Gender equality and Machine Learning

le wagon

Agenda

- ✓ What is (and is **NOT**) Machine Learning
- ✓ A context on Gender Equality
- ✓ Live Code! 🚀

le wagon

Agenda

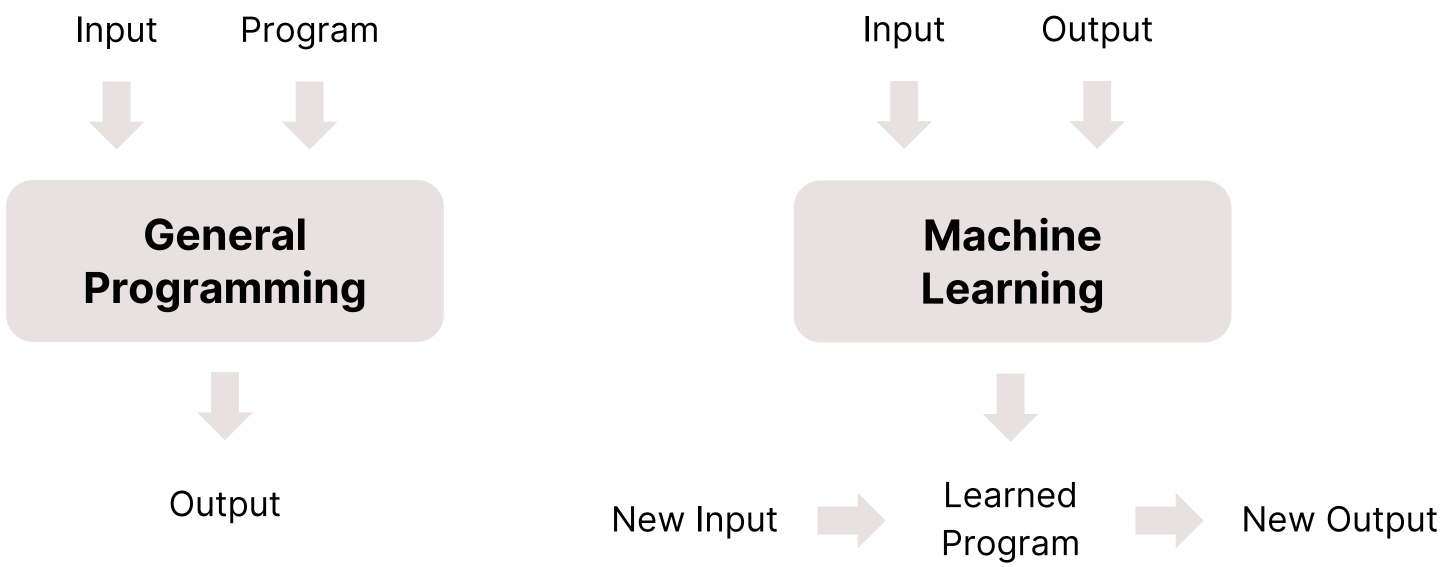
- ✓ **What is (and is NOT) Machine Learning**
- ✓ A context on Gender Equality
- ✓ Live Code! 🚀

le wagon

"[a] field of study that gives computers the **ability to learn without being explicitly programmed**"

Arthur Samuel (1959)

le wagon


Input
Program
General Programming
Output
Input
Output
Machine Learning
New Input
Learned Program
New Output

How **Machine Learning** works ?

Take basics **Observations**

My Age / Height ratio 📏

Feature

Input

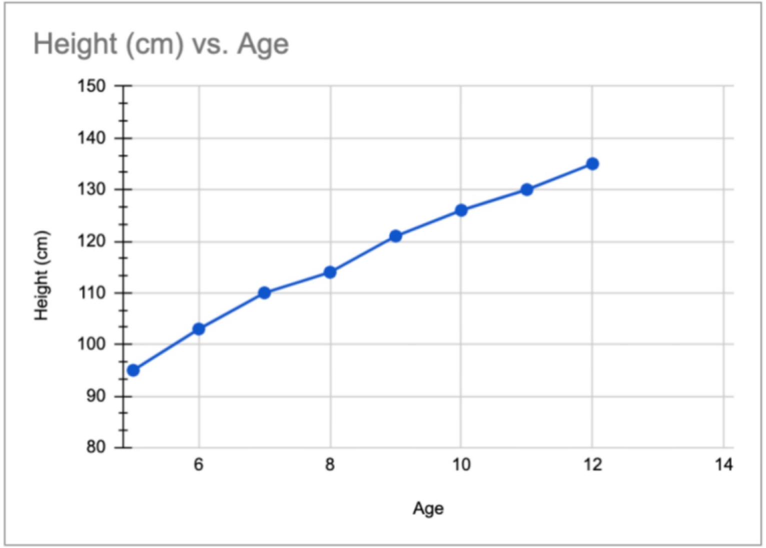
Age	Height (cm)
5	95
6	103
7	110
8	114
9	121
10	126
11	130
12	135

Target

Output

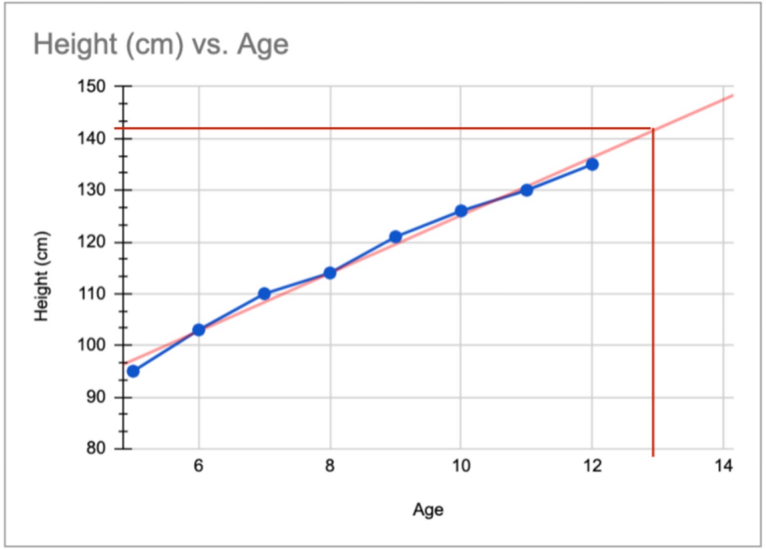
Infer a Math function

How tall was I at 13? 🙆🏻‍♂️



Extrapolate for new observations

Easy guess, isn't it ?



What if there are 150 different **features** and a million observations for 2 million different people?

What if there are 150 different **features**

and a million observations

for 2 million different people?

Every day.

What if there a  ifferent **features**

and a mi ervations

for 2 million rent people?

Every day.

Machine Learning is already all around us

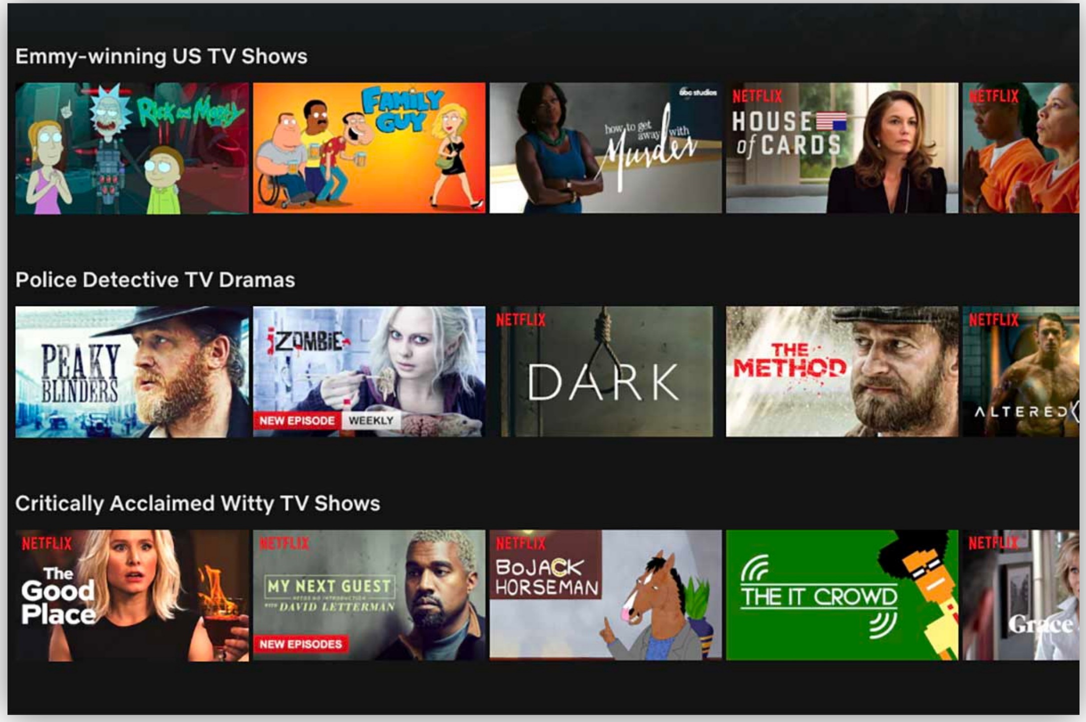
le wagon

Computer Vision

Classifying and detecting visual data

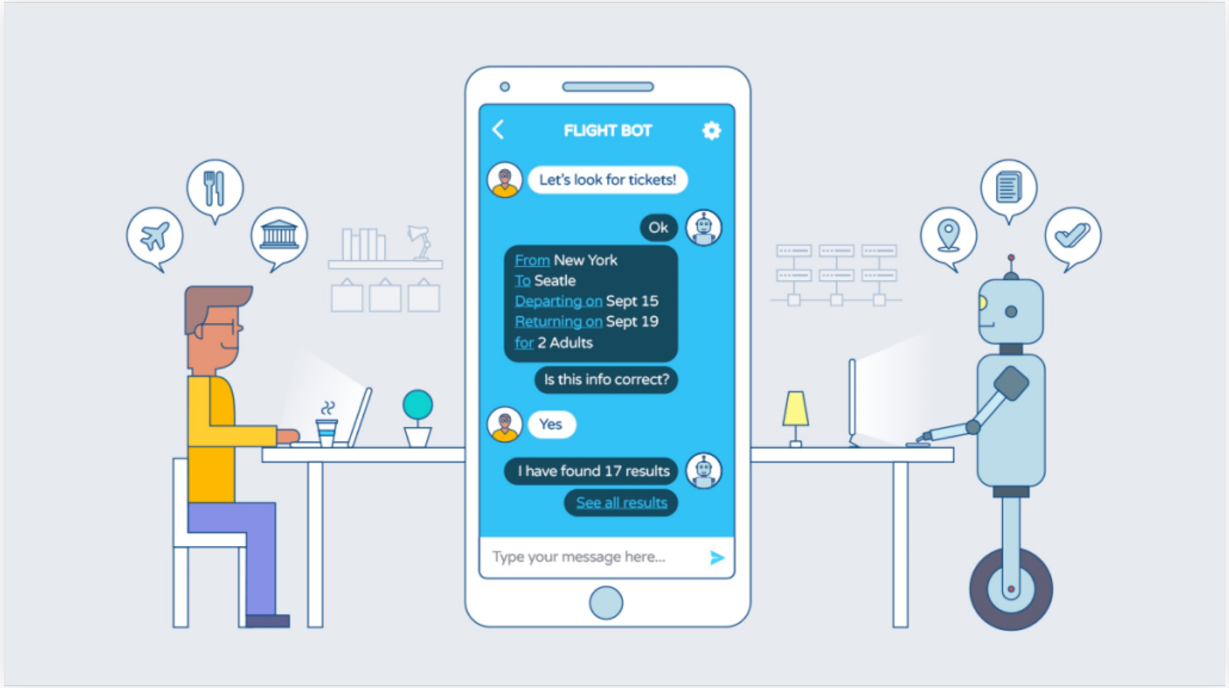
Recommendation Engine

Predicting your next action



Natural Language Processing

Finding meaning in text



Time Series

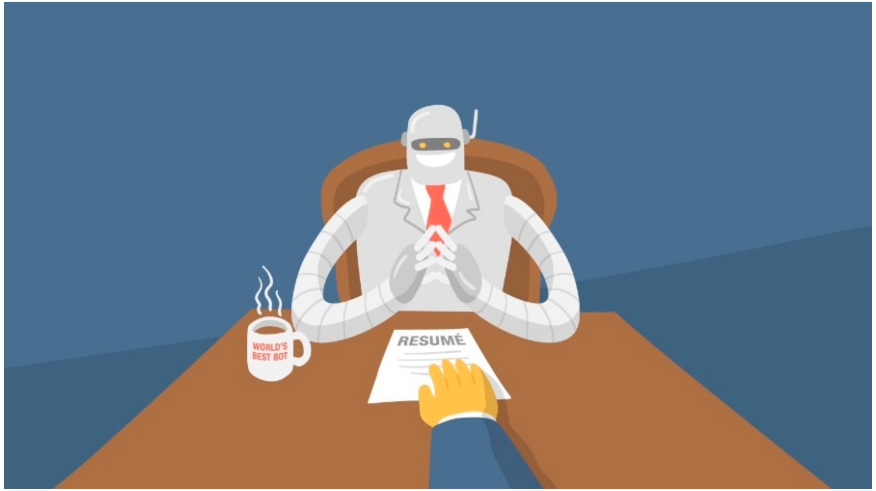
Can past change predict future change

Anomaly Detection

"Predicting patterns" in reverse

When Machine Learning becomes just hype

Missing ground truth



Machine Learning for hiring the perfect candidate!

Things have changed

Let's take a drug dose analysis 💊

Day 1 - 10mg

Day 2 - 15mg

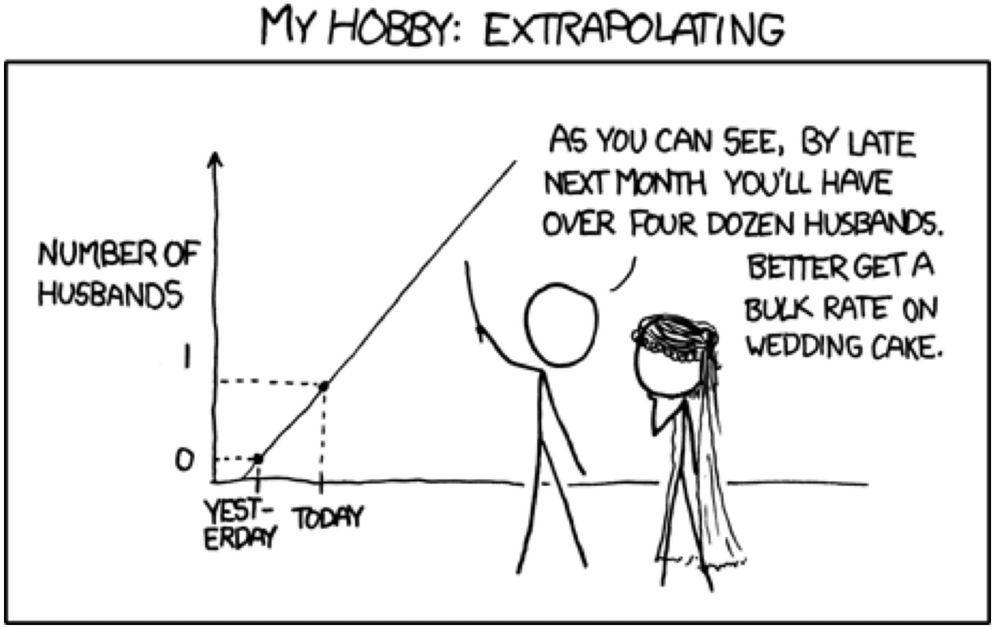
Day 3 - 20mg

Day 4 - 25mg

Day 5 - 30mg

Day 6 - ?

Things have changed



Things have really changed

TECH \ CORONAVIRUS

The algorithms big companies use to manage their supply chains don’t work during pandemics

The data the algorithms use isn’t reliable

By Nicole Wetsman | Apr 27, 2020, 1:25pm EDT

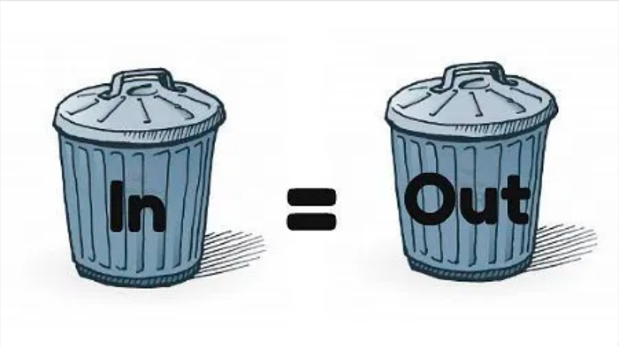
Listen to this article

SHARE

le wagon

Most firms that think they want advanced AI/ML really just need linear regression on cleaned-up data

— Robin Hanson (@robinhanson) November 28, 2016



Agenda

- ✓ What is (and is NOT) Machine Learning
- ✓ **A context on Gender Equality**
- ✓ Live Code! 🚀

le wagon

What is Gender Equality?

Let's first do a quick research on the topic.

"Equal rights, responsibilities and opportunities of women and men and girls and boys."

Source: European Institute for Gender Equality

“ **Equality does not mean that women and men will become the same** but that women’s and men’s rights, responsibilities and opportunities will not depend on whether they are born female or male. (...) ”

Source: European Institute for Gender Equality

Some references to look at

👉 https://www.coe.int/en/web/genderequality/themes

👉 https://www.humanrightscareers.com/issues/causes-gender-inequality/

👉 https://www.mckinsey.com/featured-insights/diversity-and-inclusion/ten-things-to-know-about-gender-equality

👉 https://eige.europa.eu/news/covid-19-derails-gender-equality-gains

Data Science use cases

Some examples of how Data Science can have an impact on Gender Equality.

Data Science use cases.

Only a few examples.

Make gender data visible.

check this report

Predict gender based violence.

check this paper

Know the causes for gender pay gap.

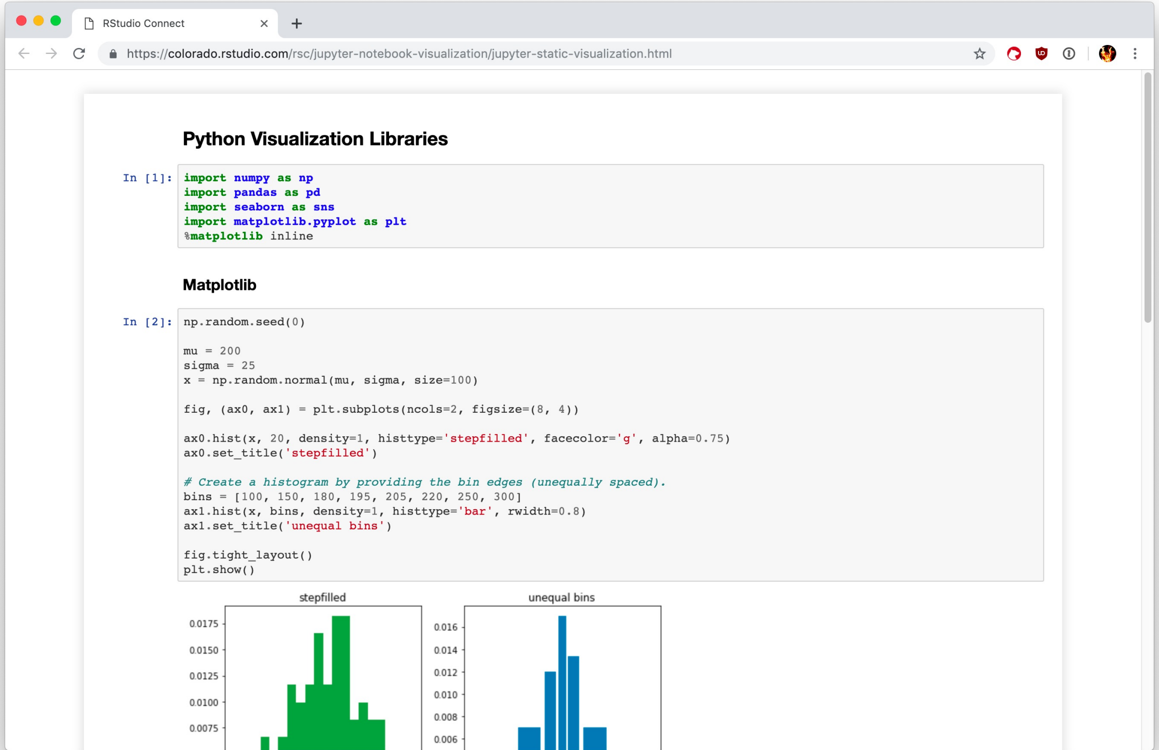
check these articles

le wagon

Agenda

- ✓ What is (and is **NOT**) Machine Learning
- ✓ A context on Gender Equality
- ✓ **Live Code!** 🚀

Ready to get nerdy? 🤓



Python Visualization Libraries

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

Matplotlib

In [2]:
```
np.random.seed(0)

mu = 200
sigma = 25
x = np.random.normal(mu, sigma, size=100)

fig, (ax0, ax1) = plt.subplots(ncols=2, figsize=(8, 4))

ax0.hist(x, 20, density=1, histtype='stepfilled', facecolor='g', alpha=0.75)
ax0.set_title('stepfilled')

# Create a histogram by providing the bin edges (unequally spaced).
bins = [100, 150, 180, 195, 205, 220, 250, 300]
ax1.hist(x, bins, density=1, histtype='bar', rwidth=0.8)
ax1.set_title('unequal bins')

fig.tight_layout()
plt.show()
```

le wagon

Before Machine Learning
We can do some analysis 🔍

What's the data

Understand what we have

- ✓ How many rows and columns?
- ✓ What are the columns we have?
- ✓ What are the averages?
- ✓ What are the minimums and maximums?

Methods to explore your data:

In [ ]:
```
# to get the number of rows, columns
salaries.shape
```

In [ ]:
```
# to get the columns and their data types
salaries.dtypes
```

In [ ]:
```
# to get a readable summary of your data
round(salaries.describe())
```

Visualize the data

Get an intuition of what it tells you

- ✓ How many of each category we have?
- ✓ Do any two columns relate to each other?
- ✓ Do some categories influence the output?
- ✓ First step of any Data Scientist!

seaborn

Visualize the data

Get an intuition of what it tells you

✓ How many of each category we have?

```
In [ ]: # to count the distribution of values in a column
        sns.countplot(data=salaries, x='Column Name')
```

✓ Do any two columns relate to each other?

```
In [ ]: # to see the relation of one column to another
        sns.scatterplot(data=salaries, x='Input Column', y='Output Column')
```

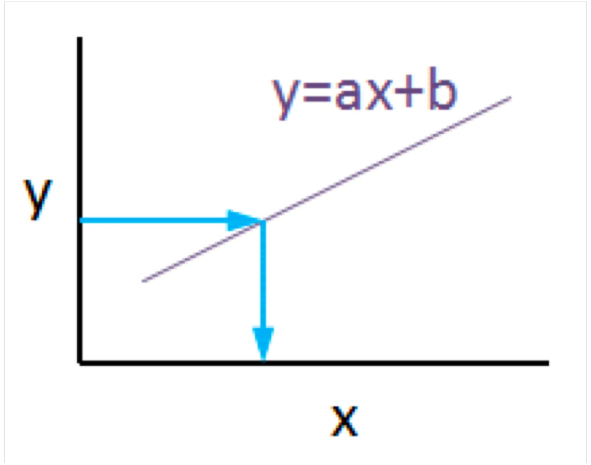
✓ Do some categories influence the output?

```
In [ ]: # to add a category to the visualization, we use `hue`
        sns.scatterplot(data=salaries, x='Input Column', y='Output Column', hue='Category Column')
```

Challenge 1:

Predicting **salaries** 💰

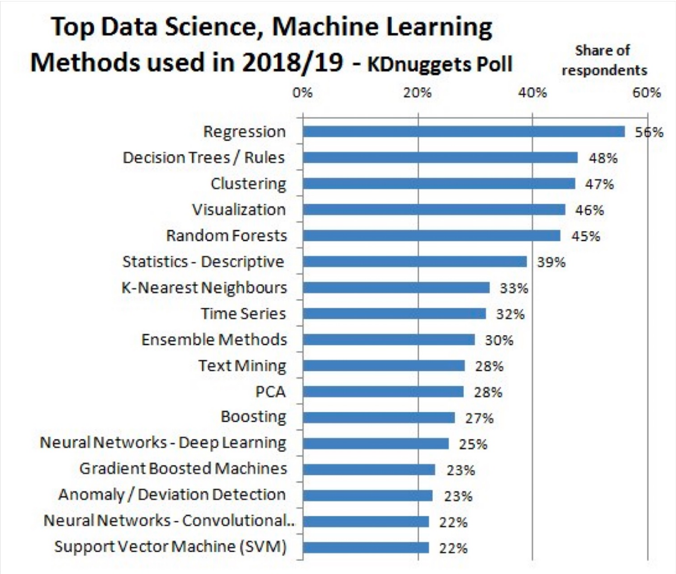
Linear Regression



Most firms that think they want advanced AI/ML really just need linear regression on cleaned-up data

— Robin Hanson (@robinhanson) November 28, 2016

Why Regression?



le wagon

Visual demo on setosa.io! 🎥

le wagon

Road to Machine Learning

1. Select the **features** and **targets**
2. **Import** the model from Sklearn
3. **Train** the model
4. **Score** the model's performance
5. **Predict** with new data

Road to Machine Learning

1. Select the **features** and **targets**

Setting our features (inputs):

```
In [ ]: # we can select all needed columns...
        features = salaries[["Gender", "Age", "Department_code", "Years_exp", "Tenure (months)"]]
```

```
In [ ]: # ...or we can simply drop the not needed!
        features = salaries.drop(["Department", "Gross"], axis="columns")
```

Setting our target (output):

```
In [ ]: # we can simply select the column we need
        target = salaries["Gross"]
```

le wagon

Road to Machine Learning

1. Select the **features** and **targets**
2. **Import** the model from Sklearn
3. **Train** the model
4. **Score** the model's performance
5. **Predict** with new data

Once we find the model we need, it's easy to import

```
In [ ]: # the syntax looks like this
        from sklearn.MODEL_TYPES import MODEL_YOU_NEED
```

```
In [ ]: # with Linear Regression we need this
        from sklearn.linear_model import LinearRegression
```

After importing, we need to **initialize** the model, like this:

```
In [ ]: model = LinearRegression()
```

Road to Machine Learning

1. Select the **features** and **targets**
2. **Import** the model from Sklearn
3. **Train** the model
4. **Score** the model's performance
5. **Predict** with new data

To train the model, we use the `.fit` method:

```
In [ ]: model.fit(features, target)
```

The `model` then finds the **best fitting** line between features and target

le wagon

Road to Machine Learning

1. Select the **features** and **targets**
2. **Import** the model from Sklearn
3. **Train** the model
4. **Score** the model's performance
5. **Predict** with new data

To score the model, we use the `.score` method:

```
In [ ]: model.score(features, targets)
```

We need to give it some **test data** to do the scoring

Road to Machine Learning

1. Select the **features** and **targets**
2. **Import** the model from Sklearn
3. **Train** the model
4. **Score** the model's performance
5. **Predict** with new data

To predict -- you guessed it ;)

We use the `.predict` method:

```
In [ ]: model.predict(new_data)
```

`new_data` is the info we want our model to use to predict a new target (output).

In our case, it's a new `hire` !

le wagon

Road to Machine Learning

1. Select the **features** and **targets**
2. **Import** the model from Sklearn
3. **Train** the model
4. **Score** the model's performance
5. **Predict** with new data
6. **Explaining** the model

The things that a Linear Regression model "*learns*" are **coefficients** and **intercept**.

The **coefficients** show how each feature influences the target:

```
In [ ]: model.coef_
```

The **intercept** shows what would the target be when all features are at zero:

```
In [ ]: model.intercept_
```

Optional Challenge 2:

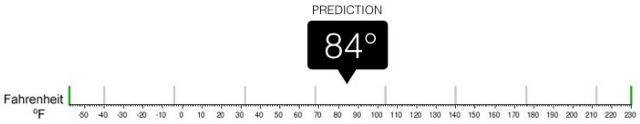
Customer churn 🛬

le wagon

Regression vs. Classification

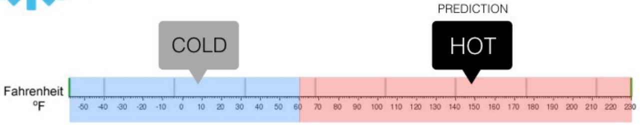
Regression

What is the temperature going to be tomorrow?



Classification

Will it be Cold or Hot tomorrow?



You now have the tools

Your turn! 🚀