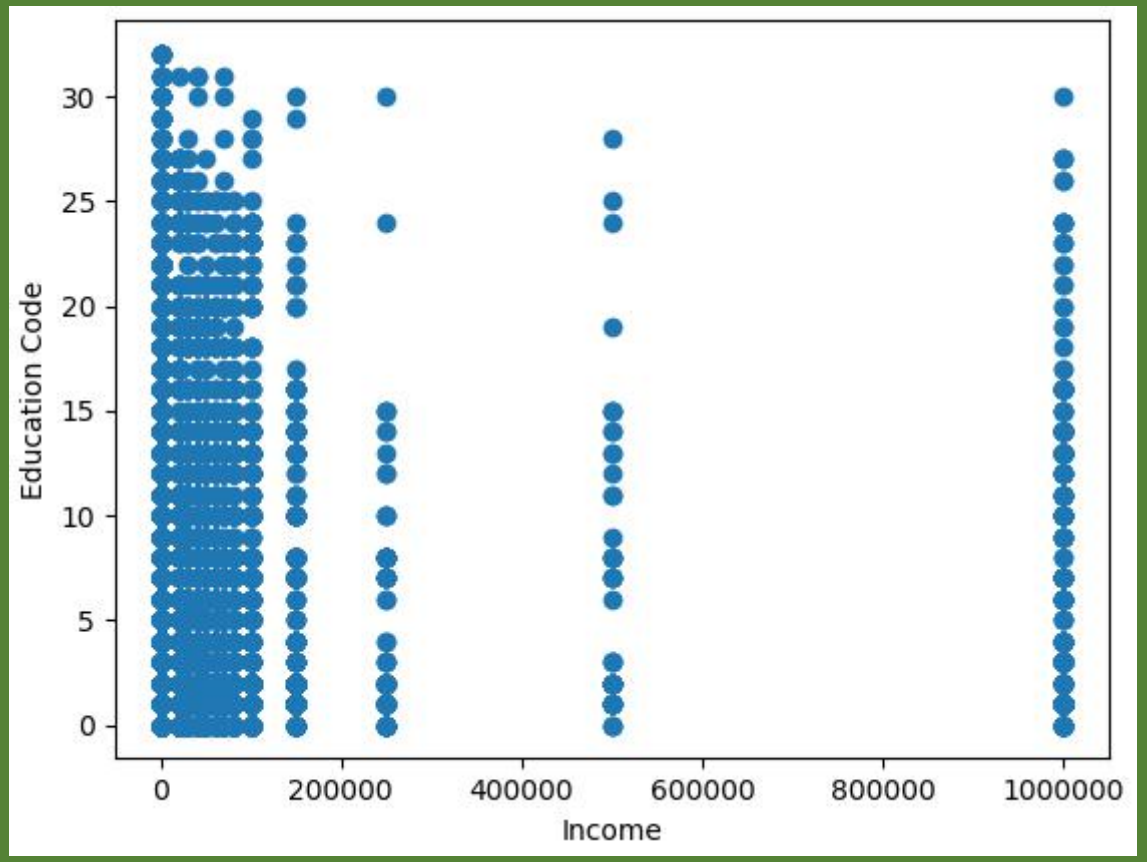
- I used method K-Nearest Neighbors Regression

- I solved Accuracy (number of correct rows) by

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

codecademy

Graphs containing exploration of the dataset

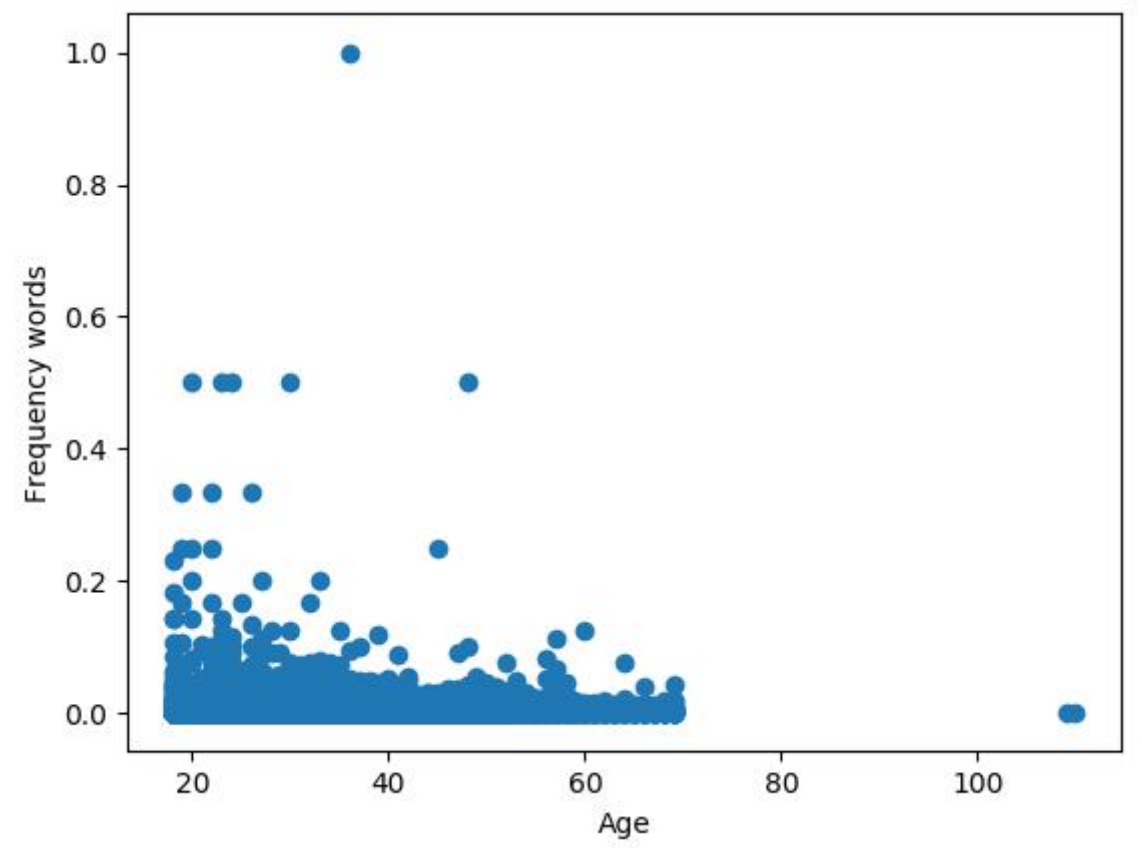


This graph present:

- Chart, where axis X is income and Y axis is education level

codecademy

Graphs containing exploration of the dataset



This graph present:

- Chart, where axis X is age and Y axis is frequency words

- Extreme values are data samples by centenarians (or incorrect data?). I had to use data normalization

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?

- In my project I used algorithms:	- Why? Because:
K-Nearest Neighbors	I were interested algorithm's time and accuracy. I think that it classified well unknown data. I think that it is good choose to predict education level by based essay text word counts, because exercise was solve fast
Support Vector Machines	This algorithm is very precision to classifying unknown data and I think that it did exercise very well. I was very interested about time algorithm's work, but I think that exist better solutions
Naive Bayes	This algorithm is the fastest which I used. I thought that it is a good algorithm to predict sex by based education level and income, because we had only two option (man or woman) similar as in lesson about Naive Bayes.
K-Nearest Neighbors Regression	I used this algorithm, because I thought that it is to help me to predict age and income, because these are numbers (continuous values). I used diferent weights because I wanted to try and to see work of algorithm.

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: 2.3185667991638184
```

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: 16.90957736968994
```

codecademy

The comparison between two classification approaches

K-Nearest Neighbors	Support Vector Machines
time to run the model accuracy, precision, and/or recall	time to run the model accuracy, precision, and/or recall

```
K-Nearest Neighbors - Question 2
Accuracy: 0.31908416048044O4
[[ 506 1695  251 ...    0    0    0]
 [1015 6586  972 ...    0    0    0]
 [ 377 2316  384 ...    0    0    0]
 ...
```

```
Support Vector Machines - Question 2
Accuracy: 0.40090916673617 48
[[   0 2667    0 ...    0    0    0]
 [   0 9613    0 ...    0    0    0]
 [   0 3461    0 ...    0    0    0]
 ...
```

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 - is faster	- 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 - is slower

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	with parameter weights = uniform
time to run the model accuracy, precision, and/or recall	time to run the model accuracy, precision, and/or recall
K-Nearest Neighbors Regression with parameter weights = distance - Question 3 Accuracy: 0.05568993799872105 time to run the model: 0.2108595371246338 K-Nearest Neighbors Regression with parameter weights = distance - Question 4 Accuracy: 0.006868475589803476 time to run the model: 1.790886640548706	K-Nearest Neighbors Regression with parameter weights = uniform - Question 3 Accuracy: 0.05570170091814932 time to run the model: 0.21787118911743164 K-Nearest Neighbors Regression with parameter weights = uniform - Question 4 Accuracy: 0.003687140562609742 time to run the model: 1.9148221015930176
- This model worked faster - This model is less accuracy	- This model worked slower - This model is more accuracy

codecademy

An overall conclusion

I write my conclusion by based results of accuracy

Can we predict sex with education level and income?

- I got accuracy more then 55 percent. I belive that exist relationship betwen sex and education level and income. Higher education means higher income, better job. I have no doubts

Can we predict education level with essay text word counts?

- I think that exist some relationship between education level and essay text word counts, because my accuracy was quite big. 30 or 40 percent is good result. I think that I need more analyse and research.

What other data can better to answer my question?

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

codecademy

An overall conclusion

Can we predict income with length of essays and average word length?

- When I see result I think that it is less probability because accuracy for 3 question is less than 6 percent, I don't sure can we be predict income by based length of essays and average word length. Probably these thinghts haven't relationship.

Can we predict age with the frequency of "I" or "me" in essays?

- I think that these things haven't relationship because accuracy for 4 question is less than 1 percent it is very small.

What other data can better to answer my question?

- I think that I need more data and analysis

codecademy

:)

I very apologise about my mistakes in English. I try to better write.
I hope that this presentation will better
enjoy weekend

The End

codecademy