MACHINE LEARNING

Mathieu Cattenoz
mathieu.cattenoz@epita.fr

INTRODUCTION

Mathieu CATTENOZ

- Engineer degrees from Supélec (France) and TUM (Germany)
- PhD in signal processing
- R&D engineer in the aeronautics industry (Thales, Airbus DS, MBDA)
- Technical consultant, mainly for IoT startups

SCHEDULE OF TODAY

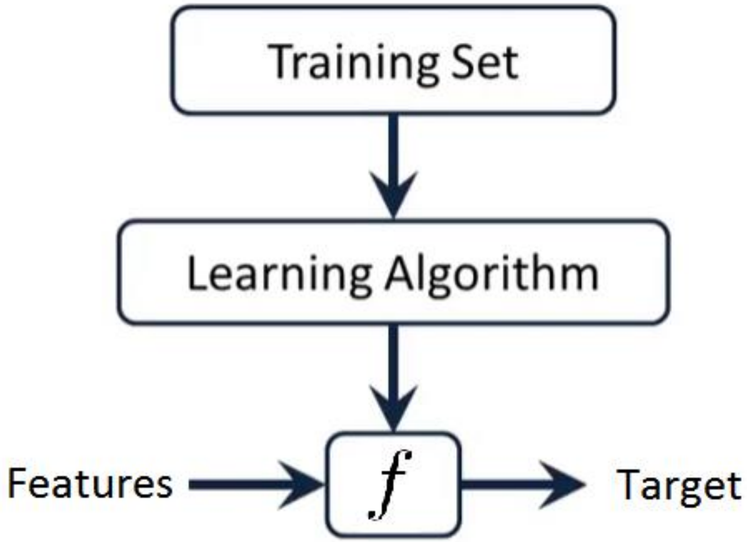
Program

- Introduction on ML
- Tools & libraries
- Data exploration
  - Overview
  - Implementation
  - Practice

WHAT IS ML?

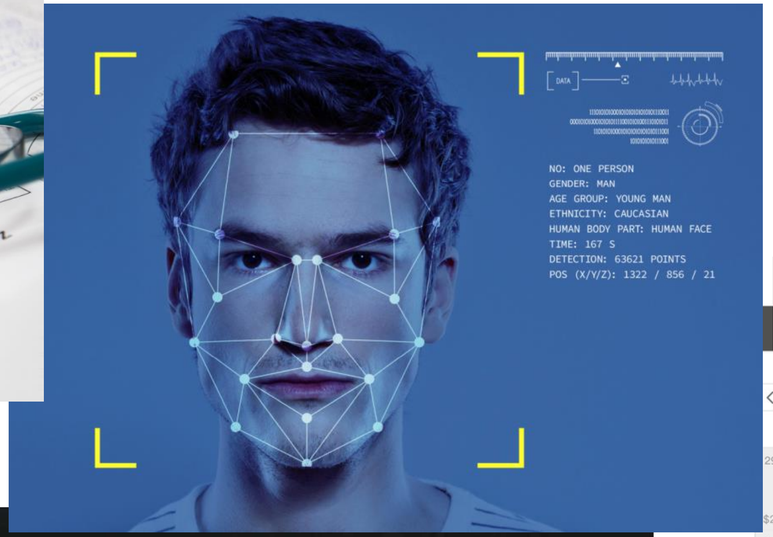
A definition

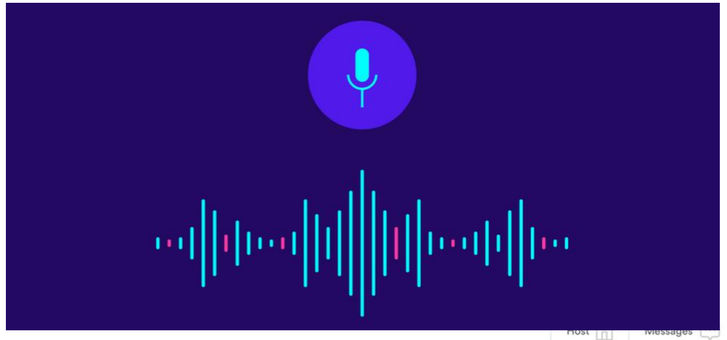
- Machine Learning (ML) is a subset of artificial intelligence that encompasses techniques aiming at making predictions after having built models based on sample data.

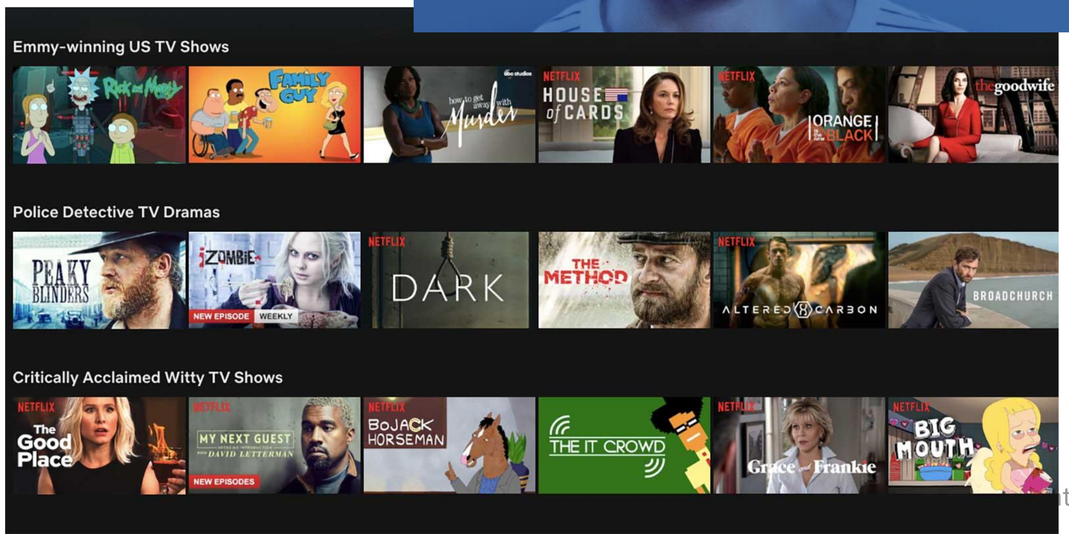


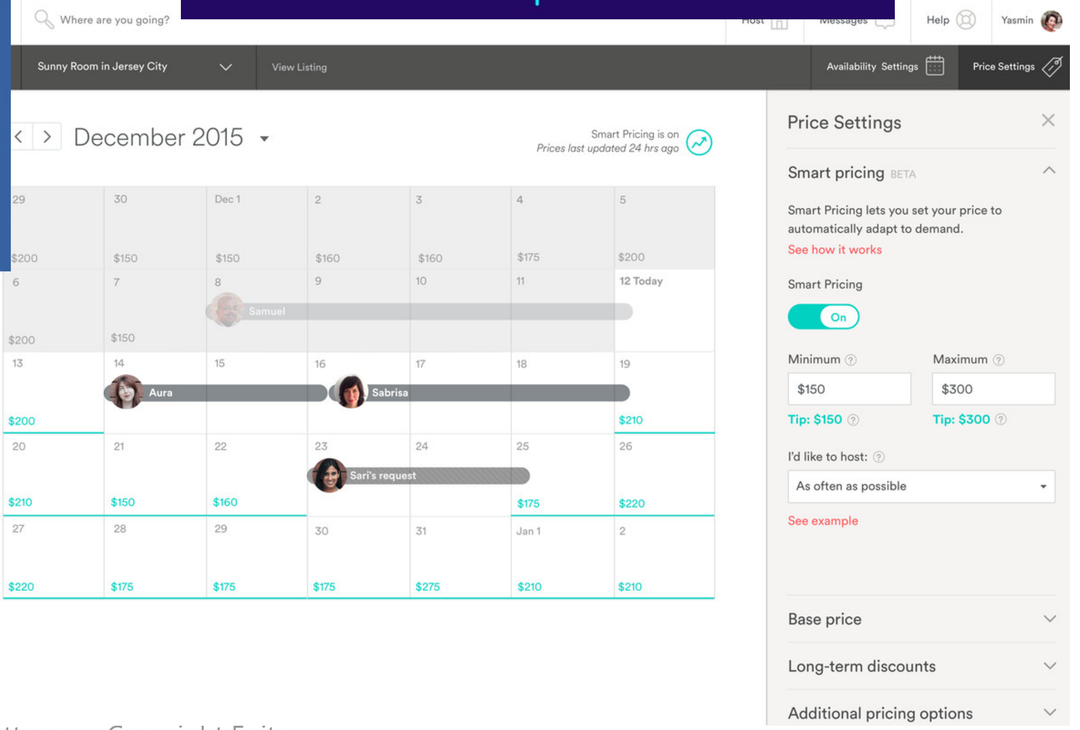
WHERE IS ML?

Use cases



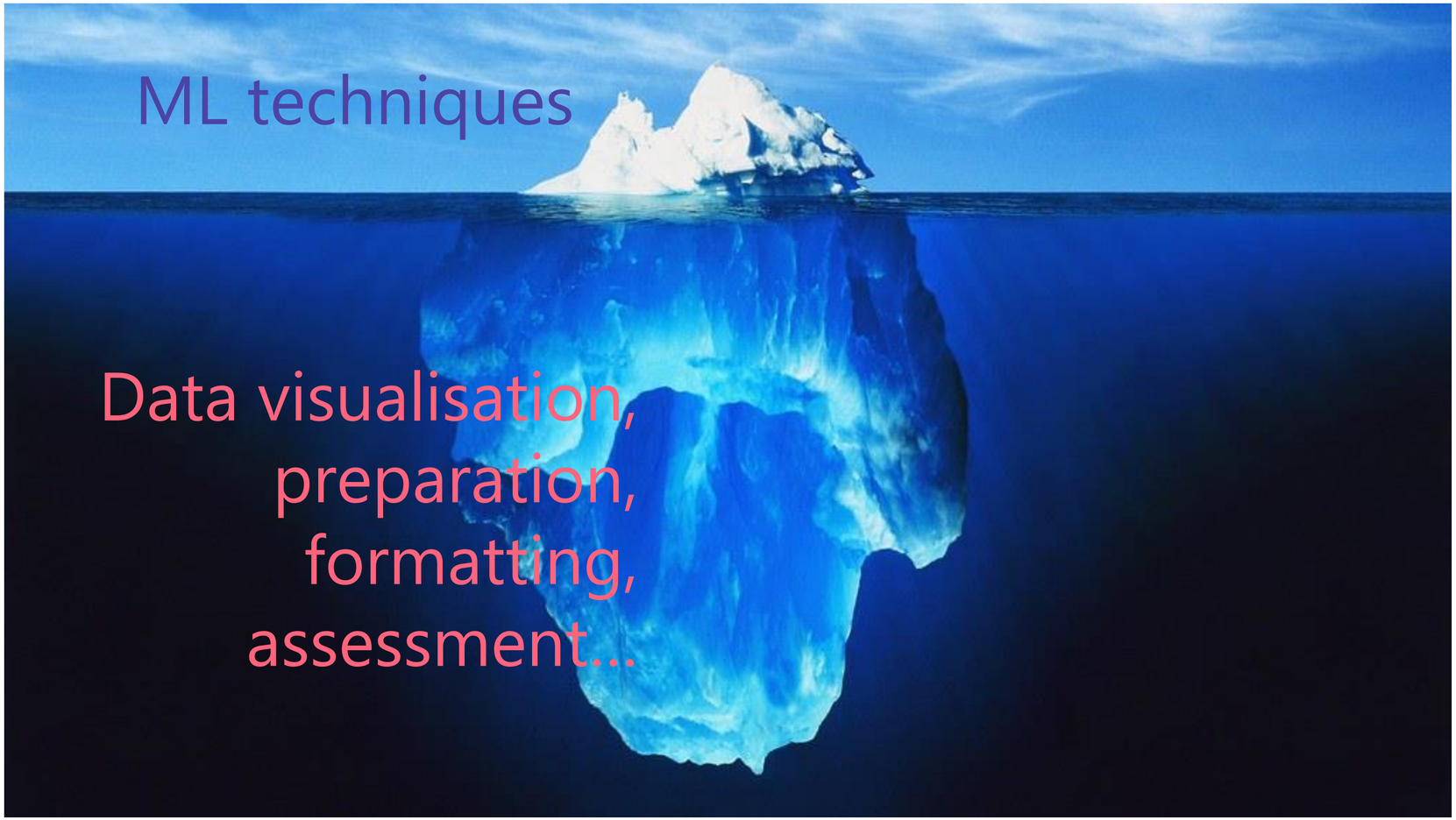






HOW TO LEARN ML?

The underestimated part of ML



➡ **Practice**

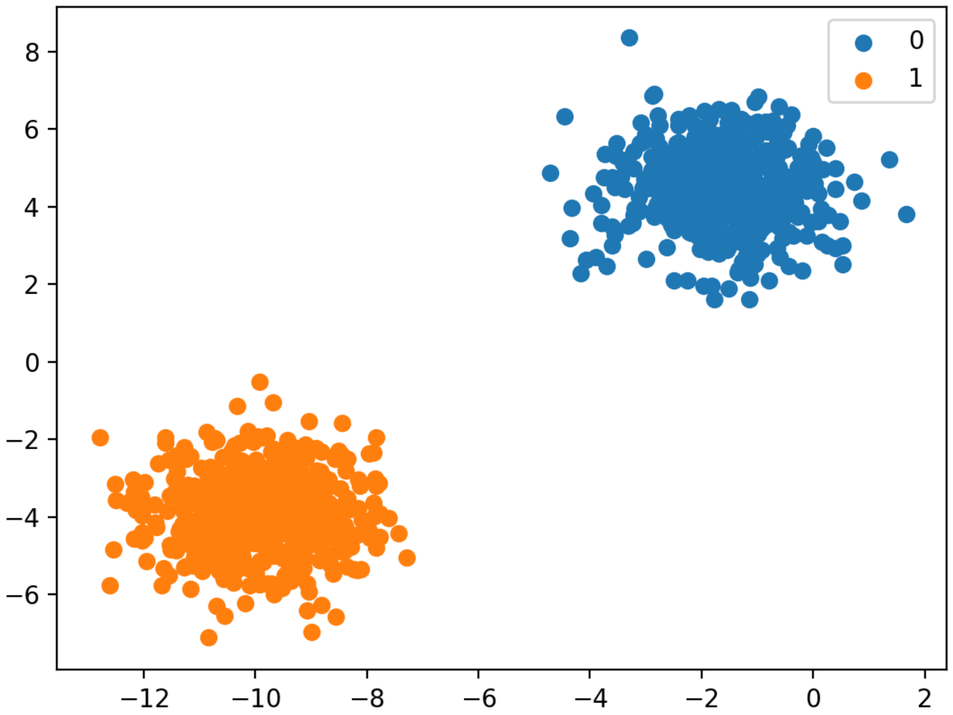
ML WORKFLOW

Usual process

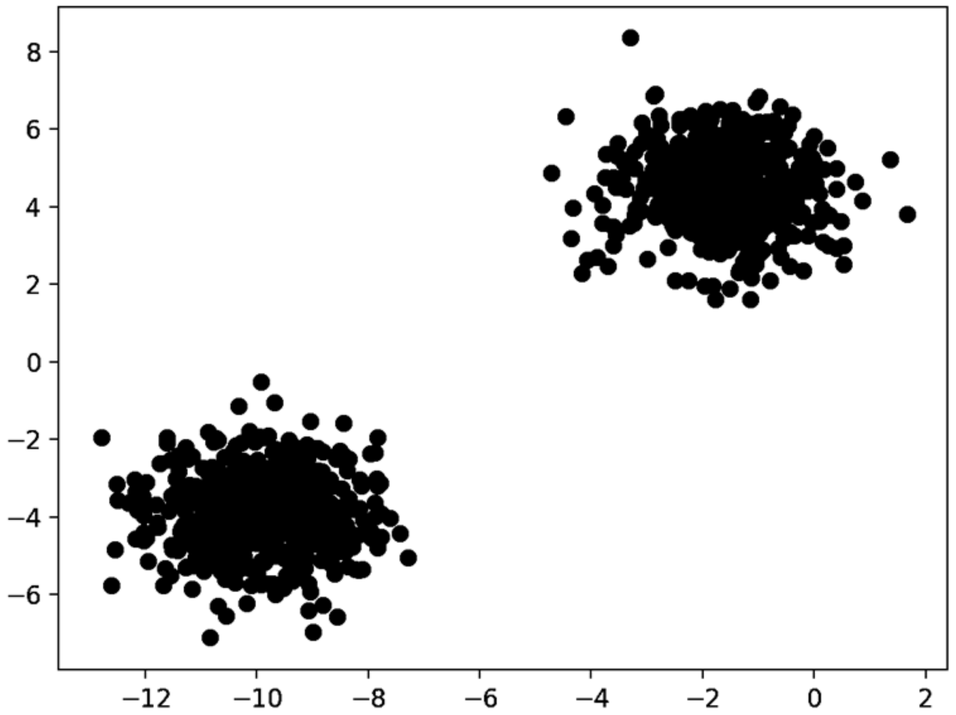
- Collect data
- Clean and prepare the data
- Pick a model
- Train the model on the data
- Test and evaluate the model
- Improve the model

SUPERVISED VS UNSUPERVISED LEARNING

Labeled vs. unlabeled



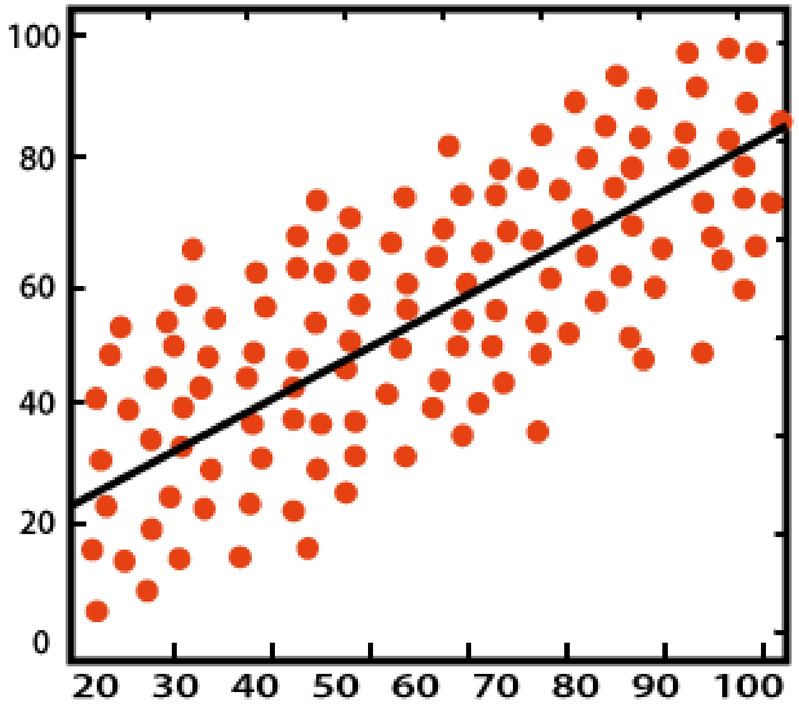
Supervised learning



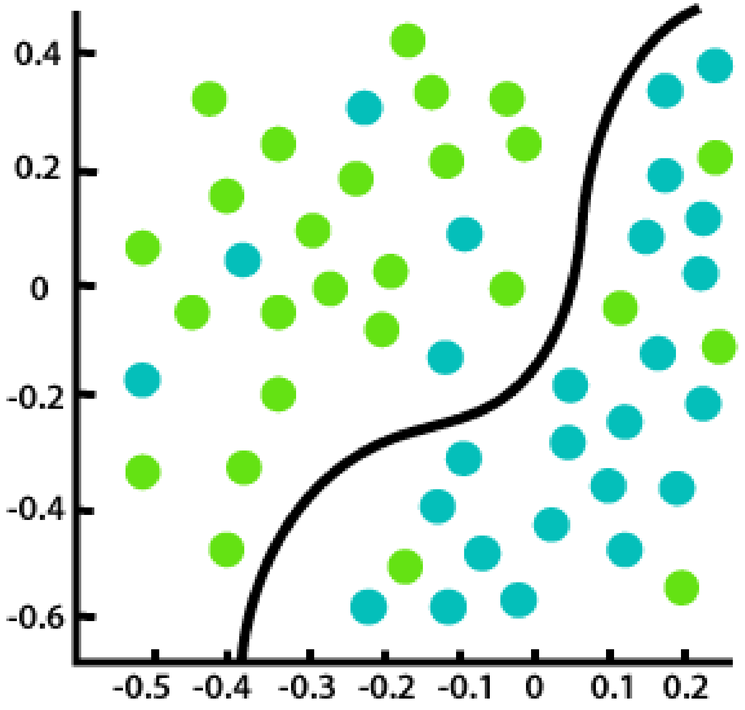
Unsupervised learning

SUPERVISED LEARNING

Continuous vs. discrete output



Regression



Classification

COURSE PROGRAM

Structure

PREPARATION	Data exploration
	Data preprocessing
REGRESSIONS	Linear regression with one variable
	Multiple and polynomial regressions
	Regression model assessment
CLASSIFICATIONS	Logistic regression
	Classification model assessment
	Discriminant Analysis
	k-NN, Decision Tree, SVM
CLUSTERING	k-means, hierarchical clustering
DIMENSIONALITY REDUCTION	Principal Components Analysis + Extensions
ALL NOTIONS	Final assignment

TOOLS & LIBRARIES

Installation

- Python 3.8 (this version precisely). Make sure it includes pip (needed to install libraries).
- A development IDE: e.g. Visual Studio Code, Spyder.
- Jupyter notebook

```
pip install jupyterlab
```

- Libraries: scikit-learn, numpy, matplotlib, pandas

```
python -m pip install pandas
python -m pip install numpy
python -m pip install -U scikit-learn
python -m pip install -U matplotlib
```

TOOLS & LIBRARIES

Import

- Import the libraries you need before to use it in your code.

```
import numpy as np                      ⟵ operations on array
import pandas as pd                     ⟵ data manipulation
import sklearn                          ⟵ ML algorithms
import matplotlib.pyplot as plt         ⟵ plotting
import …
```

METHODS

Practice

- Schedule:
  - 15 min: homework correction
  - 30 min: theoretical background
  - 15 min: example of implementation (by the teacher)
  - 60 min + homework: practice (by the students)

- Material:
  - Slides and implementation uploaded on Teams after each session
  - Practice subject on Teams as assignment (with deadline)
  - Practice correction uploaded on Teams after deadline

- Grades:
  - Practices: 50 %
  - Final exam: 50 %

METHODS

Interactions

- Don’t hesitate to ask questions during the teacher presentation
- Don’t hesitate to ask questions during the practice
- After the session: don’t hesitate to contact me on Teams or by email

COURSE PROGRAM

Structure

PREPARATION
- Data exploration
- Data preprocessing

REGRESSIONS
- Linear regression with one variable
- Multiple and polynomial regressions
- Regression model assessment

CLASSIFICATIONS
- Logistic regression
- Classification model assessment
- Discriminant Analysis
- k-NN, Decision Tree, SVM

CLUSTERING
- k-means, hierarchical clustering

DIMENSIONALITY REDUCTION
- Principal Components Analysis
- Extensions

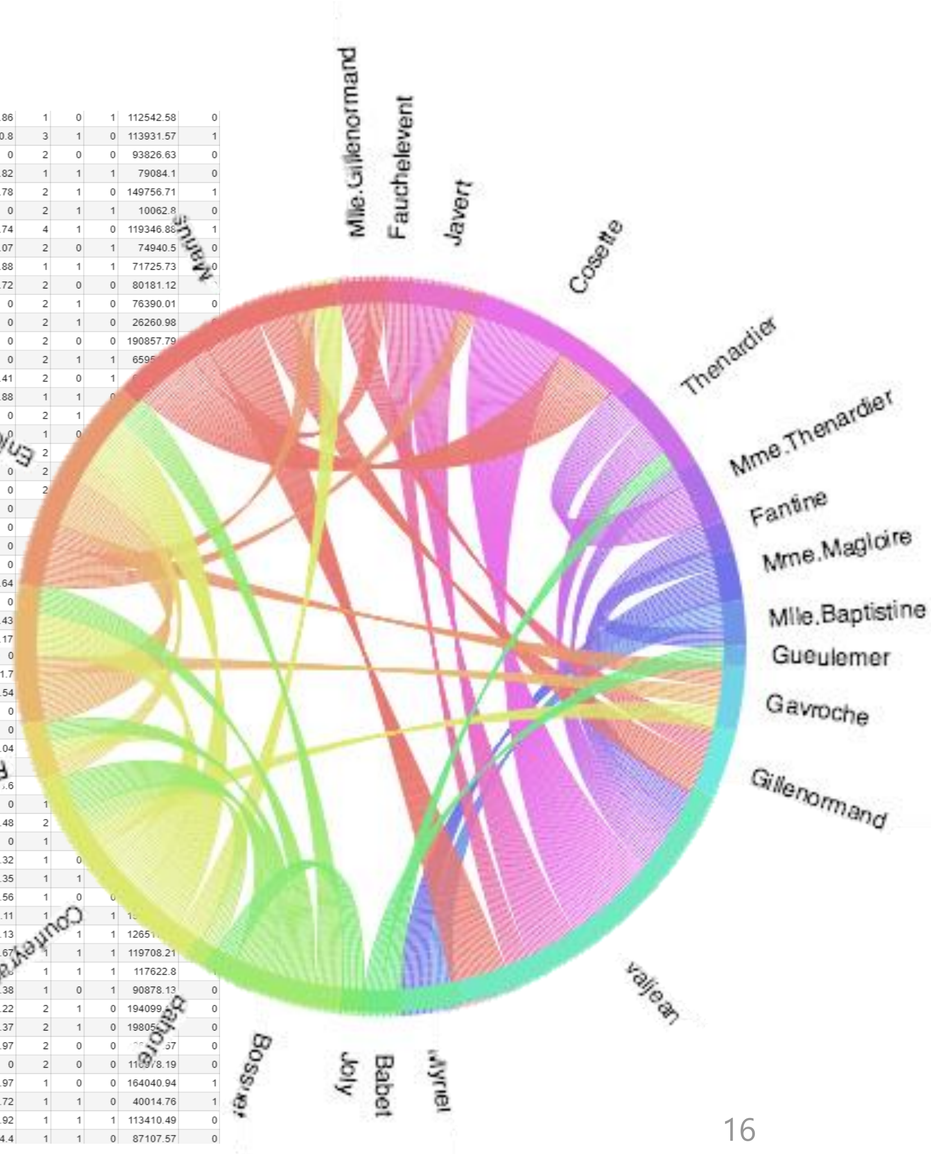
ALL NOTIONS
- Final assignment

EXPLORE THE DATA SET

- Understand your data
- Identify the variables
- Get some intuition (correlation, etc.)
- Identify problems in the data

Why?

2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
3	15619304	Onio	502	France	Female	42	8	159660.8	3	1	0	113931.57	1
4	15701354	Boni	699	France	Female	39	1	0	2	0	0	93826.63	0
5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.1	0
6	15574012	Chu	645	Spain	Male	44	8	113755.78	2	1	0	149756.71	1
7	15592531	Bartlett	822	France	Male	50	7	0	2	1	1	10062.8	0
8	15656148	Obinna	376	Germany	Female	29	4	115046.74	4	1	0	119346.88	1
9	15792365	He	501	France	Male	44	4	142051.07	2	0	1	74940.5	0
10	15592389	H?	684	France	Male	27	2	134603.88	1	1	1	71725.73	0
11	15767821	Bearce	528	France	Male	31	6	102016.72	2	0	0	80181.12	
12	15737173	Andrews	497	Spain	Male	24	3	0	2	1	0	76390.01	0
13	15632264	Kay	476	France	Female	34	10	0	2	1	0	26260.98	
14	15691483	Chin	549	France	Female	25	5	0	2	0	0	190857.79	
15	15600882	Scott	635	Spain	Female	35	7	0	2	1	1	6595	
16	15643966	Goforth	616	Germany	Male	45	3	143129.41	2	0	1		
17	15678…	Romeo	653	Germany	Male	58	1	132602.88	1	1	0		
18	15788218	Henderson	549	Spain	Female	24	9	0	2	1			
19	15661507	Muldrow	587	Spain	Male	45		0	1	0			
20	15568982	Hao	726	France	Female	24	6		2				
21	15577657	McDonald	732	France	Male	41	8	0	2				
22	15597945	Dellucci	636	Spain	Female	32	8	0	2				
23	15699309	Gerasimov	510	Spain	Female	38	4	0					
24	15725737	Mosman	669	France	Male	46	3	0					
25	15625047	Yen	846	France	Female	38	5	0					
26	15738191	Maclean	577	France	Male	25	3	0					
27	15736816	Young	756	Germany	Male	36	2	136815.64					
28	15700772	Nebechi	571	France	Male	44	9	0					
29	15728693	McWilliams	574	Germany	Female	43	3	141349.43					
30	15656300	Lucciano	411	France	Male	29	0	59697.17					
31	15589475	Azikiwe	591	Spain	Fe			0					
32	15706552	linakachukwu	533	France	Male	36	7	85311.7					
33	15750181	Sanderson	553	Germany	Male	41	9	110112.54					
34	15659428	Maggard	520	Spain	Female	42	6	0					
35	15732963	Clements	722	Spain	Female	29	9	0					
36	15794171	Lombardo	475	France	Female	45	0	134264.04					
37	15788448	Watson	490	Spain	Male	31							
38	15729599	Lorenzo	804	Spain	Male	33		.6					
39	15717426	Armstrong	850	France	Male	36	7	0	1				
40	15585768	Cameron	582	Germany	Male	41	6	70349.48	2				
41	15619360	Hsiao	472	Spain	Male	40	4	0	1				
42	15738148	Clarke	465	France	Female	51	8	122522.32	1	0			
43	15687946	Osborne	556	France	Female	61	2	117419.35	1	1			
44	15755196	Lavine	834	France	Female	49	2	131394.56	1	0			
45	15684171	Bianchi	660	Spain	Female	61	5	155931.11	1		1		
46	15754849	Tyler	776	Germany	Female	32	4	109421.13			1	1265	
47	15602280	Martin	829	Germany	Female	27	9	112045.67	1	1	1	119708.21	
48	15771573	Okagbue	637	Germany	Female	39	9	137	1	1	1	117622.8	
49	15766205	Yin	550	Germany	Male	38	2	103	1	0	1	90878.13	0
50	15771873	Buccho	776	Germany	Female	37	2	103769.22	2	1	0	194099	0
51	15616550	Chidiebele	698	Germany	Male	44	10	116363.37	2	1	0	19805	0
52	15768193	Trevisani	585	Germany	Male	36	5	146050.97	2	0	0		0
53	15683553	O'Brien	788	France	Female	33	5	0	2	0	0	11078.19	0
54	15702298	Parkhill	655	Germany	Male	41	8	125561.97	1	0	0	164040.94	1
55	15569590	Yoo	601	Germany	Male	42	1	98495.72	1	1	0	40014.76	1
56	15760861	Phillipps	619	France	Male	43	1	125211.92	1	1	1	113410.49	0
57	15630053	Tsao	656	France	Male	45	5	127864.4	1	1	0	87107.57	0



EXPLORE THE DATA SET

Load

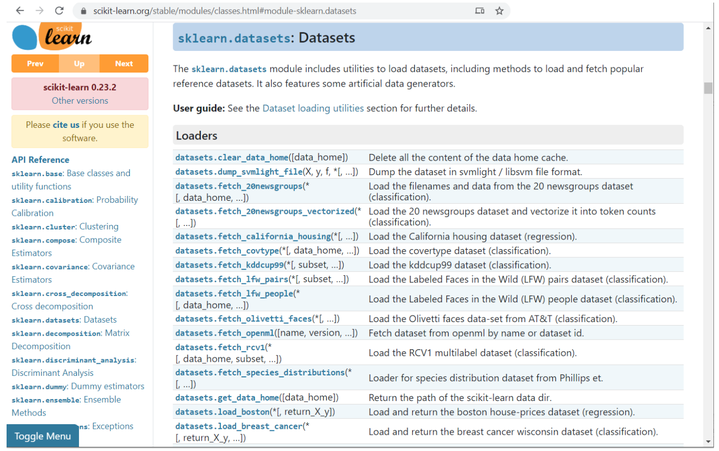
- Load a data set from a local file:

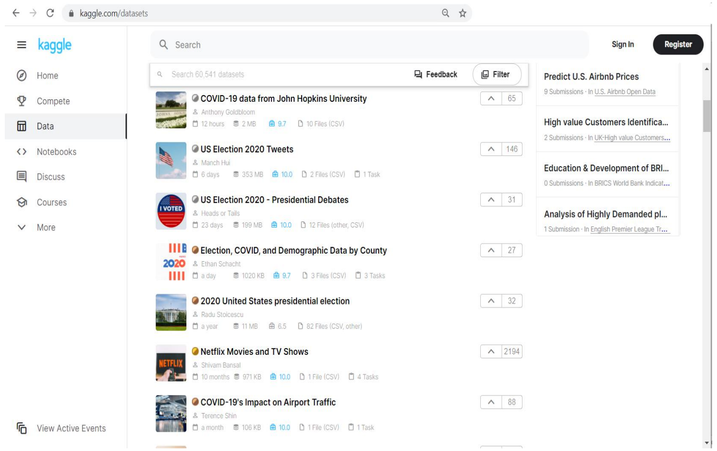
```
dataset = pd.read_csv('iris.csv')
```

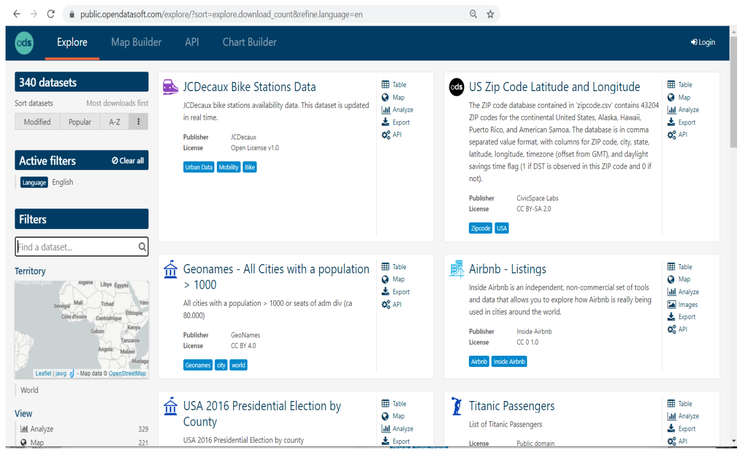
- Load a "reference" data set from a library:

```
dataset = sklearn.datasets.load_iris()
```

- Several bases of data sets:







EXPLORE THE DATA SET

Load

Predictor/
independent variable

Response/
dependent variable

	sepal_length	sepal_width	petal_length	petal_width	species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
11	5.4	3.7	1.5	0.2	setosa
12	4.8	3.4	1.6	0.2	setosa
13	4.8	3.0	1.4	0.1	setosa
14	4.3	3.0	1.1	0.1	setosa
15	5.8	4.0	1.2	0.2	setosa
16	5.7	4.4	1.5	0.4	setosa
17	5.4	3.9	1.3	0.4	setosa
18	5.1	3.5	1.4	0.3	setosa
19	5.7	3.8	1.7	0.3	setosa

EXPLORE THE DATA SET

Get general information

- Information on format:

```
dataset.shape
dataset.columns.values
dataset.info()
dataset.dtypes
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

- Information on content:

```
dataset.head()
dataset.tail()
dataset.sample(n=7)
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

EXPLORE THE DATA SET

Basic operations

- Conversion:

```
pd.DataFrame(data=dataset, index=…, columns=…)  # array to dataframe
df.to_numpy()    # dataframe to array
```

- Selecting:

```
df['feature']  # column
df[0:3]        # rows
df[df['feature']>0]  # rows that satisfy a condition
```

- Manipulation:

```
df_copy = df.copy()
df['new_feature'] = ['high' if x>10 else 'low' for x in df['feature']]
df['feature'].values.reshape(-1,1)
```

EXPLORE THE DATA SET

Compute basic statistics

- Count values:

```
dataset.isnull().sum()
dataset.isna().sum()
dataset.duplicated().sum()
dataset.nunique()
(dataset['species']=='setosa').sum()
```

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
```

- Mean, min/max, standard deviation, etc.:

```
dataset.describe()
```

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

EXPLORE THE DATA SET

Data frame vs array

Data frame (pandas)

```
dataset
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

```
type(dataset)
```

```
pandas.core.frame.DataFrame
```

```
dataset['sepal_width'][0]
```

```
3.5
```

```
dataset.iloc[0]
dataset.iloc[0,:]
```

```
sepal_length       5.1
sepal_width        3.5
petal_length       1.4
petal_width        0.2
species         setosa
Name: 0, dtype: object
```

```
dataset.iloc[0][1]
```

```
3.5
```

Array (numpy)

```
dataset.to_numpy()
dataset.values
```

```
array([[5.1, 3.5, 1.4, 0.2, 'setosa'],
       [4.9, 3.0, 1.4, 0.2, 'setosa'],
       [4.7, 3.2, 1.3, 0.2, 'setosa'],
       [4.6, 3.1, 1.5, 0.2, 'setosa'],
       [5.0, 3.6, 1.4, 0.2, 'setosa'],
       [5.4, 3.9, 1.7, 0.4, 'setosa'],
       [4.6, 3.4, 1.4, 0.3, 'setosa'],
       [5.0, 3.4, 1.5, 0.2, 'setosa'],
       [4.4, 2.9, 1.4, 0.2, 'setosa'],
       [4.9, 3.1, 1.5, 0.1, 'setosa'],
       [5.4, 3.7, 1.5, 0.2, 'setosa'],
       [4.8, 3.4, 1.6, 0.2, 'setosa'],
       [4.8, 3.0, 1.4, 0.1, 'setosa'],
       [4.3, 3.0, 1.1, 0.1, 'setosa'],
       [5.8, 4.0, 1.2, 0.2, 'setosa'],
       [5.7, 4.4, 1.5, 0.4, 'setosa'],
```

```
type(dataset.values)
```

```
numpy.ndarray
```

```
dataset.values[0][1]
dataset.values[0,1]
```

```
3.5
```

```
type(dataset.values[0,1])
```

```
float
```

```
dataset.values[0,4]
```

```
'setosa'
```

```
type(dataset.values[0,4])
```

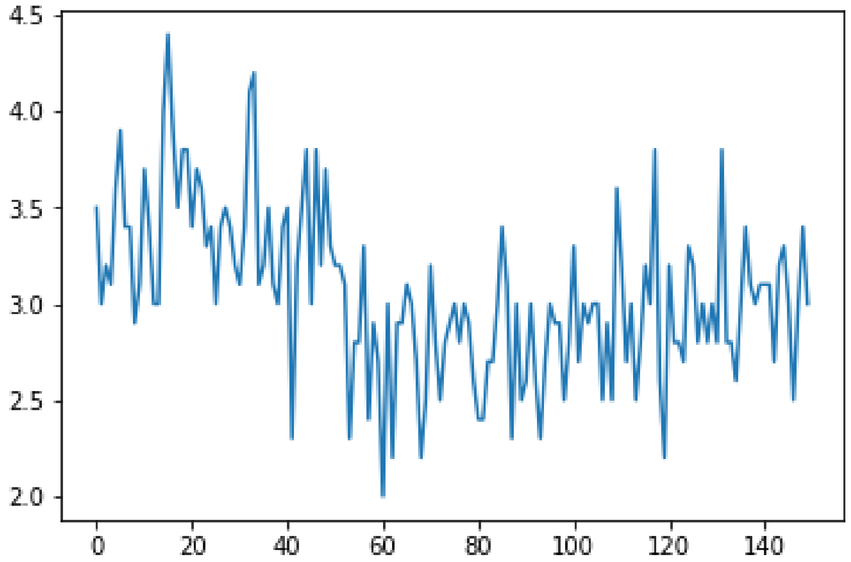
```
str
```

EXPLORE THE DATA SET

Visualize

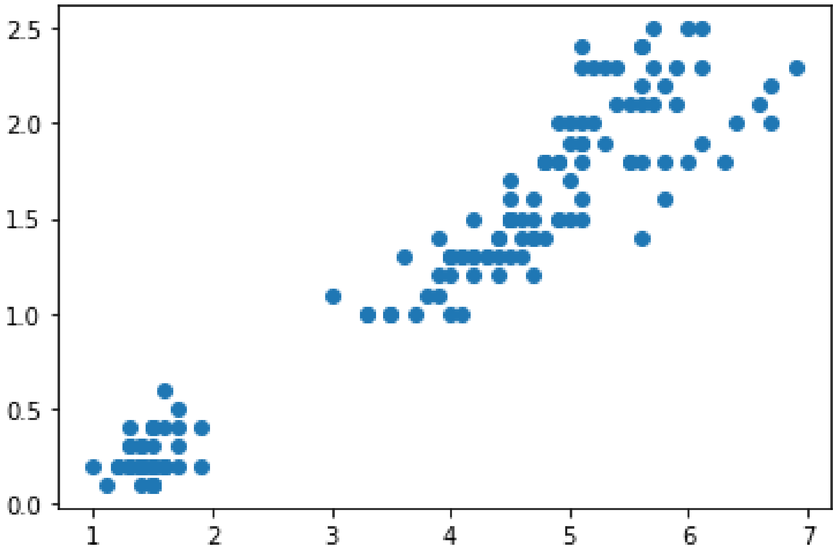
- Curve plot:

```
plt.plot(dataset['sepal_width'])
plt.show()
```



- Scatter plot:

```
plt.scatter(dataset['petal_length'],
   dataset['petal_width'])
```

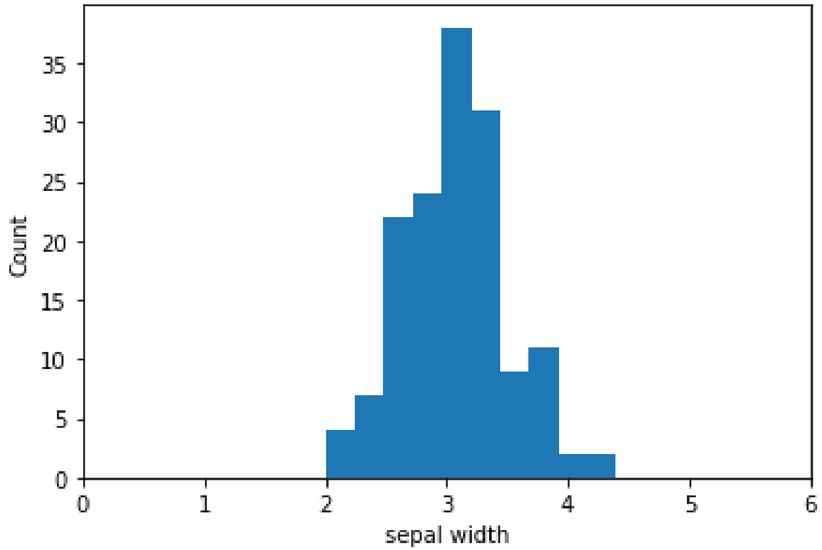


EXPLORE THE DATA SET

Visualize

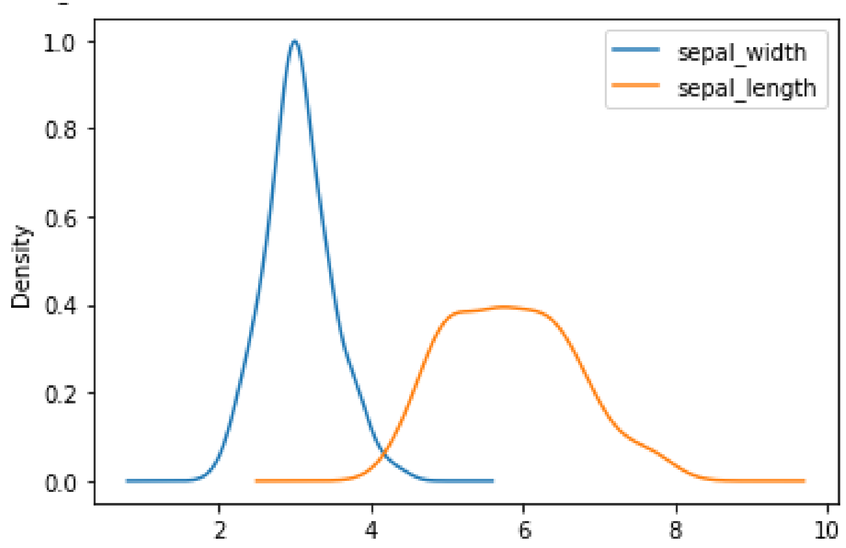
- Histogram:

```
plt.hist(dataset['sepal_width'])
```



- Density plot:

```
dataset[['sepal_width',
   'sepal_length']].plot(kind='density')
```

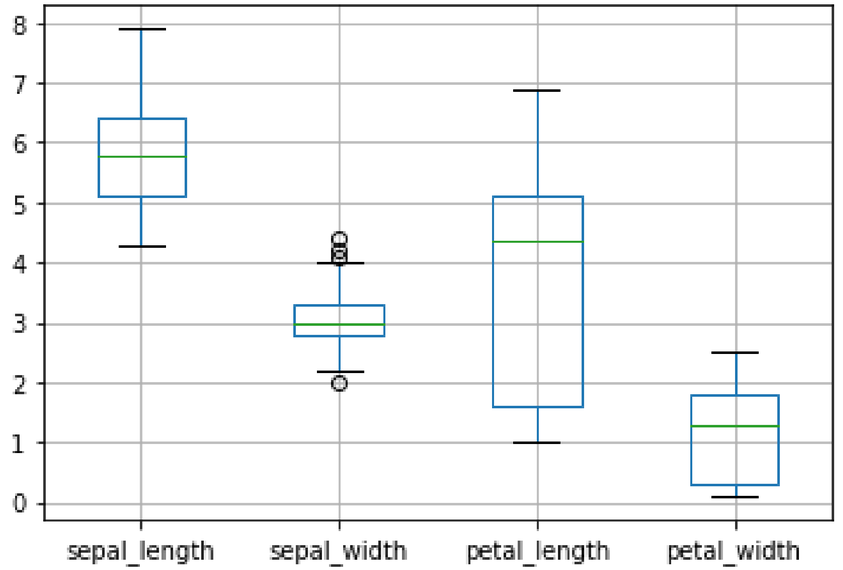


EXPLORE THE DATA SET

Visualize

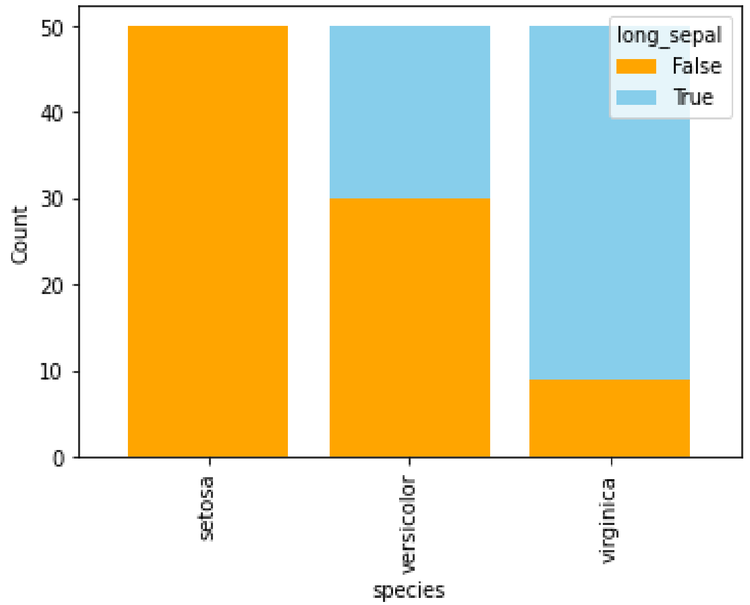
- Boxplot:

```
dataset[['sepal_length', 'sepal_width',
   'petal_length', 'petal_width']].boxplot()
```



- Stacked bars:

```
df_plot = dataset.groupby(['species',
   'long_sepal']).size().reset_index().pivot(index=
   'species', columns='long_sepal', values=0)
df_plot.plot(kind='bar', stacked=True, color=
   ['orange', 'skyblue'], width=0.8)
```

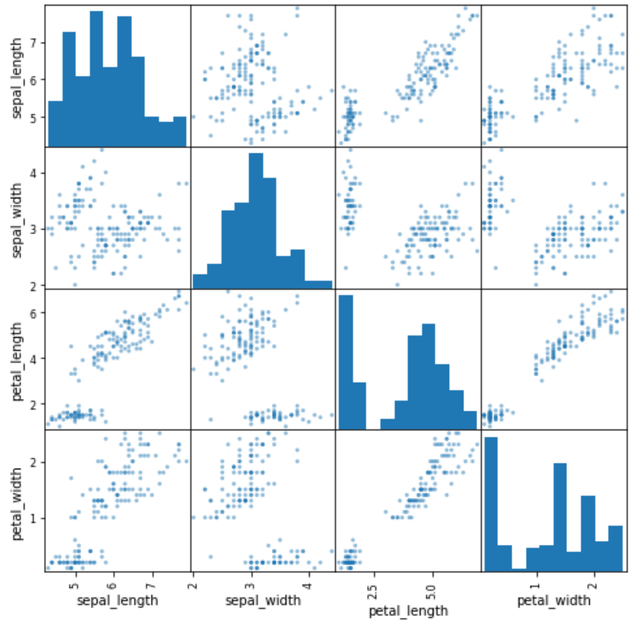


EXPLORE THE DATA SET

Visualize

- Scatter matrix:

```
scatter_matrix(dataset.iloc[:,:4])
```

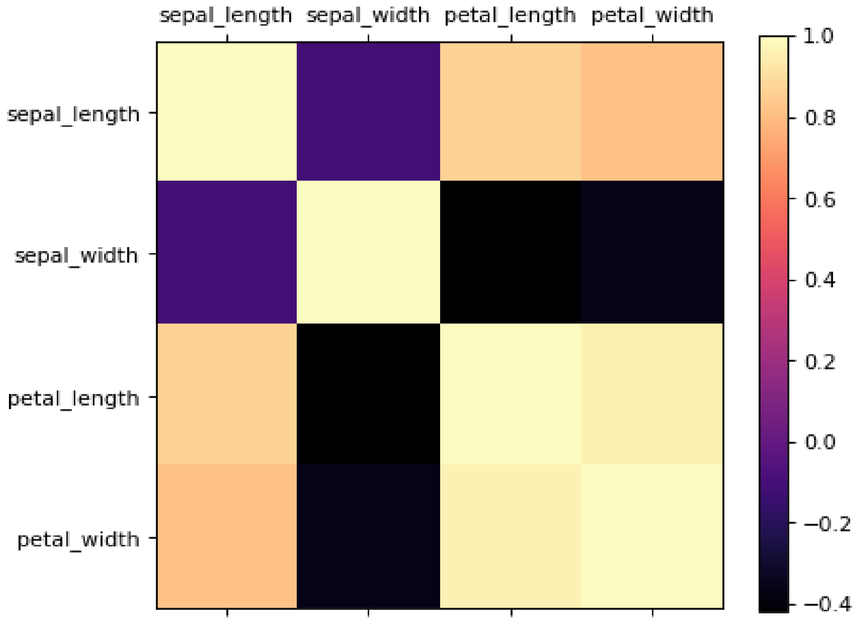


- Correlation matrix:

```
correlation = dataset.iloc[:,:4].corr()
matshow(correlation)
```

alternative

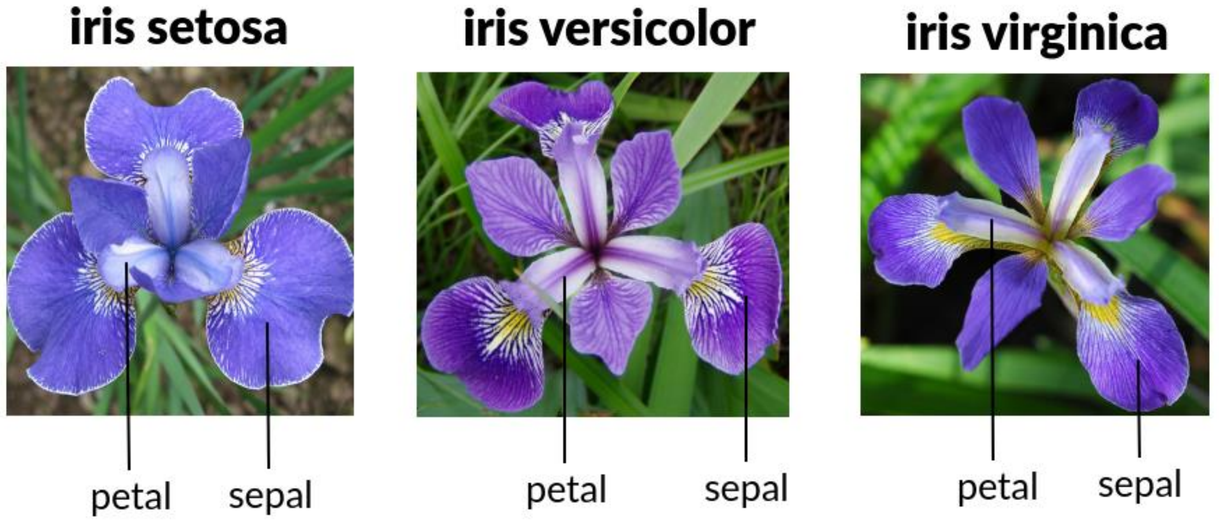
```
import seaborn as sns
sns.heatmap(flights, annot=True, fmt="d")
```



EXPLORE THE DATA SET

Implementation

- Iris data set: characteristics of three species of iris.
- Objectives:
  - Explore the data
  - Work with Jupyter notebook



EXPLORE THE DATA SET

Practice

- Titanic data set: “what sorts of people were more likely to survive?”
- Guidelines:
  - Data set is provided
  - Work with Jupyter notebook
  - Start during the course, finish at home
  - Report should be submitted before Wednesday 6 p.m. on Teams

EXPLORE THE DATA SET

Practice

Some advices:

- Submit an executed Jupyter notebook
- Don’t hesitate to check the documentation of functions
- Don’t do copy-paste from others’ work (of from past correction)
- Always write code that you understand
- Provide written explanations (not only numbers) when an interpretation is needed

COURSE PROGRAM

Structure

PREPARATION
- Data exploration
- Data preprocessing

REGRESSIONS
- Linear regression with one variable
- Multiple and polynomial regressions
- Regression model assessment

CLASSIFICATIONS
- Logistic regression
- Classification model assessment
- Discriminant Analysis
- k-NN, Decision Tree, SVM

CLUSTERING
- k-means, hierarchical clustering

DIMENSIONALITY REDUCTION
- Principal Components Analysis
- Extensions

ALL NOTIONS
- Final assignment

PREPARE THE DATA SET

Why?

- Remove unrelevant data
- "Repair" incorrect/missing data
- Set to numeric format
- Split data: predictors/response, train/test

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
1	0	3	Braund, Mr. Owen Harris	male		1	0	A/5 21171	7.25
2	1	1	n Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	2. 3101282	7.925
4	1	1	lrs. Jacques Heath (Lily May Peel)						
5	0	3							
6	0	3	Moran, Mr. James						
7	0		McCarthy, Mr. Timothy J						
8	0	3	Palsson, Master. Gosta Leonard						
9	1	3	scar W (Elisabeth Vilhelmina Berg)						
10	1	2	sser, Mrs. Nicholas (Adele Achem)						
11	1		Sandstrom, Miss. Marguerite Rut						
12	1	1	Bonnell, Miss. Elizabeth						
13	0	3	Saundercock, Mr. William Henry						
14	0	3	Andersson, Mr. Anders Johan						
15		3	rom, Miss. Hulda Amanda Adolfina						
16	1	2	Hewlett, Mrs. (Mary D Kingcome)						
17		3							
18	1	2	Williams, Mr. Charles Eugene						
19	0	3	lius (Emelia Maria Vandemoortele)						
20		3	Masselmani, Mrs. Fatima						
21	0	2	Fynney, Mr. Joseph J						
22	1		Beesley, Mr. Lawrence						
23		3	McGowan, Miss. Anna "Annie"						
24	1	1							
25	0	3	Palsson, Miss. Torborg Danira						
26	1	3	Selma Augusta Emilia Johansson)						
27	0	3	Emir, Mr. Farred Chehab						
28	0	1	Fortune, Mr. Charles Alexander						
29		3	O'Dwyer, Miss. Ellen "Nellie"						
30		3							
31	0	1	Uruchurtu, Don. Manuel E	male	40	0	0	PC 17601	27.7208
32	1	1	William Augustus (Marie Eugenie)	female	57	1	0	PC 17569	146.5208

	X					y
	Embarked	Pclass	Fare	Sex	Age	Survived
training	2	3	7.25	0	22.0	0
	0	1	71.2833	1	38.0	1
	2	3	7.925	1	26.0	1
	2	1	53.1	1	35.0	1
	2	3	8.05	0	35.0	0
	1	3	8.4583	0	21.0	0
	2	1	51.8625	0	54.0	0
	2	3	21.075	0	2.0	0
	2	3	11.1333	1	27.0	1
	0	2	30.0708	1	14.0	1
	2	3	16.7	1	4.0	1
	2	1	26.55	1	58.0	1
	2	3	8.05	0	20.0	0
	2	3	31.275	0	39.0	0
	2	3	7.8542	1	14.0	0
testing	1	3	7.8792	1	35.0	1
	2	3	7.8958	0	25.0	0
	0	1	27.7208	0	40.0	0
	0	1	146.5208	1	57.0	0

PREPARE THE DATA SET

Removing unrelevant data

- Removing a column

```
dataset.drop(['PassengerId'], axis=1)
```

- Removing rows

```
dataset.drop_duplicates()
dataset.drop(dataset[dataset['Fare'] <= 0].index,
   axis=0)
```

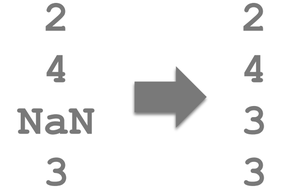
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp
1	1	0	3	Braund, Mr. Owen Harris	male	22	1
2	2	1	1	n Bradley (Florence Briggs Thayer)	female	38	1
3	3	1	3	Heikkinen, Miss. Laina	female	26	0
4	4	1	1	rs. Jacques Heath (Lily May Peel)	female	35	1
5	5	0	3	Allen, Mr. William Henry	male	35	0
6	6	0	3	Moran, Mr. James	male	21	0
7	7	0	1	McCarthy, Mr. Timothy J	male	54	0
8	8	0	3	Palsson, Master. Gosta Leonard	male	2	3
9	9	1	3	scar W (Elisabeth Vilhelmina Berg)	female	27	0
10	10	1	2	sser, Mrs. Nicholas (Adele Achem)	female	14	1
11	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4	1
12	12	1	1	Bonnell, Miss. Elizabeth	female	58	0
13	13	0	3	Saundercock, Mr. William Henry	male	20	0
14	14	0	3	Andersson, Mr. Anders Johan	male	39	1
15	15	0	3	rom, Miss. Hulda Amanda Adolfina	female	14	0
16	16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55	0
17	17	0	3	Rice, Master. Eugene	male	2	4
18	18	1	2	Williams, Mr. Charles Eugene	male	32	0
19	19	0	3	lius (Emelia Maria Vandemoortele)	female	31	1
20	20	1	3	Masselmani, Mrs. Fatima	female	45	0
21	21	0	2	Fynney, Mr. Joseph J	male	35	0
22	22	1	2	Beesley, Mr. Lawrence	male	34	0
23	23	1	3	McGowan, Miss. Anna "Annie"	female	15	0
24	24	1	1	Sloper, Mr. William Thompson	male	28	0
25	25	0	3	Palsson, Miss. Torborg Danira	female	8	3
26	26	1	3	Selma Augusta Emilia Johansson)	female	38	1
27	27	0	3	Emir, Mr. Farred Chehab	male	28	0
28	28	0	1	Fortune, Mr. Charles Alexander	male	19	3
29	29	1	3	O'Dwyer, Miss. Ellen "Nellie"	female	35	0
30	30	0	3	Todoroff, Mr. Lalio	male	25	0
31	31	0	1	Uruchurtu, Don. Manuel E	male	40	0
32	32	1	1	William Augustus (Marie Eugenie)	female	57	1

PREPARE THE DATA SET

Filling missing data

- Replace the missing values:

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
imputer.fit(X[:, 1:3])
X[:, 1:3] = imputer.transform(X[:, 1:3])
```




```
imputer2 = SimpleImputer(missing_values='error', strategy='constant’,
   fill_value=0)
```

PREPARE THE DATA SET

Preparing features and response sets

- Identification of the predictors

```
X = dataset.iloc[:, :-1]
```

- Identification of the response (case of supervised learning)

```
y = dataset.iloc[:, -1]
```

X

Embarked	Pclass	Fare	Sex	Age
2	3	7.25	0	22.0
0	1	71.2833	1	38.0
2	3	7.925	1	26.0
2	1	53.1	1	35.0
2	3	8.05	0	35.0
1	3	8.4583	0	21.0
2	1	51.8625	0	54.0
2	3	21.075	0	2.0
2	3	11.1333	1	27.0
0	2	30.0708	1	14.0
2	3	16.7	1	4.0
2	1	26.55	1	58.0
2	3	8.05	0	20.0
2	3	31.275	0	39.0
2	3	7.8542	1	14.0

y

Survived
0
1
1
1
0
0
0
0
1
1
1
1
0
0
0

PREPARE THE DATA SET

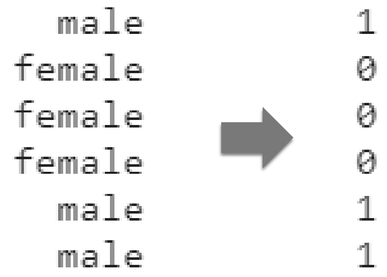
Encoding data

- Boolean data:
  - Explicit encoding:

```
encoding_dict = {'female':1, 'male': 0}
X['Sex'] = X['Sex'].replace(encoding_dict)
```

  - Implicit encoding:

```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
X['Sex'] = label_encoder.fit_transform(X['Sex'])
```



PREPARE THE DATA SET

Encoding data

- Categorical data:
  - Explicit encoding (suitable for an ordered set only):

```
encoding_dict = {'bad':0, 'middle': 1, 'good': 2}
X['Level'] = X['Level'].replace(encoding_dict)
```

  - Implicit encoding (suitable for distinct categories):

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
oh_encoder = ColumnTransformer(transformers=[('encoder’,
  OneHotEncoder(), ['Embarked'])], remainder='passthrough')
X = oh_encoder.fit_transform(X)
```

alternative

```
pd.get_dummies(X, columns=['Embarked'],prefix=['is'])
```

Dummy variables

```
S        0.  0.  1.
S        0.  0.  1.
C   ➡    1.  0.  0.
S        0.  0.  1.
S        0.  0.  1.
S        0.  0.  1.
Q        0.  1.  0.
```

PREPARE THE DATA SET

Splitting the data set

- Split the data set so to have data dedicated to training a model and data dedicated to testing this model:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
   test_size = 0.25, random_state = 1)
```

Embarked	Pclass	Fare	Sex	Age	Survived
2	3	7.25	0	22.0	0
0	1	71.2833	1	38.0	1
2	3	7.925	1	26.0	1
2	1	53.1	1	35.0	1
2	3	8.05	0	35.0	0
1	3	8.4583	0	21.0	0
2	1	51.8625	0	54.0	0
2	3	21.075	0	2.0	0
2	3	11.1333	1	27.0	1
0	2	30.0708	1	14.0	1
2	3	16.7	1	4.0	1
2	1	26.55	1	58.0	1
2	3	8.05	0	20.0	0
2	3	31.275	0	39.0	0
2	3	7.8542	1	14.0	0

1	3	7.8792	1	35.0	1
2	3	7.8958	0	25.0	0
0	1	27.7208	0	40.0	0
0	1	146.5208	1	57.0	0

PREPARE THE DATA SET

Scaling data

Standardisation	Normalisation
$x_{stand} = \frac{x - \text{mean}(x)}{\text{standard deviation }(x)}$	$x_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)}$

- Standardisation:

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train[:, 5:])
X_train[:, 5:] = scaler.transform(X_train[:, 5:])
```

- Normalisation:

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range=(0,1))
data_rescaled = scaler.fit_transform(data)
```

16.	-0.93
22.	-0.52
31.	0.08
50.	1.36
53.	1.56
38.	0.55
17.	-0.86
40.	0.69
20.	-0.66
47.	1.16
48.	1.23
26.	-0.25
26.	-0.25
40.	0.69
31.	0.08
20.5	-0.62
19.	-0.72
31.	0.08
24.	-0.39
32.5	0.18

PREPARE THE DATA SET

Implementation

- Titanic data set: information on passengers (age, class, survived, etc.)
- Objectives:
  - Make it ready for ML processing

PREPARE THE DATA SET

Practice

- Bank churn: list of customers of a loan bank.
- Objectives:
  - Explore the data
  - Prepare the data set so that it can be directly used for ML processing for churn prediction
- Deadline: Tuesday 29th 6 p.m.

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0
5	6	15574012	Chu	645	Spain	Male	44	8	113755.78	2	1	0	149756.71	1
6	7	15592531	Bartlett	822	France	Male	50	7	0.00	2	1	1	10062.80	0
7	8	15656148	Obinna	376	Germany	Female	29	4	115046.74	4	1	0	119346.88	1
8	9	15792365	He	501	France	Male	44	4	142051.07	2	0	1	74940.50	0
9	10	15592389	H?	684	France	Male	27	2	134603.88	1	1	1	71725.73	0
10	11	15767821	Bearce	528	France	Male	31	6	102016.72	2	0	0	80181.12	0
11	12	15737173	Andrews	497	Spain	Male	24	3	0.00	2	1	0	76390.01	0
12	13	15632264	Kay	476	France	Female	34	10	0.00	2	1	0	26260.98	0
13	14	15691483	Chin	549	France	Female	25	5	0.00	2	0	0	190857.79	0
14	15	15600882	Scott	635	Spain	Female	35	7	0.00	2	1	1	65951.65	0
15	16	15643966	Goforth	616	Germany	Male	45	3	143129.41	2	0	1	64327.26	0
16	17	15737452	Romeo	653	Germany	Male	58	1	132602.88	1	1	0	5097.67	1
17	18	15788218	Henderson	549	Spain	Female	24	9	0.00	2	1	1	14406.41	0
18	19	15661507	Muldrow	587	Spain	Male	45	6	0.00	1	0	0	158684.81	0
19	20	15568982	Hao	726	France	Female	24	6	0.00	2	1	1	54724.03	0

COURSE PROGRAM

Structure

PREPARATION
- Data exploration
- Data preprocessing

REGRESSIONS
- Linear regression with one variable
- Multiple and polynomial regressions
- Regression model assessment

CLASSIFICATIONS
- Logistic regression
- Classification model assessment
- Discriminant Analysis
- k-NN, Decision Tree, SVM

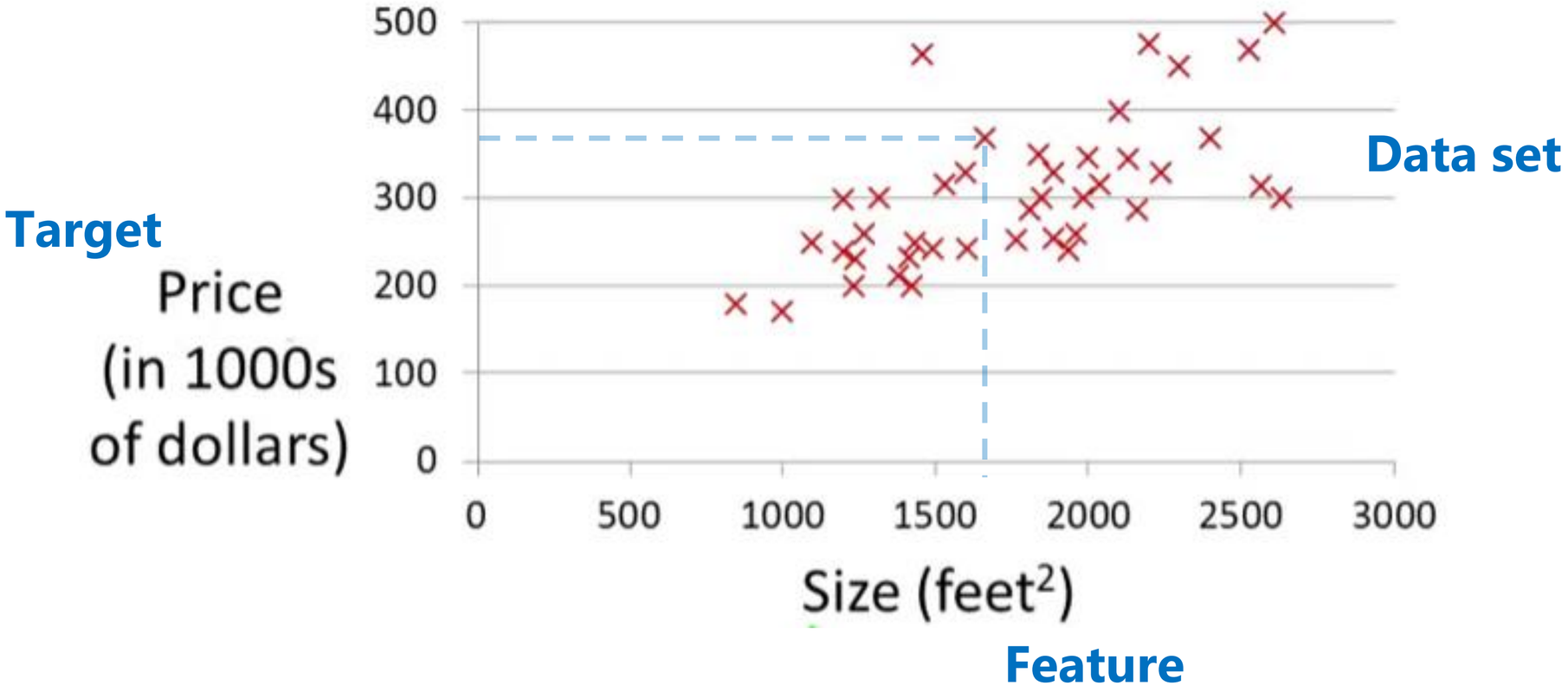
CLUSTERING
- k-means, hierarchical clustering

DIMENSIONALITY REDUCTION
- Principal Components Analysis
- Extensions

ALL NOTIONS
- Final assignment

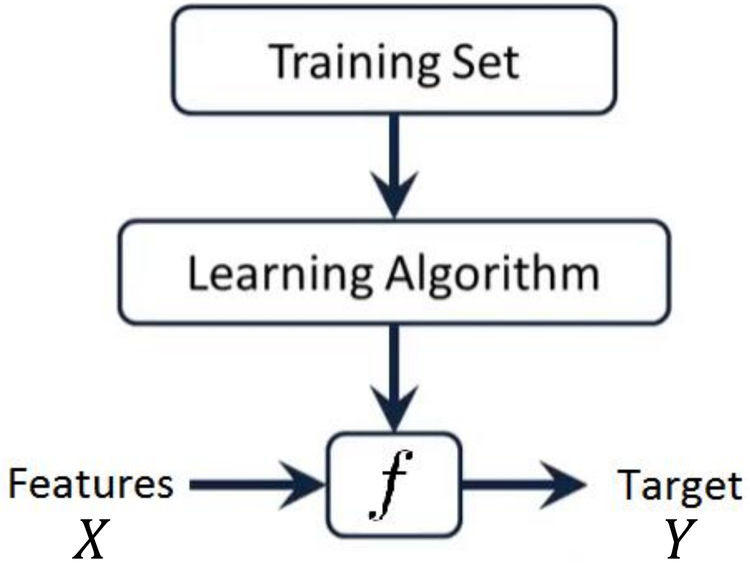
GENERAL APPROACH FOR ML

Problem statement



GENERAL APPROACH FOR ML

Problem statement



- Why estimate f?
  - **Prediction**: provide a response estimation for any feature values ($\hat{Y} = \hat{f}(X)$).
  - **Inference**: understand the relationship between X and Y, i.e. how Y changes as function of X_1, X_2, ..., X_p.

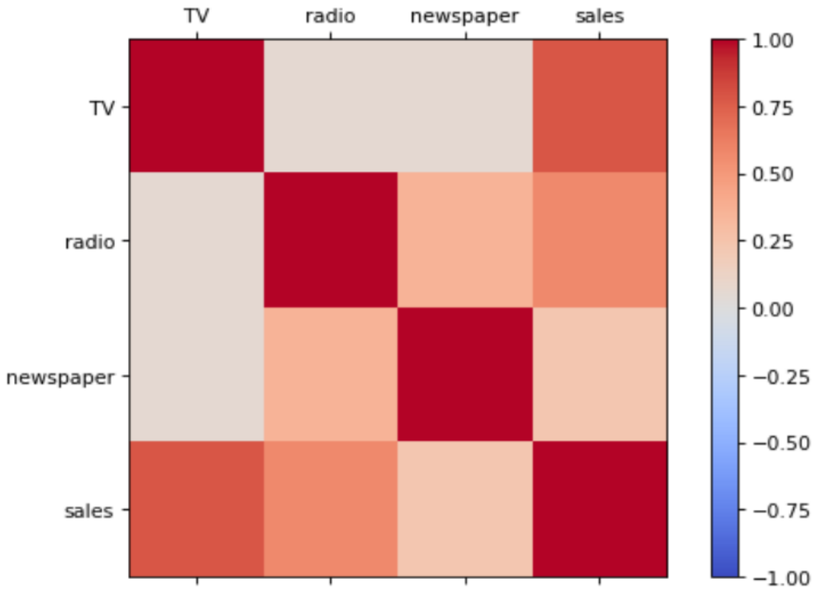
GENERAL APPROACH FOR ML

Solving process

1. Get some intuition from **data inspection** (visualization, correlation, etc.)
2. **Choose a model**
3. **Find the parameters** that minimize a criteria (cost function)
4. **Evaluate the performance**

REGRESSIONS

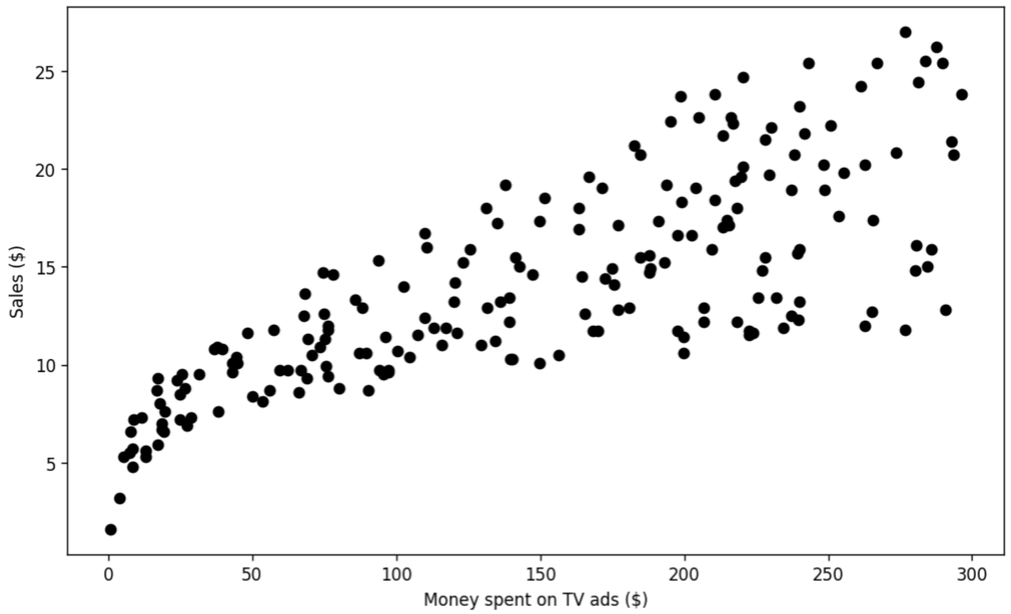
Data inspection



Correlation matrix

Target

Continuous values



Feature

REGRESSIONS

Model choice

- Linear regression (one single predictor, or more)
- Polynomial regression
- Non-linear regression model
- Data transformation

REGRESSIONS

Model parameters estimation

- **Analytical solving** through Ordinary Least Squares (linear algebra operations)
- **Numerical solving** through an optimization algorithm, e.g. Gradient Descent

REGRESSIONS

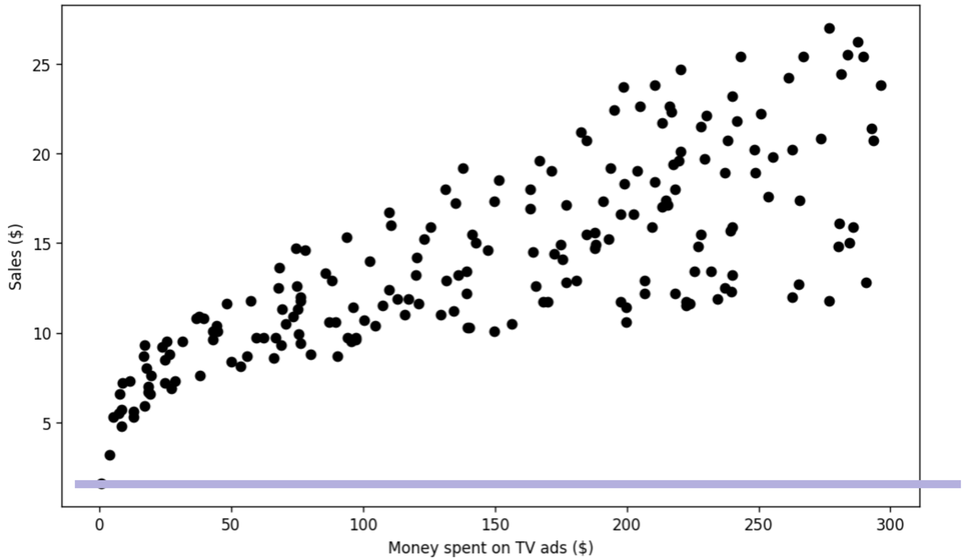
Performance evaluation

- RSS, MSE, MAE, …
- R^2 score (ANOVA)
- Hypothesis testing (p-values)
- F-statistics

SIMPLE LINEAR REGRESSION

Model definition

- Inspection:



- Assumption of a linear relationship between the response Y and a single predictor variable X:

$$Y = \beta_0 + \beta_1 X + e$$

Intercept **Slope** **Error term**

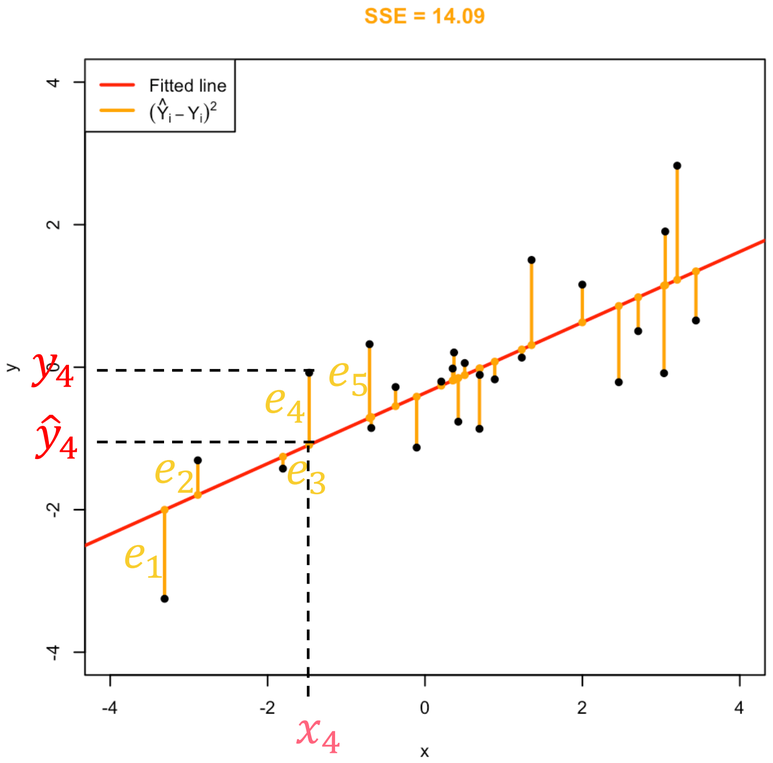
SIMPLE LINEAR REGRESSION

Model fitting

- Cost function: Residual Sum of Square (RSS or SSE)

$$\text{RSS} = \sum_{i=1}^{n} e_i^2 \quad \text{with } e_i \text{ the } i^{\text{th}} \text{ residual: } e_i = y_i - \hat{y}_i$$

$$\text{RSS} = \sum_{i=1}^{n} \left(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i\right)^2$$



- Minimization through Least Squares approach:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})\,(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{s_{xy}}{s_x^2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

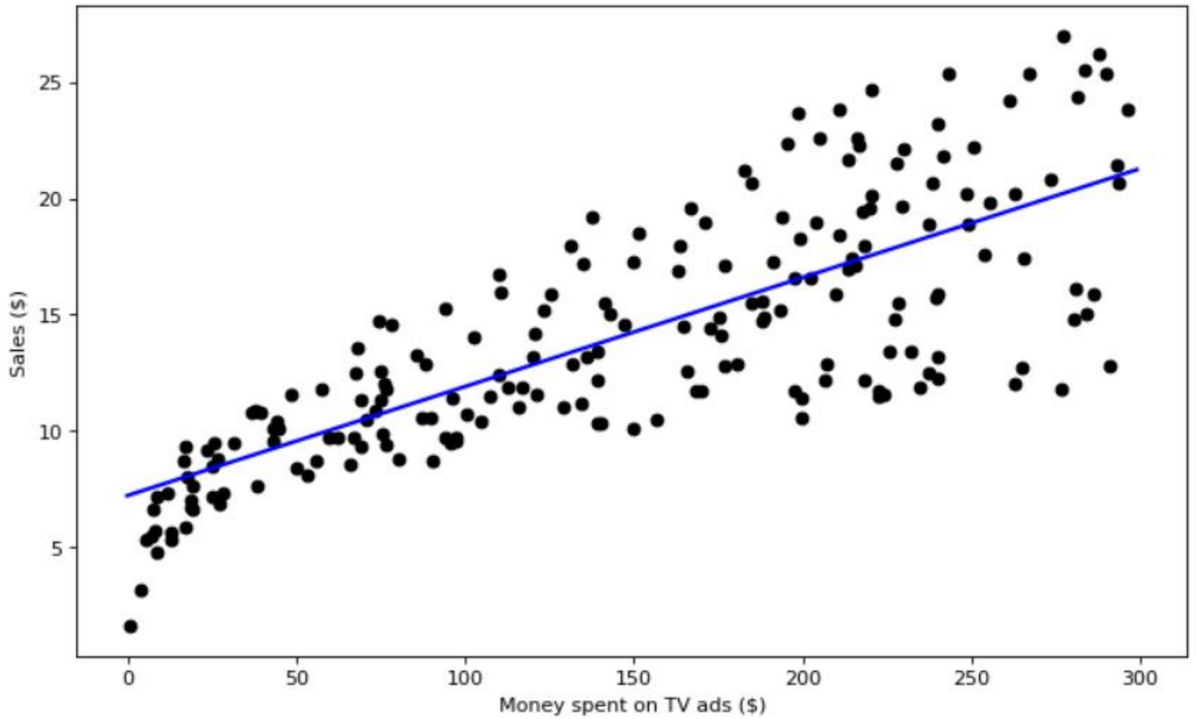
- Sample mean: $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$
- Sample variance: $s_x^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$
- Sample covariance: $s_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$

SIMPLE LINEAR REGRESSION

Model fitting

- Prediction:

$$\hat{y} = 7.22 + 0.047x$$



SIMPLE LINEAR REGRESSION

Model performance assessment

- Accuracy of the prediction:
  - Sum of Squared Errors or Residual Sum of Squares: $\mathrm{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$
  - Mean Absolute Error: $\mathrm{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$
  - Mean Squared Error: $\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$
  - Root Mean Squared Error: $\mathrm{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$

SIMPLE LINEAR REGRESSION

Model performance assessment

- Accuracy of the coefficient estimates:
  - The standard error of an estimator reflects how it varies under repeated sampling:

$$\mathrm{SE}(\hat{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \quad \text{with } \sigma^2 = \mathrm{Var}(e)$$

$$\mathrm{SE}(\hat{\beta}_0)^2 = \sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \right)$$

  - σ is the residual standard error and can be estimated as: $\mathrm{RSE} = \sqrt{\frac{\mathrm{RSS}}{n-2}}$
  - Confidence interval, e.g. range of values such that there is a 95 % probability that the true parameter value is within the interval:

$$[\hat{\beta}_1 - 2 \times \mathrm{SE}(\hat{\beta}_1), \hat{\beta}_1 + 2 \times \mathrm{SE}(\hat{\beta}_1)]$$

$$[\hat{\beta}_0 - 2 \times \mathrm{SE}(\hat{\beta}_0), \hat{\beta}_0 + 2 \times \mathrm{SE}(\hat{\beta}_0)]$$

SIMPLE LINEAR REGRESSION

Python implementation

- Training the linear regression model:

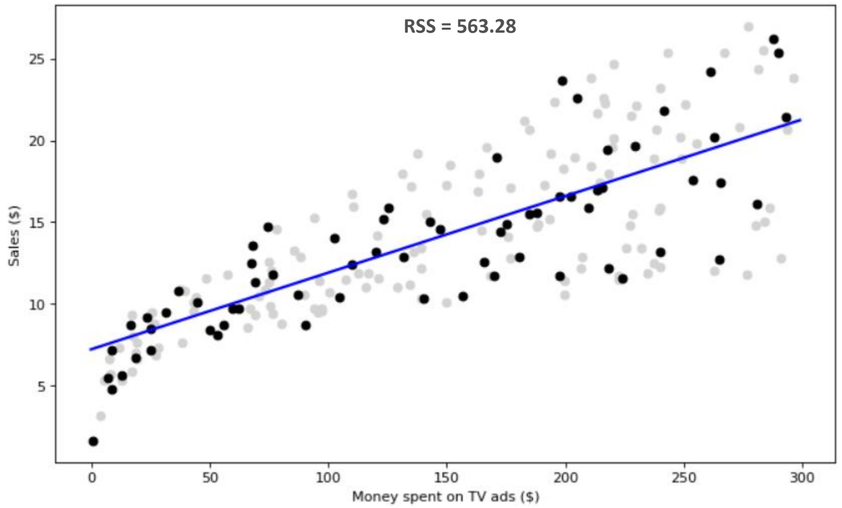
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

- Using the model for predicting:

```
y_pred = regressor.predict(X_test)
```

- Assessing the accuracy:

```
RSS = sum((y_pred - y_test) ** 2)
MAE = np.mean(abs(y_pred - y_test))
MSE = np.mean((y_pred - y_test) ** 2)
RMSE = np.sqrt(np.mean((y_pred - y_test) ** 2)))
```



SIMPLE LINEAR REGRESSION

Example of implementation

- Data set: product sales w.r.t. ads expenditures
- Objectives:
  - Inspect correlation between candidate features and the target
  - Train a simple linear regression with one feature
  - Evaluate the model accuracy

TV	radio	newspaper	sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	9.3
151.5	41.3	58.5	18.5
180.8	10.8	58.4	12.9
8.7	48.9	75	7.2
57.5	32.8	23.5	11.8
120.2	19.6	11.6	13.2
8.6	2.1	1	4.8

SIMPLE LINEAR REGRESSION

Practice

- Data set: CO_2 emission w.r.t. vehicle characteristics
- Objectives:
  - Check for possible correlations
  - Train simple linear regressions with one feature
  - Assess and compare the accuracy of the regressions
- Deadline: Sunday July 4th 6 p.m.

	YEAR	MAKE	MODEL	VEHICLECLASS	ENGINESIZE	CYLINDERS	MISSION	FUELTYPE	ON_CITY	_HWY	OMB	MPG	CO2EMISSIONS
1	2014	ACURA	ILX	COMPACT	2	4	AS5	Z	9.9	6.7	8.5	33	196
2	2014	ACURA	ILX	COMPACT	2.4	4	M6	Z	11.2	7.7	9.6	29	221
3	2014	ACURA	ILX HYBRID	COMPACT	1.5	4	AV7	Z	6	5.8	5.9	48	136
4	2014	ACURA	MDX 4WD	SUV - SMALL	3.5	6	AS6	Z	12.7	9.1	11.1	25	255
5	2014	ACURA	RDX AWD	SUV - SMALL	3.5	6	AS6	Z	12.1	8.7	10.6	27	244
6	2014	ACURA	RLX	MID-SIZE	3.5	6	AS6	Z	11.9	7.7	10	28	230
7	2014	ACURA	TL	MID-SIZE	3.5	6	AS6	Z	11.8	8.1	10.1	28	232
8	2014	ACURA	TL AWD	MID-SIZE	3.7	6	AS6	Z	12.8	9	11.1	25	255
9	2014	ACURA	TL AWD	MID-SIZE	3.7	6	M6	Z	13.4	9.5	11.6	24	267
10	2014	ACURA	TSX	COMPACT	2.4	4	AS5	Z	10.6	7.5	9.2	31	212
11	2014	ACURA	TSX	COMPACT	2.4	4	M6	Z	11.2	8.1	9.8	29	225
12	2014	ACURA	TSX	COMPACT	3.5	6	AS5	Z	12.1	8.3	10.4	27	239
13	2014	ASTON MARTIN	DB9	MINICOMPACT	5.9	12	A6	Z	18	12.6	15.6	18	359
14	2014	ASTON MARTIN	RAPIDE	SUBCOMPACT	5.9	12	A6	Z	18	12.6	15.6	18	359
15	2014	ASTON MARTIN	V8 VANTAGE	TWO-SEATER	4.7	8	AM7	Z	17.4	11.3	14.7	19	338
16	2014	ASTON MARTIN	V8 VANTAGE	TWO-SEATER	4.7	8	M6	Z	18.1	12.2	15.4	18	354
17	2014	ASTON MARTIN	V8 VANTAGE S	TWO-SEATER	4.7	8	AM7	Z	17.4	11.3	14.7	19	338

COURSE PROGRAM

Structure

PREPARATION
- Data exploration
- Data preprocessing

REGRESSIONS
- Linear regression with one variable
- Multiple and polynomial regressions
- Regression model assessment

CLASSIFICATIONS
- Logistic regression
- Classification model assessment
- Discriminant Analysis
- k-NN, Decision Tree, SVM

CLUSTERING
- k-means, hierarchical clustering

DIMENSIONALITY REDUCTION
- Principal Components Analysis
- Extensions

ALL NOTIONS
- Final assignment

MULTIPLE LINEAR REGRESSION

Model fitting

- We consider p distinct predictors: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + e$

- In matrix terms: $\mathbf{Y} = \mathbf{X}\beta + e$

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} \quad \mathbf{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \quad \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} \quad e = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}$$

MULTIPLE LINEAR REGRESSION

Model fitting

- We choose $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$ to minimize the residual sum of squares:

$$\text{RSS} = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \left(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2} - \cdots - \hat{\beta}_p x_{ip}\right)^2$$

- Linear combination of $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$ ➔ Least Squares (analytical solving):

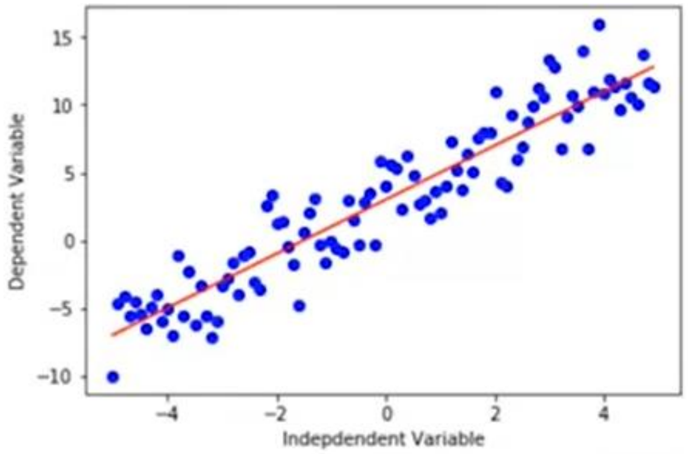
$$\hat{\beta} = \mathbf{X}^{+}\mathbf{Y} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

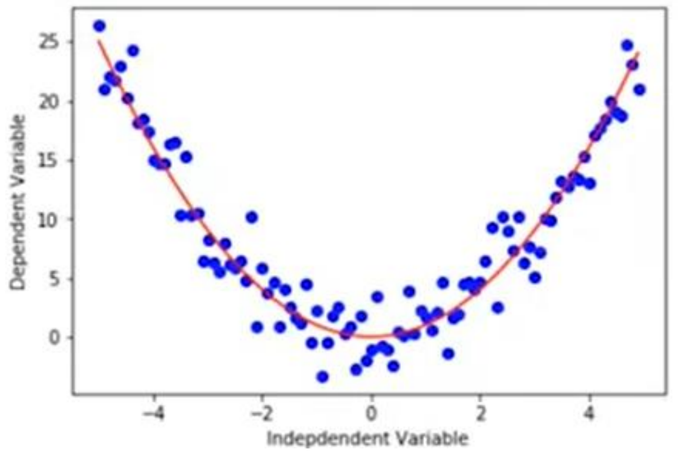
Pseudoinverse

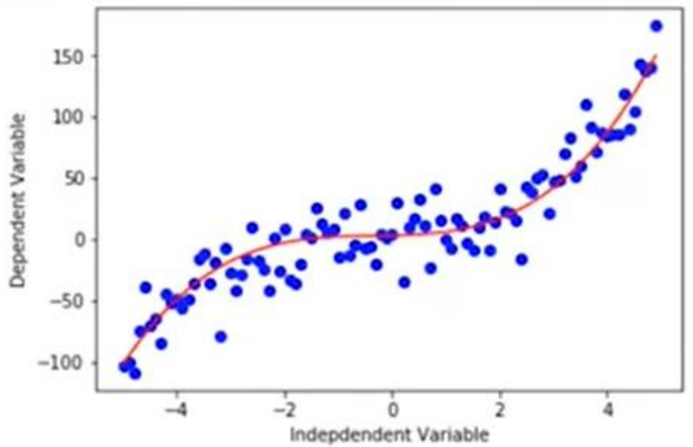
MULTIPLE LINEAR REGRESSION

Polynomial regression

- Some curvy data can be modeled by a polynomial regression:







- It can be transformed into a linear regression model. E.g.:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + e$$

$$\Rightarrow Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + e \quad \text{with } X_1 = X, X_2 = X^2, X_3 = X^3$$

MULTIPLE LINEAR REGRESSION

Linear model extensions

- Categorical variables. E.g.:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + e \quad \text{with } X_1 = \begin{cases} 0 \text{ if male} \\ 1 \text{ if female} \end{cases}$$

- Interaction terms. E.g.:

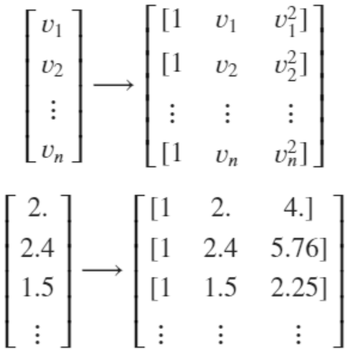
$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + e$$

MULTIPLE LINEAR REGRESSION

Python implementation

- Creating the polynomial features:

```
from sklearn.preprocessing import PolynomialFeatures
poly = PolynomialFeatures(degree=2)
X_poly = poly.fit_transform(X)
```



- Training the linear regression model:

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

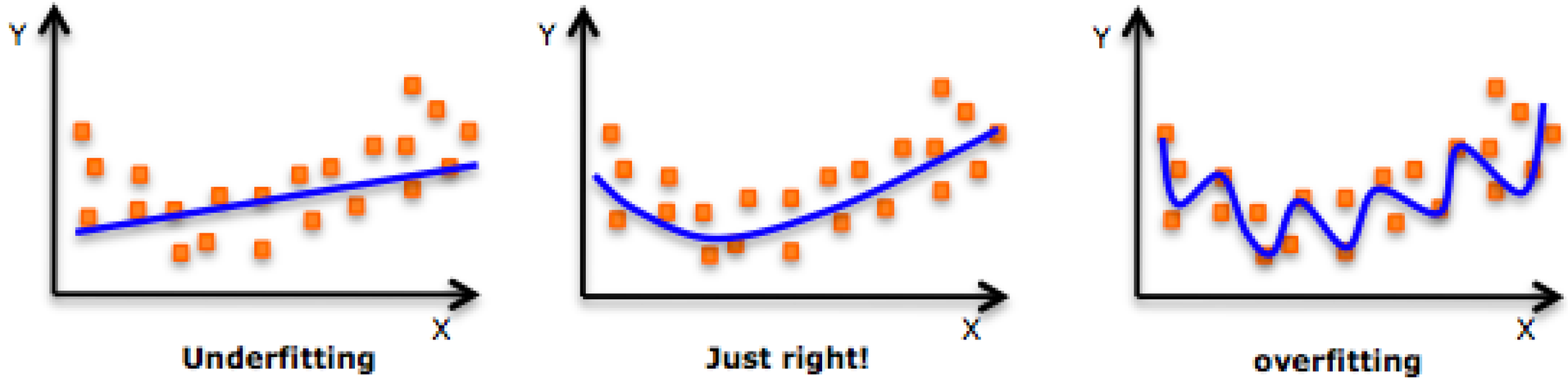
- Using the model for predicting:

```
y_pred = regressor.predict(X_test)
```

MULTIPLE LINEAR REGRESSION

The overfitting issue

- Should we consider all data for building the model?
  - Overfitting is when the model is overly trained to the dataset, which may result in capturing noise and producing a non-generalized model.
    ➔ mutually exclusive datasets (train/test split).
  - Still dependent on which datasets the data is trained and tested .
    ➔ K-fold cross-validation (check this page from towardsdatascience.com).



NON-LINEAR REGRESSION

Extended model

- Non-linear when it is not a linear combination of $\hat{\beta}_0$, $\hat{\beta}_1$, …, $\hat{\beta}_p$
  - Examples: $Y = \beta_0 + \beta_1 \beta_2{}^X + e$

    $Y = \log(\beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + e)$

    $Y = \dfrac{\beta_0}{1 + \beta_1{}^{X-\beta_2}}$

- In most situations, solving through a numerical approach.

MULTIPLE LINEAR REGRESSION

Example of implementation

- Data set: product sales w.r.t. ads expenditures (again)
- Objectives:
  - Train a multiple linear regression
  - Train a polynomial regression
  - Compare the accuracies

TV	radio	newspaper	sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	9.3
151.5	41.3	58.5	18.5
180.8	10.8	58.4	12.9
8.7	48.9	75	7.2
57.5	32.8	23.5	11.8
120.2	19.6	11.6	13.2
8.6	2.1	1	4.8

MULTIPLE LINEAR REGRESSION

Student practice

- Data set: CO_2 emission w.r.t. vehicle characteristics (again)
- Objectives:
  - Check for possible correlations
  - Train multiple linear regressions
  - Assess and compare the accuracy of the regressions
- Deadline: Wednesday July 7th 6pm

	YEAR	MAKE	MODEL	VEHICLECLASS	ENGINESIZE	CYLINDERS	MISSION	FUELTYPE	ON_CITY	_HWY	OMB	MPG	CO2EMISSIONS
1	2014	ACURA	ILX	COMPACT	2	4	AS5	Z	9.9	6.7	8.5	33	196
2	2014	ACURA	ILX	COMPACT	2.4	4	M6	Z	11.2	7.7	9.6	29	221
3	2014	ACURA	ILX HYBRID	COMPACT	1.5	4	AV7	Z	6	5.8	5.9	48	136
4	2014	ACURA	MDX 4WD	SUV - SMALL	3.5	6	AS6	Z	12.7	9.1	11.1	25	255
5	2014	ACURA	RDX AWD	SUV - SMALL	3.5	6	AS6	Z	12.1	8.7	10.6	27	244
6	2014	ACURA	RLX	MID-SIZE	3.5	6	AS6	Z	11.9	7.7	10	28	230
7	2014	ACURA	TL	MID-SIZE	3.5	6	AS6	Z	11.8	8.1	10.1	28	232
8	2014	ACURA	TL AWD	MID-SIZE	3.7	6	AS6	Z	12.8	9	11.1	25	255
9	2014	ACURA	TL AWD	MID-SIZE	3.7	6	M6	Z	13.4	9.5	11.6	24	267
10	2014	ACURA	TSX	COMPACT	2.4	4	AS5	Z	10.6	7.5	9.2	31	212
11	2014	ACURA	TSX	COMPACT	2.4	4	M6	Z	11.2	8.1	9.8	29	225
12	2014	ACURA	TSX	COMPACT	3.5	6	AS5	Z	12.1	8.3	10.4	27	239
13	2014	ASTON MARTIN	DB9	MINICOMPACT	5.9	12	A6	Z	18	12.6	15.6	18	359
14	2014	ASTON MARTIN	RAPIDE	SUBCOMPACT	5.9	12	A6	Z	18	12.6	15.6	18	359
15	2014	ASTON MARTIN	V8 VANTAGE	TWO-SEATER	4.7	8	AM7	Z	17.4	11.3	14.7	19	338
16	2014	ASTON MARTIN	V8 VANTAGE	TWO-SEATER	4.7	8	M6	Z	18.1	12.2	15.4	18	354
17	2014	ASTON MARTIN	V8 VANTAGE S	TWO-SEATER	4.7	8	AM7	Z	17.4	11.3	14.7	19	338

COURSE PROGRAM

Structure

PREPARATION
- Data exploration
- Data preprocessing

REGRESSIONS
- Linear regression with one variable
- Multiple and polynomial regressions
- Regression model assessment

CLASSIFICATIONS
- Logistic regression
- Classification model assessment
- Discriminant Analysis
- k-NN, Decision Tree, SVM

CLUSTERING
- k-means, hierarchical clustering

DIMENSIONALITY REDUCTION
- Principal Components Analysis
- Extensions

ALL NOTIONS
- Final assignment

REGRESSION MODEL ASSESSMENT

ANOVA

- **ANalysis Of VAriance**: the variance of Y is decomposed into a part corresponding to the regression and a part corresponding to the error: $\mathrm{SST} = \mathrm{SSR} + \mathrm{SSE}$

$$\mathrm{SST} = \mathrm{TSS} = \sum_{i=1}^{n} (Y_i - \bar{Y})^2$$

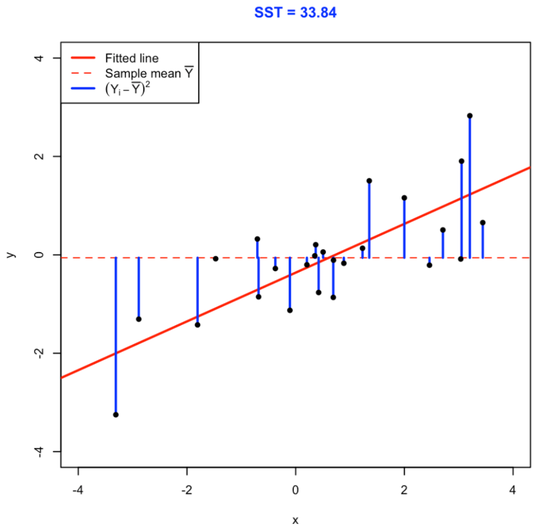
Total Sum of Squares. This is the total variation of $Y_1 \ldots Y_n$.

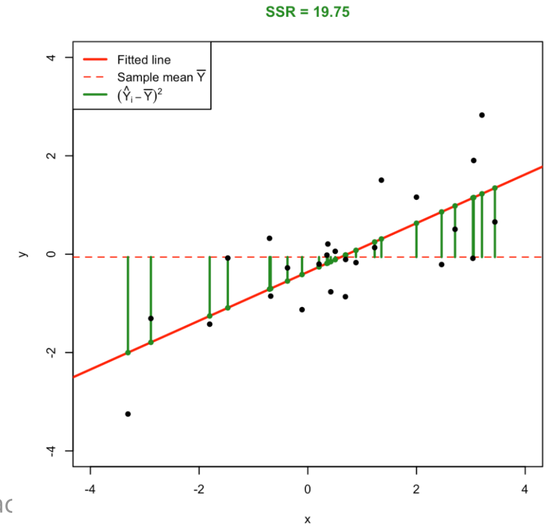
$$\mathrm{SSR} = \mathrm{ESS} = \sum_{i=1}^{n} \left(\hat{Y}_i - \bar{Y}\right)^2$$

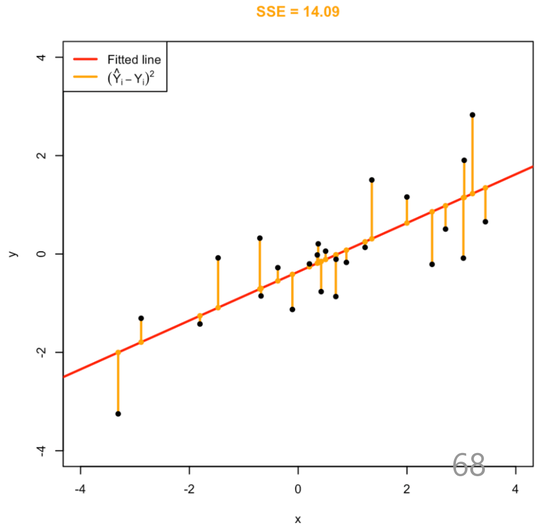
Regression Sum of Squares or **Explained Sum of Squares**. This is the variation explained by the regression line.

$$\mathrm{SSE} = \mathrm{RSS} = \sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)^2$$

Sum of Squared Errors or Residual Sum of Squares. This is the variation explained by the residuals.







REGRESSION MODEL ASSESSMENT

ANOVA

- The coefficient of determination R^2 measures the proportion of variability in Y that can be explained using X:

$$R^2 = \frac{\text{SSR}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}}$$

  - When close to 1: a large proportion of the variability in the response has been explained by the regression
  - When close to 0: the regression did not explain much of the variability in the response. This can occur when the linear model is wrong or when the inherent error is high.

REGRESSION MODEL ASSESSMENT

p-value

- Objective: check if a coefficient is statistically significant, i.e. if the predictor is related to the response.
- Method: testing the null hypothesis of (case of simple linear regression):

$$\boldsymbol{H_0}\textbf{: There is no relationship between } \boldsymbol{X} \textbf{ and } \boldsymbol{Y}$$

$$\textbf{(}\boldsymbol{Y = \beta_0 + e}\textbf{ i.e. }\boldsymbol{\beta_1 = 0}\textbf{)}$$

versus the alternative hypothesis:

$$\boldsymbol{H_1}\textbf{: There is some relationship between } \boldsymbol{X} \textbf{ and } \boldsymbol{Y}\ \textbf{(}\boldsymbol{\beta_1 \neq 0}\textbf{)}$$

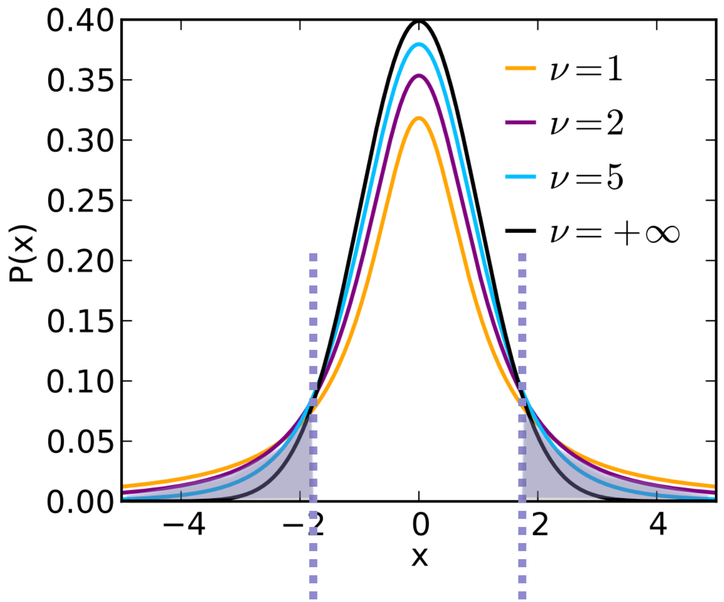
- To test the null hypothesis H_0, we compute the t-statistic, given by:

$$t = \frac{\hat{\beta}_1 - 0}{\mathrm{SE}(\hat{\beta}_1)}$$

REGRESSION MODEL ASSESSMENT

p-value

- Assuming $\beta_1 = 0$, it has a ***t*-distribution** (Student) with $n - 2$ degrees of freedom.
- Numeric computation of the **p-value**, that is the probability of observing any value equal or larger than $|t|$.
- If the p-value is small enough, we reject the null hypothesis, i.e. we declare that a relationship exists between X and Y.



	Coefficient	Std. error	t-statistic	p-value
Constant	2.939	0.3119	9.42	<0.0001
X_1	0.046	0.0014	32.81	<0.0001

< 1 % ➔ significant

REGRESSION MODEL ASSESSMENT

F-statistic

- Objective: check if the whole regression is explaining anything at all.
- Method: testing the null hypothesis of (case of multiple regression):

$$\boldsymbol{H_0: \beta_1 = \beta_2 = \cdots = \beta_p = 0}$$

versus the alternative hypothesis:

$$\boldsymbol{H_1}\textbf{: At least one } \boldsymbol{\beta_j} \textbf{ is non-zero}$$

- To test the null hypothesis H_0, we compute the F-statistic, given by:

$$F = \frac{(\mathrm{TSS} - \mathrm{RSS})/p}{\mathrm{RSS}/(n-p-1)} \sim F_{p,n-p-1}$$

REGRESSION MODEL ASSESSMENT

Python implementation

- Computing the coefficient of determination:

```
from sklearn.metrics import r2_score
print("R2-score: %.2f" % r2_score(y_test , y_pred))
```

- Computing the assessment report of a regression:

```
import statsmodels.api as sm
X_sm = sm.add_constant(X)
regressor = sm.OLS(y, X_sm).fit()
print(regressor.summary())
```

REGRESSION MODEL ASSESSMENT

Python implementation

- Interpretating an assessment report:

OLS Regression Results

Dep. Variable:	y	R-squared:	0.897
Model:	OLS	Adj. R-squared:	0.896
Method:	Least Squares	F-statistic:	570.3
Date:	Wed, 30 Dec 2020	Prob (F-statistic):	1.58e-96
Time:	14:07:38	Log-Likelihood:	-386.18
No. Observations:	200	AIC:	780.4
Df Residuals:	196	BIC:	793.6
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	2.9389	0.312	9.422	0.000	2.324	3.554
x1	0.0458	0.001	32.809	0.000	0.043	0.049
x2	0.1885	0.009	21.893	0.000	0.172	0.206
x3	-0.0010	0.006	-0.177	0.860	-0.013	0.011

Omnibus:	60.414	Durbin-Watson:	2.084
Prob(Omnibus):	0.000	Jarque-Bera (JB):	151.241
Skew:	-1.327	Prob(JB):	1.44e-33
Kurtosis:	6.332	Cond. No.	454.

Coefficients

p-values

Confidence interval

R-squared has a high value (89.6 % of the target variance is explained by the regression)

High F-statistics and very low p-value of the model, meaning that the model fit is statistically significant (the explained variance isn't purely by chance)

The coefficient of x1 is statistically significant (the association is not purely by chance)

The coefficient of x3 is not statistically significant (probability of 86 % to have this value or more when assuming H_0)

REGRESSION MODEL ASSESSMENT

Forward-Backward selection

- **Forward selection:**
  - Begin with the null model – a model that contains an intercept (β_0) but no predictors.
  - Fit p simple linear regressions and add to the null model the variable that results in the lowest RSS.
  - Add to that model the variable that results in the lowest RSS amongst all two-variable models.
  - Continue until some stopping rule is satisfied, for example when all remaining variables have a p-value above some threshold.

- **Backward selection:**
  - Start with all variables in the model.
  - Remove the variable with the largest p-value – that is, the variable that is the least statistically significant.
  - The new $(p-1)$-variable model is fit, and the variable with the largest p-value is removed.
  - Continue until a stopping rule is reached. For instance, we may stop when all remaining variables have a significant p-value defined by some significance threshold.

MULTIPLE LINEAR REGRESSION

Example of implementation

- Data set: product sales w.r.t. ads expenditures (again)
- Objectives:
  - Add assessment reports + interpretation

TV	radio	newspaper	sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	9.3
151.5	41.3	58.5	18.5
180.8	10.8	58.4	12.9
8.7	48.9	75	7.2
57.5	32.8	23.5	11.8
120.2	19.6	11.6	13.2
8.6	2.1	1	4.8

MULTIPLE LINEAR REGRESSION

Student practice

- Data set: CO_2 emission w.r.t. vehicle characteristics (again)
- Objectives:
  - Practices 3 & 4: add assessment reports to accuracy evaluation + interpretation
  - Bonus: forward selection on the candidate predictors
- Deadline: Sunday July 11th 6pm

	YEAR	MAKE	MODEL	VEHICLECLASS	ENGINESIZE	CYLINDERS	MISSION	FUELTYPE	ON_CITY	_HWY	OMB	MPG	CO2EMISSIONS
1	2014	ACURA	ILX	COMPACT	2	4	AS5	Z	9.9	6.7	8.5	33	196
2	2014	ACURA	ILX	COMPACT	2.4	4	M6	Z	11.2	7.7	9.6	29	221
3	2014	ACURA	ILX HYBRID	COMPACT	1.5	4	AV7	Z	6	5.8	5.9	48	136
4	2014	ACURA	MDX 4WD	SUV - SMALL	3.5	6	AS6	Z	12.7	9.1	11.1	25	255
5	2014	ACURA	RDX AWD	SUV - SMALL	3.5	6	AS6	Z	12.1	8.7	10.6	27	244
6	2014	ACURA	RLX	MID-SIZE	3.5	6	AS6	Z	11.9	7.7	10	28	230
7	2014	ACURA	TL	MID-SIZE	3.5	6	AS6	Z	11.8	8.1	10.1	28	232
8	2014	ACURA	TL AWD	MID-SIZE	3.7	6	AS6	Z	12.8	9	11.1	25	255
9	2014	ACURA	TL AWD	MID-SIZE	3.7	6	M6	Z	13.4	9.5	11.6	24	267
10	2014	ACURA	TSX	COMPACT	2.4	4	AS5	Z	10.6	7.5	9.2	31	212
11	2014	ACURA	TSX	COMPACT	2.4	4	M6	Z	11.2	8.1	9.8	29	225
12	2014	ACURA	TSX	COMPACT	3.5	6	AS5	Z	12.1	8.3	10.4	27	239
13	2014	ASTON MARTIN	DB9	MINICOMPACT	5.9	12	A6	Z	18	12.6	15.6	18	359
14	2014	ASTON MARTIN	RAPIDE	SUBCOMPACT	5.9	12	A6	Z	18	12.6	15.6	18	359
15	2014	ASTON MARTIN	V8 VANTAGE	TWO-SEATER	4.7	8	AM7	Z	17.4	11.3	14.7	19	338
16	2014	ASTON MARTIN	V8 VANTAGE	TWO-SEATER	4.7	8	M6	Z	18.1	12.2	15.4	18	354
17	2014	ASTON MARTIN	V8 VANTAGE S	TWO-SEATER	4.7	8	AM7	Z	17.4	11.3	14.7	19	338

EXTENSIONS

WORK METHODS

Efficiency tips

- Jupyter Notebook shortcuts:
  - **M**: turn selected cell in Markdown
  - **Enter**: edit selected cell
  - **Ctrl+Enter**: run selected cell
  - **A**: add new cell above selected cell
  - **B**: add new cell below selected cell
  - **D+D**: delete selected cell
  - **C**: copy selected cell
  - **V**: paste copied cell
  - **Ctrl+Shift+"-"**: split the current cell at cursor position

REFERENCES

Refreshers on Python

Crash Course on Python

Google

Cours

4.8 (12976) | 260K étudiants

Beginner

Introduction to Data Science in Python

University of Michigan

Cours

4.5 (23064) | 570 000 étudiants

Intermediate

DATA - COURSE

Learn Python Basics for Data Analysis

Easy 12 hours

Find out how fun and rewarding programming in Python can be! In this course, you will learn how to use and write functions, get hands-on ...

DATA - COURSE

Use Python libraries for Data Science

Medium 8 hours

Python has emerged as a prominent language for all things Data Science. Operate in a Jupyter notebook and learn how to use the essential ...

REFERENCES

Materials on ML techniques