esade

Intermediate Python
&
a Glimpse into AI applications

Class 4

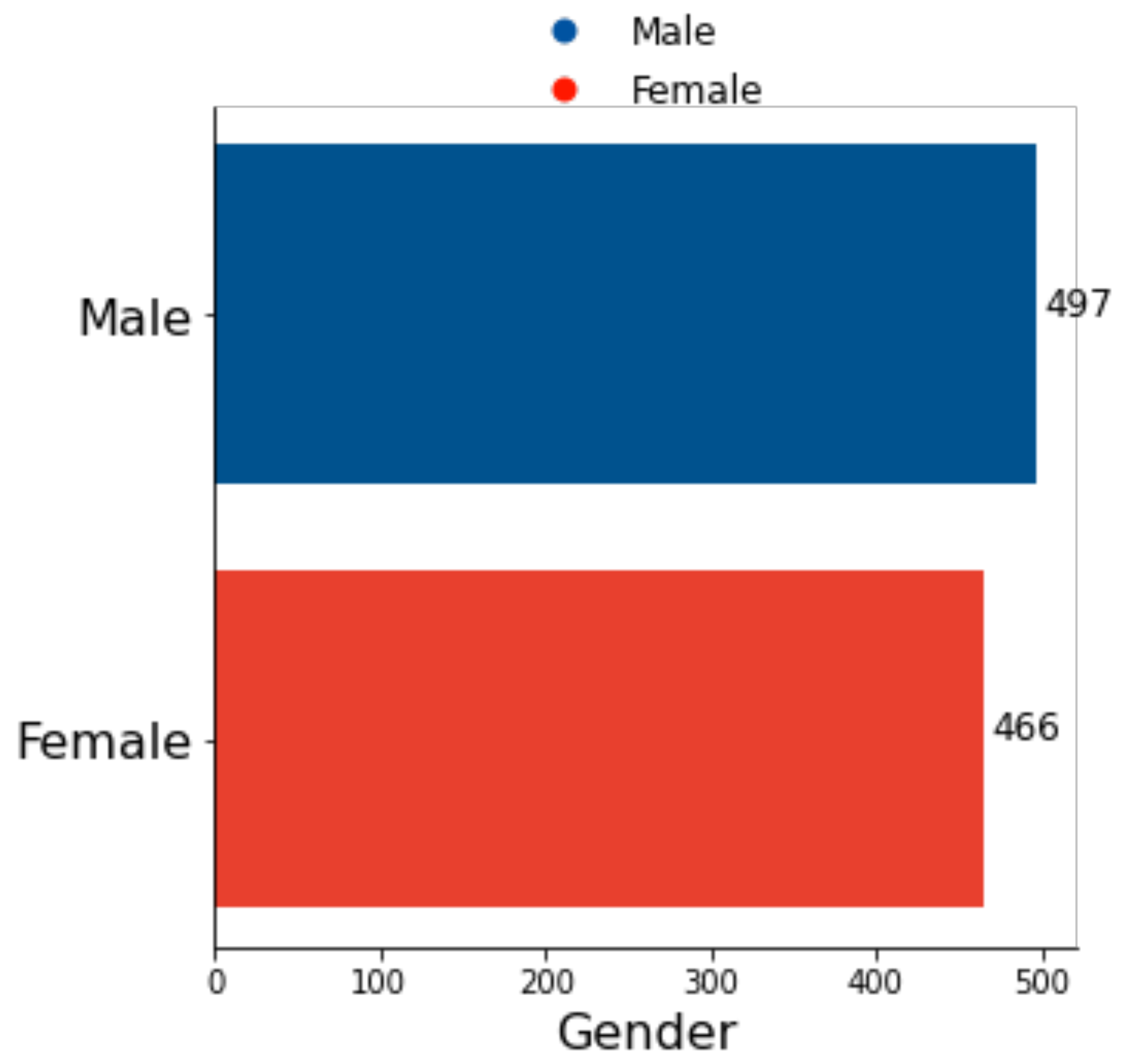
Pepe Bonet Giner

6th June 2023

Index Class 4

Topic 1: Recap

Recap: Matplotlib/Seaborn and EDA steps



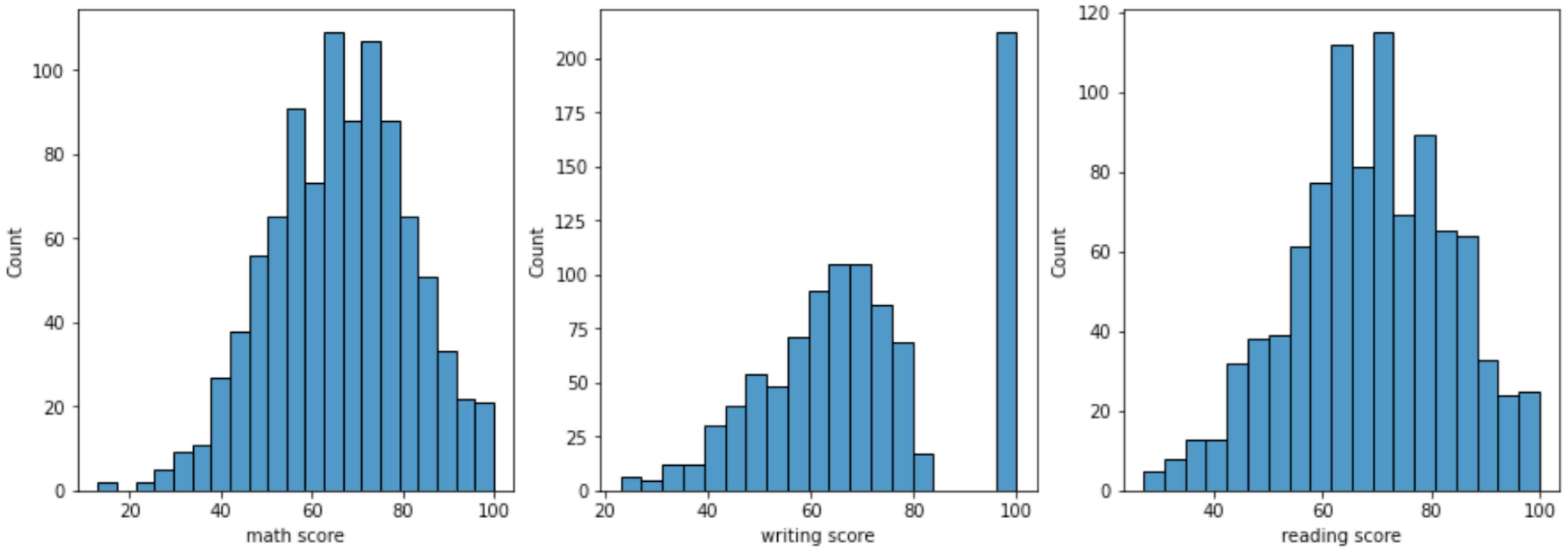
matplotlib seaborn

```
#Start Figure
fig, ax = plt.subplots(figsize=(5, 5))

#Body of the figure to build and the data to use
sns.barplot(x=to_plot['gender'], y=to_plot['index'],
            palette=['#08519c', '#f03b20'])

#Change Axes
ax.set_xlabel("Gender", fontsize=16)
ax.set_ylabel("", fontsize=16)
ax.set_yticklabels(['Male', 'Female'], fontsize=16)
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)

# Add Numbers to plot
for index, row in to_plot.iterrows():
    ax.text(row.gender + 25, index, row.gender,
            color='black', ha="center", fontsize=12)

## Add legend
custom_lines = []
for el in [('Male', '#08519c'), ('Female', '#f03b20')]:
    custom_lines.append(
        plt.plot([],[], marker="o", ms=8, ls="", mec='black',
        mew=0, color=el[1], label=el[0])[0]
    )
ax.legend(
    bbox_to_anchor=(0., 1.05, 1., .102),
    handles=custom_lines, loc='upper center',
    facecolor='white', ncol=1, fontsize=12, frameon=False
    )
#Save or show
plt.show()
```

Recap: EDA & Feature Engineering

EDA

- Is a process where users look at and understand their data with statistical and visualization methods
- Categorical EDA
- Numerical EDA



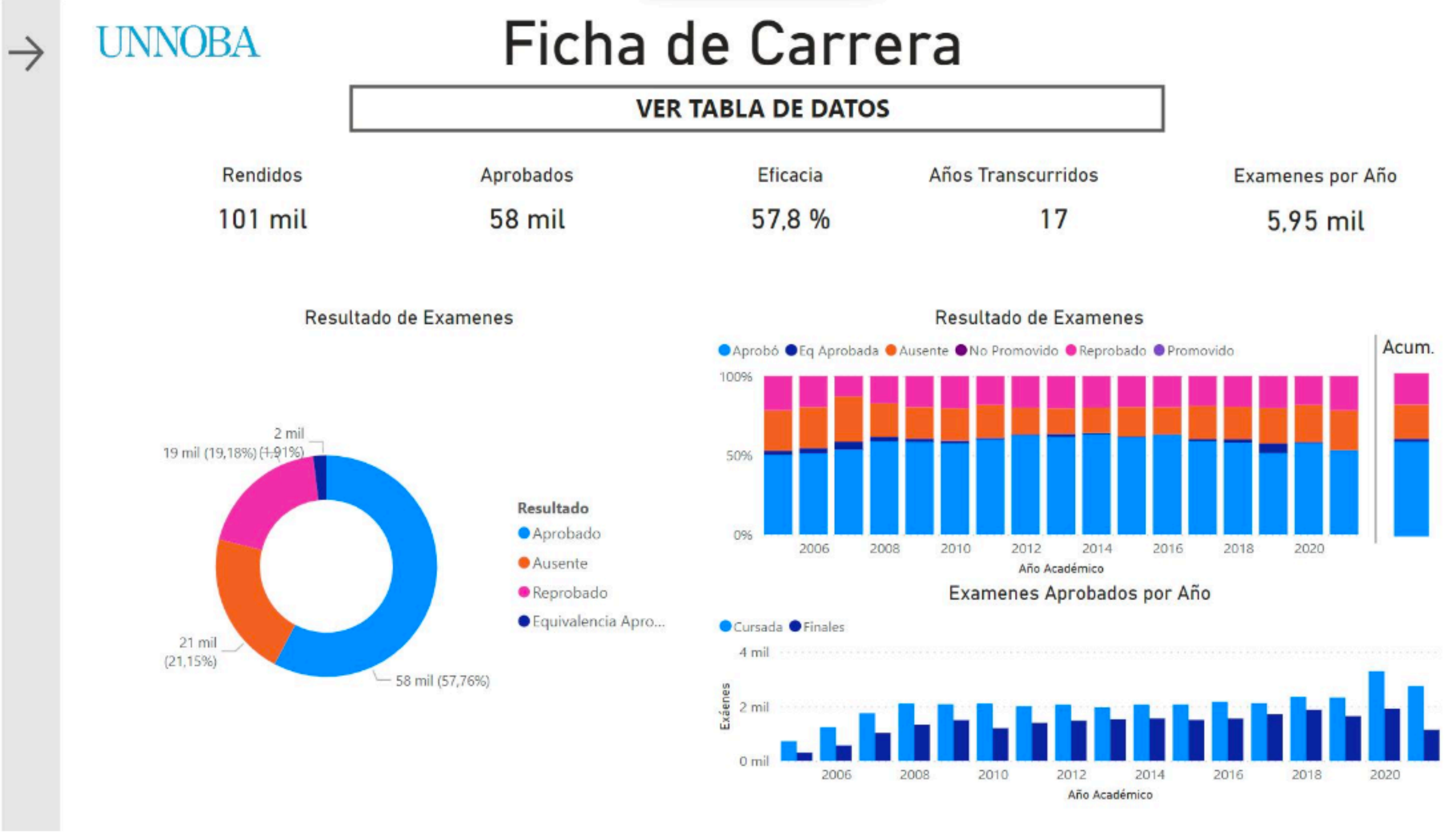
esade

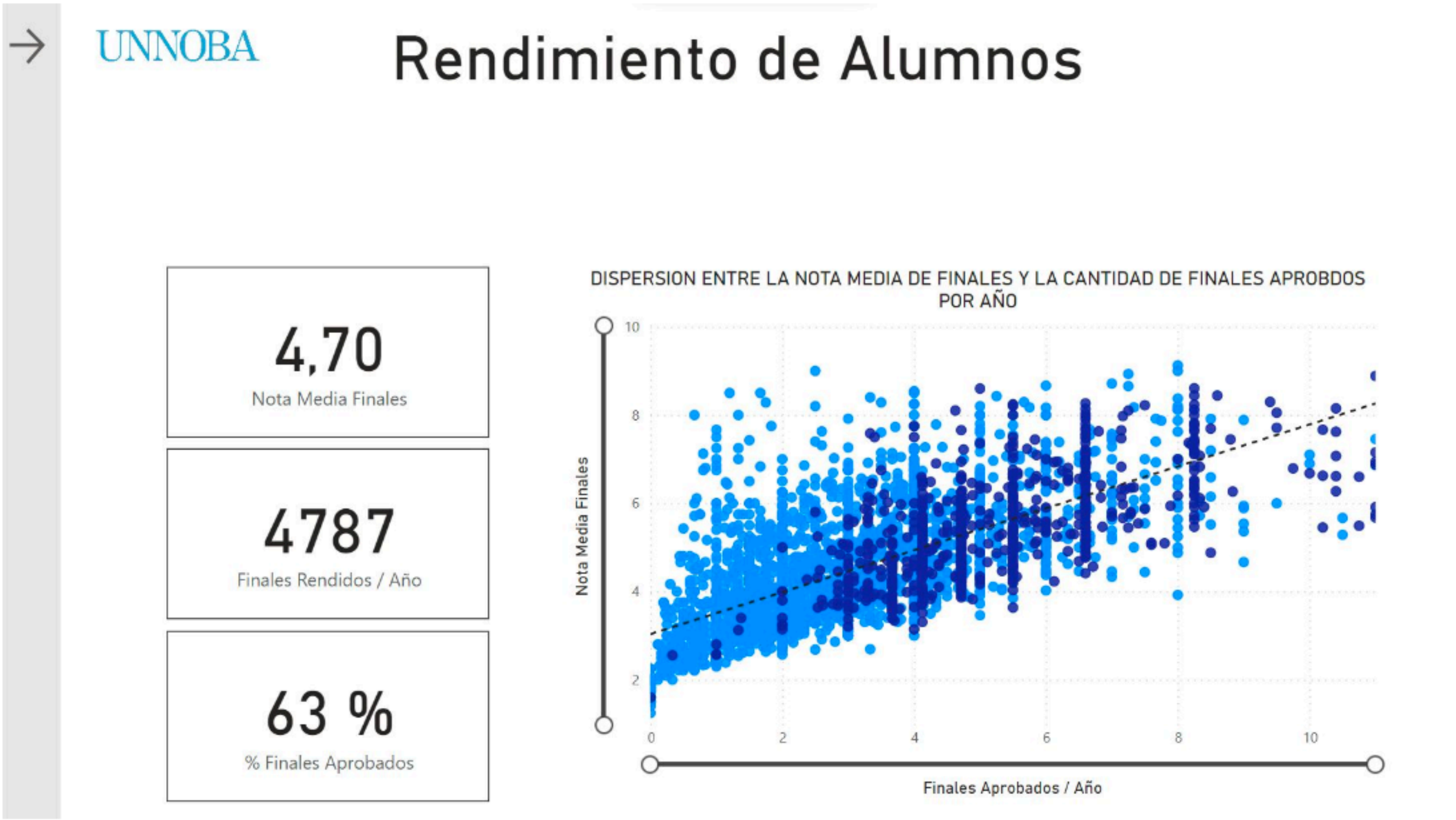
Topic 2: A bit more of seaborn + Saving outputs

Exercise

- Run your code of the second class and save the resulting dataframe after fixing structural errors, imputing missing values etc…
- Load it in the Jupyter notebooks of today and show that it is the cleaned version of it (shape, isna().sum(), df['gender'].value_counts(), etc…)

All this process can already give significant value to companies

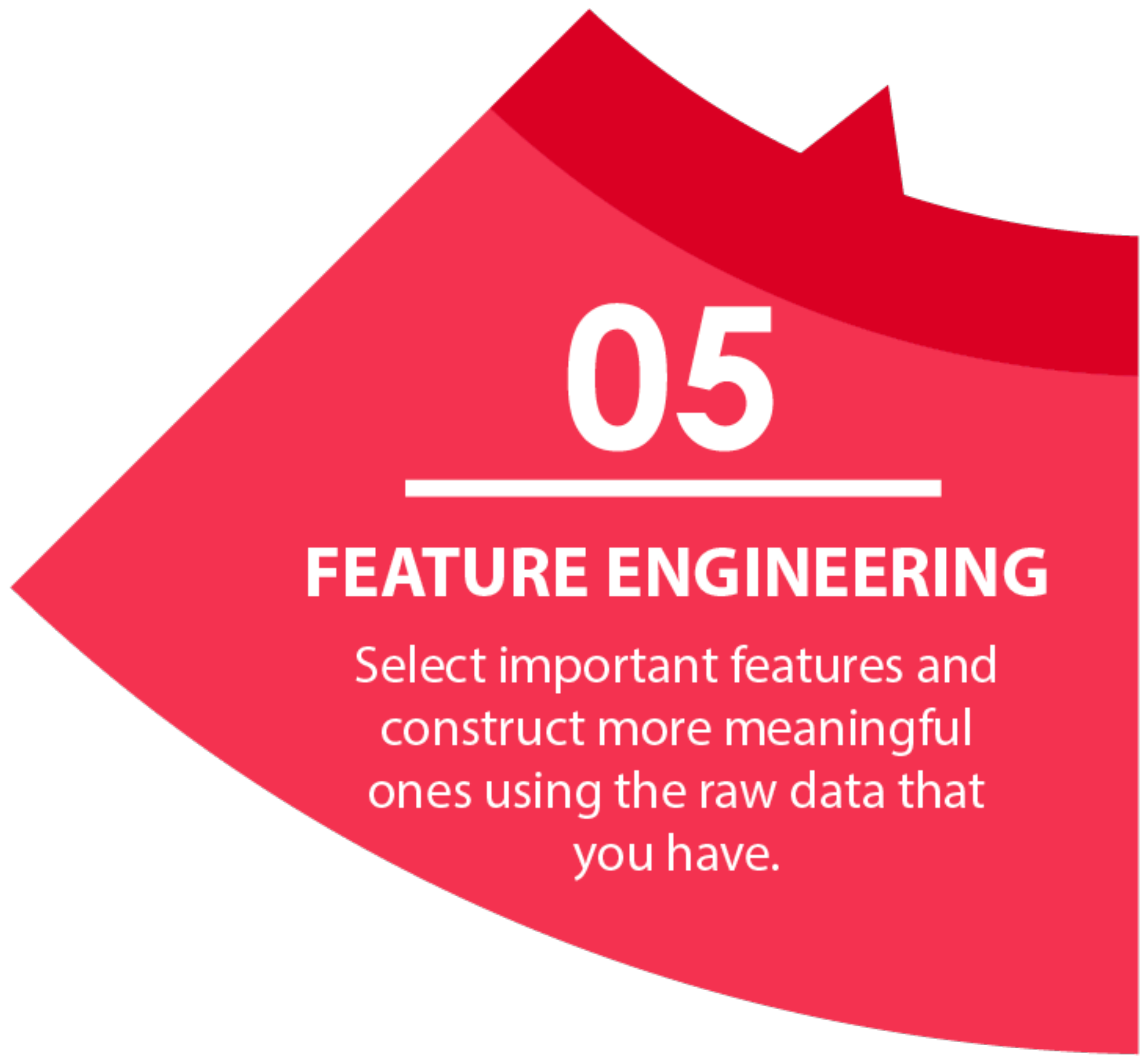




esade

Topic 3: Feature Engineering

Feature engineering



- Is the process of using domain knowledge to extract features (characteristics, properties, attributes) from raw data
- Examples of current projects:
    - Chatbot: #TimesAsking4help, #Words
    - Universities: #PassedSubjects, GradeMean

Feature engineering

Transform the parental level of education

```
df['parental level of education'].value_counts()
```

```
some college          219
associate's degree    196
high school           194
some high school      180
bachelor's degree     107
master's degree        67
Name: parental level of education, dtype: int64
```

```
new_col = []
for el in df['parental level of education'].tolist():
    if el in ["master's degree", "bachelor's degree", "some college"]:
        new_col.append('went to college')
    else:
        new_col.append('no college')
df['Education'] = new_col
```

Feature engineering

Transform the parental level of education

```
df['parental level of education'].value_counts()
```

```
some college          219
associate's degree    196
high school           194
some high school      180
bachelor's degree     107
master's degree        67
Name: parental level of education, dtype: int64
```

```
new_col = []
for el in df['parental level of education'].tolist():
    if el in ["master's degree", "bachelor's degree", "some college"]:
        new_col.append('went to college')
    else:
        new_col.append('no college')
df['Education'] = new_col
```

BMI Calculation

```
df['height'] = 1.70
df['weight'] = 70
```

```
df['BMI'] =  round(df['weight'] / (df['height'] * df['height']), 2)
```

```
df.drop(['height', 'weight'], axis=1, inplace=True)
```

Exercise

- Add a new feature (column) to the dataset named *scores mean* that represents the mean of all three grades (math, writing and reading)

Exercise

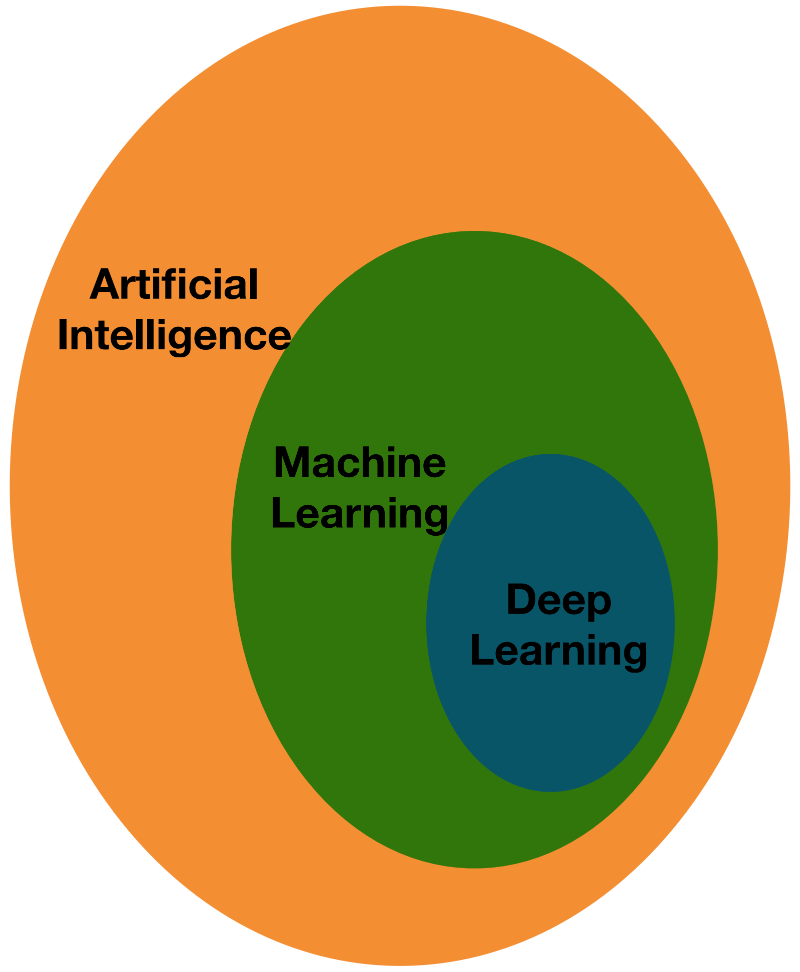
- Add a new feature (column) to the dataset named *scores mean* that represents the mean of all three grades (math, writing and reading)

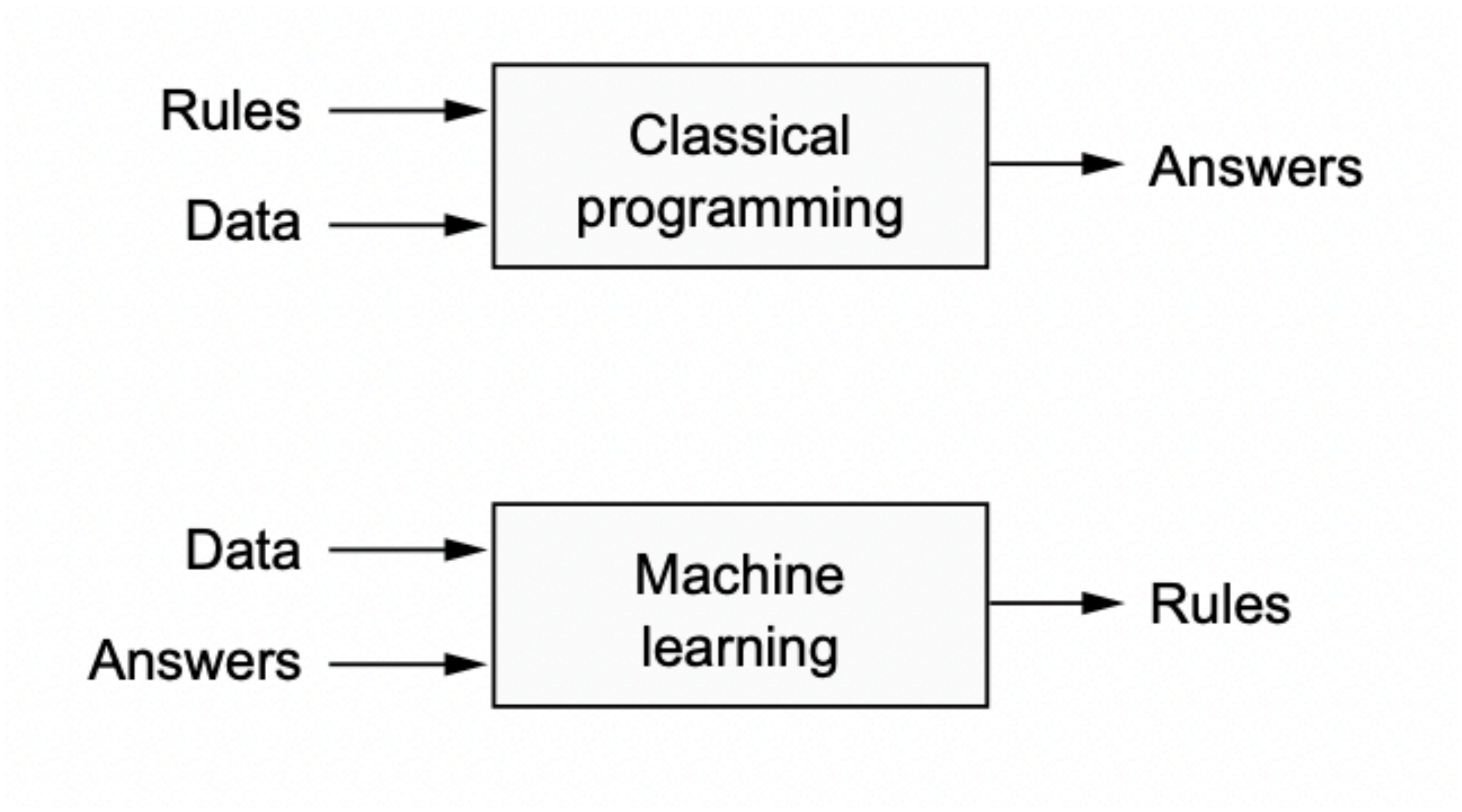
```
df['scores mean'] = df[['math score', 'writing score', 'reading score']].mean(axis=1)

df = df.assign(mean = round((df['writing score'] + df['math score'] + df['reading score'])/3, 2))

df = df.assign(Scores_Mean = lambda x: (x['math score']+x['writing score']+x['reading score'])/3)

df['scores_mean'] = round((df['writing score'] + df['math score'] + df['reading score'])/3, 2)
```

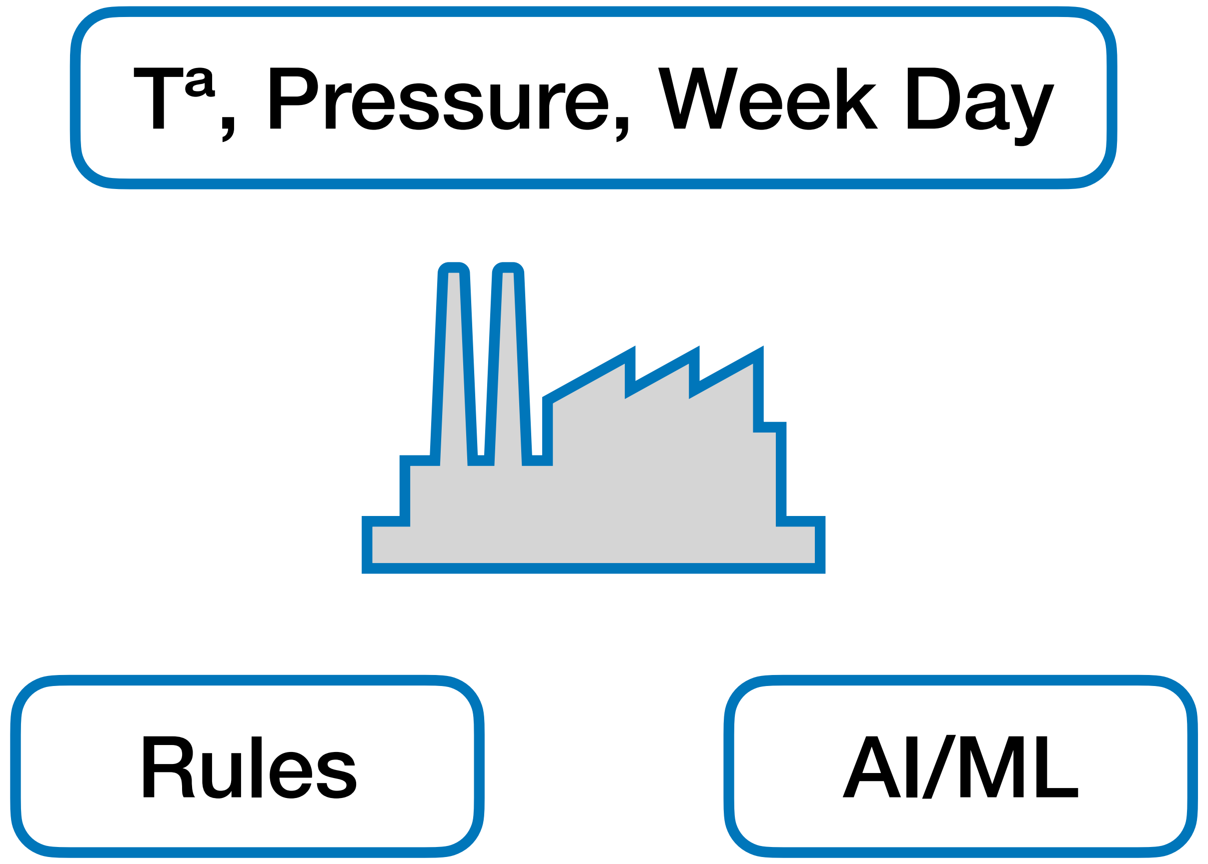
esade

Topic 4: Intro to the world of predictive models

A shift of paradigm



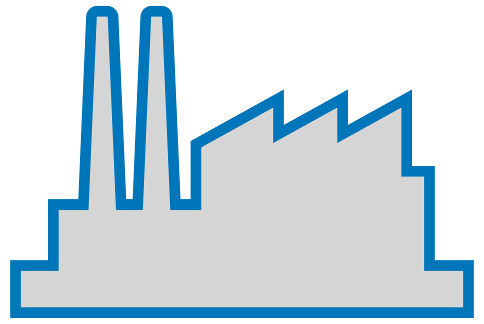




- AI: Effort to automate intellectual tasks normally performed by humans
- ~ 1950-1980 Symbolic AI. Rules + Data —> Answers
- ML: Data + Answers —> Rules

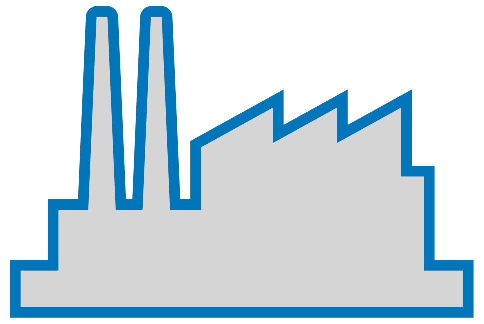
Learning from the past to predict the future

Rules



```
if Tª > 50º:
    return “We are in danger!”
elif Tª < 10º:
    return “The tank is too cold!”
else:
    return “Everything is fine!”
```

AI/ML



Collect data throughout time

Identifier | Features | Label

ID	Tª	Pressure	Week Day	Fault
1	35	10	1	No
2	68	25	2	Yes
3	42	10	1	No
4	15	12	6	No

Learning from the past to predict the future

Identifier

Features

Label

ID	Tª	Pressure	Week Day	Fault
1	35	10	1	No
2	68	25	2	Yes
3	42	10	1	No
4	15	12	6	No

Predict Failure

Learning from the past to predict the future

Identifier

Features

Label

ID	Tª	Pressure	Week Day	Fault
1	35	10	1	No
2	68	25	2	Yes
3	42	10	1	No
4	15	12	6	No

Predict Failure

ID	#Bedrooms	#bathrooms	M2	Price(Euro)
1	3	2	120	190.000
2	3	3	150	250.000
3	1	1	60	100.000
4	2	2	80	140.000

Predict House Prices

Learning from the past to predict the future

Identifier | Features | Label

ID	Tª	Pressure	Week Day	Fault
1	35	10	1	No
2	68	25	2	Yes
3	42	10	1	No
4	15	12	6	No

Predict Failure

ID	#Bedrooms	#bathrooms	M2	Price(Euro)
1	3	2	120	190.000
2	3	3	150	250.000
3	1	1	60	100.000
4	2	2	80	140.000

Predict House Prices

Name	MeanGrades	#PassedSubjects	Dropout
David	4	10	Yes
Evi	8	20	No
Fran	9	25	No
Encarna	7	18	No

Predict Student Dropout

Learning from the past to predict the future

Identifier | Features | Label

Data From the past

ID	Tª	Pressure	Week Day	Fault
1	35	10	1	No
2	68	25	2	Yes
3	42	10	1	No
4	15	12	6	No

New data at the present point

New data point:
ID: 1234
Tª: 65
Pressure: 24
Week Day 3

Predict the future

Predict whether it will fail

It will fail!!!!

Learning from the past to predict the future

Identifier

Features

Label

Data From the past

ID	#Bedrooms	#bathrooms	M^2	Price(Euro)
1	3	2	120	190.000
2	3	3	150	250.000
3	1	1	60	100.000
4	2	2	80	140.000

New data at the present point

New data point:
ID: 1234
#Bedrooms: 5
#bathrooms: 4
M^2 : 200

Predict the future

Predict the price

350.000

How do we learn from the past? We need something. **A model**

Identifier

Features

Label

Data From the past

ID	#Bedrooms	#bathrooms	M^2	Price(Euro)
1	3	2	120	190.000
2	3	3	150	250.000
3	1	1	60	100.000
4	2	2	80	140.000

How do we learn from the past? We need something. **A model**

Identifier | Features | Label

Data From the past

ID	#Bedrooms	#bathrooms	M^2	Price(Euro)
1	3	2	120	190.000
2	3	3	150	250.000
3	1	1	60	100.000
4	2	2	80	140.000

Train a model using the data from the past

Predictive Model
Black box

How do we learn from the past? We need something. **A model**

Identifier | Features | Label

Data From the past

ID	#Bedrooms	#bathrooms	M^2	Price(Euro)
1	3	2	120	190.000
2	3	3	150	250.000
3	1	1	60	100.000
4	2	2	80	140.000

Train a model using the data from the past

Predictive Model
Black box

Trained Model

How do we learn from the past? We need something. **A model**

Identifier | Features | Label

Data From the past

ID	#Bedrooms	#bathrooms	M^2	Price(Euro)
1	3	2	120	190.000
2	3	3	150	250.000
3	1	1	60	100.000
4	2	2	80	140.000

Train a model using the data from the past

Predictive Model
Black box

Trained Model

New data point:
ID: 1234
#Bedrooms: 5
#bathrooms: 4
M^2 : 200

Use the trained model to predict the price of future/ new data point

350.000

What is a model

- Models are programs that can find patterns in the data that is fed to them

#Bedrooms	#bathrooms	M^2
3	2	120
3	3	150
1	1	60
2	2	80

Find Patterns

Price(Euro)
190.000
250.000
100.000
140.000

More bathrooms more money
More square meter more money
Less noise more money
Less bedrooms less money…

esade

Topic 5: Supervised vs. Unsupervised. Regression vs. Classification

AI & Machine Learning

ML: Data (input) + Answers (predictive label) —> Rules (Model)

Input Data

#Bedrooms	#bathrooms	M^2
3	2	120
3	3	150
1	1	60
2	2	80

+

Answers

Price(Euro)
190.000
250.000
100.000
140.000

=

Rules

Model

- Transform data into meaningful outputs
  (Learning from exposure to known examples)

Supervised vs. Unsupervised learning

- We distinguish between two types of models based on how they function and learn

Supervised Learning

#Bedrooms	#bathrooms	M²
3	2	120
3	3	150
1	1	60
2	2	80

Find Patterns

Price(Euro)
190.000
250.000
100.000
140.000

Gradient Boosting
Random Forest
Decision Trees
Neural Networks

F E A T U R E S

PREDICTION LABEL

Unsupervised Learning

#Bedrooms	#bathrooms	M²
3	2	120
3	3	150
1	1	60
2	2	80

Find Patterns

Clustering: Customer Segmentation
PCA: Dimensionality Reduction
NNs: Anomaly Detection

Supervised vs. Unsupervised learning

- We distinguish between two types of models based on how they function and learn

Supervised Learning

#Bedrooms	#bathrooms	M^2
3	2	120
3	3	150
1	1	60
2	2	80

Find Patterns

Price(Euro)
190.000
250.000
100.000
140.000

Gradient Boosting
Random Forest
Decision Trees
Neural Networks

FEATURES

PREDICTION LABEL

We focus on supervised learning

Unsupervised Learning

#Bedrooms	#bathrooms	M^2
3	2	120
3	3	150
1	1	60
2	2	80

Find Patterns

Clustering: Customer Segmentation
PCA: Dimensionality Reduction
NNs: Anomaly Detection

Classification vs. Regression Problems

Classification

ID	Tª	Pressure	Week Day	Fault
1	35	10	1	No
2	68	25	2	Yes
3	42	10	1	No
4	15	12	6	No

Regression

ID	#Bedrooms	#bathrooms	M^2	Price(Euro)
1	3	2	120	190.000
2	3	3	150	250.000
3	1	1	60	100.000
4	2	2	80	140.000

Classification vs. Regression Problems

Classification

ID	Tª	Pressure	Week Day	Fault
1	35	10	1	No
2	68	25	2	Yes
3	42	10	1	No
4	15	12	6	No

Regression

ID	#Bedrooms	#bathrooms	M^2	Price(Euro)
1	3	2	120	190.000
2	3	3	150	250.000
3	1	1	60	100.000
4	2	2	80	140.000

Classification is about predicting a label (Categorical) and regression is about predicting a quantity (Numerical)

Classification vs. Regression Problems

Classification

ID	Tª	Pressure	Week Day	Fault
1	35	10	1	No
2	68	25	2	Yes
3	42	10	1	No
4	15	12	6	No

Regression

ID	#Bedrooms	#bathrooms	M^2	Price(Euro)
1	3	2	120	190.000
2	3	3	150	250.000
3	1	1	60	100.000
4	2	2	80	140.000

Classification is about predicting a label (Categorical) and regression is about predicting a quantity (Numerical)

1.- Spam email detection
2.- Sentiment analysis
3.- Image classification
4.- Fraud detection
5.- Disease diagnosis
6.- Customer churn prediction
7.- Handwritten digit recognition
8.- Intrusion detection in network security
9.- Document classification

1.- Predicting house prices
2. -Estimating the sales volume of a product
3.- Forecasting stock market prices
4.- Predicting a student's GPA
5.- Estimating the energy consumption of a building
6.- Predicting the lifespan of a mechanical component
7.- Forecasting the demand for a particular product
8.- Estimating the price of a used car
9.- Predicting the success rate of a marketing campaign

esade

Topic 6: Arranging data for models - Data Preparation

Computers only understand numbers!

Categorical Features

Numerical Features

Gender	Parent Education	Age	Grade
Male	College	21	8
Female	High School	22	9
Male	None	20	6
Female	College	21	5

Computers need to code the categorical features/ columns as numbers. How do we do it?

We have two main approaches:

- Label Encoding
- One Hot Encoding

Label Encoding

Process of converting the labels into a numeric form so as to convert them into the machine-readable form

Parent Education
College
High School
None
College

Label Encoding

None: 0
High School: 1
College: 2

Parent Education
2
1
0
2

Label Encoding

Process of converting the labels into a numeric form so as to convert them into the machine-readable form

Parent Education
College
High School
None
College

Label Encoding

None: 0
High School: 1
College: 2

Parent Education
2
1
0
2

Limitations / Disadvantages

Numeric values can be "misinterpreted" by the algorithms. For example, the value of 0 is obviously less than the value of 4 but is that so?

Label Encoding

Process of converting the labels into a numeric form so as to convert them into the machine-readable form

Parent Education
College
High School
None
College

Label Encoding

None: 0
High School: 1
College: 2

Parent Education
2
1
0
2

Limitations / Disadvantages

Numeric values can be "misinterpreted" by the algorithms. For example, the value of 0 is obviously less than the value of 4 but is that so?

Python Code

```
df["parent_ed_cat_encode"] = df["parental level of education"]\
    .astype('category').cat.codes
```

```
df[['parental level of education', 'parent_ed_cat_encode']].head()
```

	parental level of education	parent_ed_cat_encode
0	high school	2
1	some college	4
2	high school	2
3	associate's degree	0
4	high school	2

Exercise

- Test label encoding to the column race/ethnicity. You can use pandas or scikitlearn
- Delete the new column when done or do not modify the source dataframe.
- Does label encoding makes sense here?

One hot encoding

Convert each category value into a new column and assigns a 1 or 0 (True/False) value to the column.

Parent Ed
College
High School
None
College

One Hot Encoding

Add dummy variables for each category

Parent Ed_College	Parent Ed_High School	Parent Ed_None
1	0	0
0	1	0
0	0	1
1	0	0

One hot encoding

Convert each category value into a new column and assigns a 1 or 0 (True/False) value to the column.

Parent Ed
College
High School
None
College

One Hot Encoding

Add dummy variables for each category

Parent Ed_College	Parent Ed_High School	Parent Ed_None
1	0	0
0	1	0
0	0	1
1	0	0

Limitations / Disadvantages

It has the benefit of not weighting a value improperly but does have the downside of adding more columns to the data set.

One hot encoding

Convert each category value into a new column and assigns a 1 or 0 (True/False) value to the column.

Parent Ed
College
High School
None
College

One Hot Encoding

Add dummy variables for each category

Parent Ed_College	Parent Ed_High School	Parent Ed_None
1	0	0
0	1	0
0	0	1
1	0	0

Limitations / Disadvantages

It has the benefit of not weighting a value improperly but does have the downside of adding more columns to the data set.

Python Code

```
pd.get_dummies(df['gender'], columns=['gender']).head()
```

	female	male
0	0	1
1	0	1
2	0	1
3	0	1
4	1	0

Exercise

- Do one-hot encoding of all categorical columns

Data Normalisation

- Normalization techniques can be used to change a continuous feature to fall within a specified range while maintaining the relative differences between the values for the feature.

- The simplest approach to normalization is **range normalization**,

- Range normalization performs a linear scaling of the original values of the continuous feature into a given range ([0, 1]).

$$x_{inorm} = \frac{x_i - X_{min}}{X_{max} - X_{min}} \cdot (range_{max} - range_{min}) + range_{min}$$

Data Normalisation Practice

$$x_{inorm} = \frac{x_i - X_{min}}{X_{max} - X_{min}} \cdot (range_{max} - range_{min}) + range_{min}$$

Age
10
25
53
42

Data Normalisation Practice

$$x_{inorm} = \frac{x_i - X_{min}}{X_{max} - X_{min}} \cdot (range_{max} - range_{min}) + range_{min}$$

Age
10
25
53
42

Age_norm
0
0.35
1
0.74

Exercise

- Write a function that normalizes any column that you input in the range [0, 1] and test it.
- Check that your output is correct using MinMaxScaler from scikit learn

Train & Test sets. Why are they necessary?

Overfitting: The model is not good on never seen data

Underfitting: The model is bad everywhere

The reason to have a separate train and test set is to avoid overfitting of the model to the data

Train & Test sets. Why are they necessary?

Overfitting: The model is not good on never seen data

Underfitting: The model is bad everywhere

The reason to have a separate train and test set is to avoid overfitting of the model to the data

Training Set

Gender	Parent Education	Age	Grade
Male	College	21	8
Female	High School	22	9

To train the model. Can represent up to 90 % of the data

Test Set

Gender	Parent Education	Age	Grade
Male	None	20	6
Female	College	21	5

To test how the model generalizes to never seen data

esade

Documentation

Datacamp Courses

Python Fundamentals

Are you ready to gain the foundational skills you need to become a Python programmer? In this track, you'll learn the Python basics you need to start on your programming journey, including how to clean real-world data ready for analysis, use data visualization libraries, and even how to write your own Python functions.

Your instructor Hugo will introduce you to how companies worldwide use Python to gain a competitive edge. Through hands-on coding exercises you'll then learn how to store, manipulate, and explore data using NumPy. Then it's time to level-up as you learn how to visualize your data using Matplotlib, manipulate DataFrames and dictionaries using pandas, and write your own functions and list comprehension. Start this track to add these essential Python skills to your data science toolbox.

Switch Track

Python 15 hours 4 Courses 1 Skill Assessment

Data Scientist with Python

Gain the career-building Python skills you need to succeed as a data scientist. No prior coding experience required.

In this track, you'll learn how this versatile language allows you to import, clean, manipulate, and visualize data—all integral skills for any aspiring data professional or researcher. Through interactive exercises, you'll get hands-on with some of the most popular Python libraries, including pandas, NumPy, Matplotlib, and many more. You'll then work with real-world datasets to learn the statistical and machine learning techniques you need to train decision trees and use natural language processing (NLP). Start this track, grow your Python skills, and begin your journey to becoming a confident data scientist.

Resume Track

Python 88 hours 23 Courses 6 Projects 3 Skill Assessments

Free Books to learn Python & data analytics