What Shapes Prices in Airbnb ?
- OCI Data Science in Action

Make IT
2025

About Me – Tomasz Ziss

- ~15 years in IT industry
- Cloud Advisory Innovation Principal - Accenture Enkitec Group
- Oracle Ace Associate
- Oracle Certified Master
- Multi-cloud certifications – OCI, AWS, Azure, GCP
- Postgraduated in Data Engineering and Machine Learning
- Blogger -> https://tziss.wordpress.com

Agenda

- ❑ Business use case – What shapes Prices in Airbnb?
- ❑ Machine Learning Lifecycle, AutoML and ML Interpretability
- ❑ Regression Analysis – quick guide
- ❑ Airbnb – Data Analysis and Executive Summary
- ❑ Summary

Github Repo with Code

https://github.com/zizu1985/MakeIT2025

Stanisław Lem – Polish futorologist

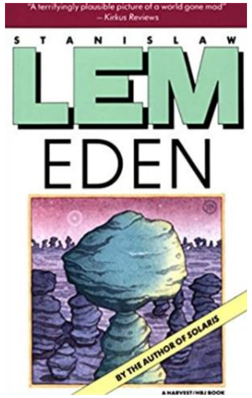
<u>Themes raised in the work</u>

- Microservices win with „big machines”
- Using neural networks (funny example of diplomacy with planet 200 light-years away)
- Human vs technology
- Shortcomings of humans
- World with elimination of social evil (and killed human development and freedom of choice)
- World where (nearly) everything has been automated

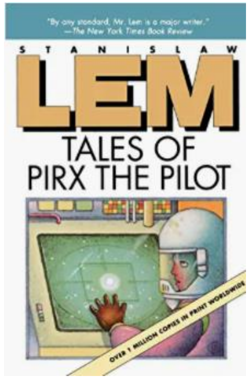
Stanisław Lem – Polish futorologist

[Solaris]

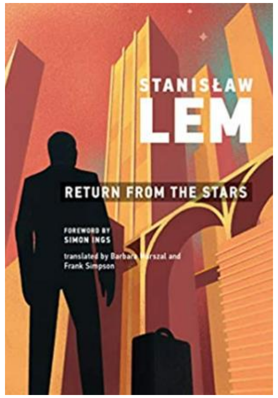
[Local vision]



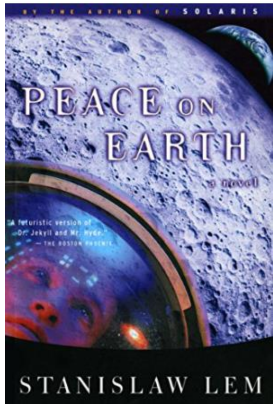
[Eden]



[Pirx Pilot]



[Return from Stars]



[Peace on Earth]

Business use case

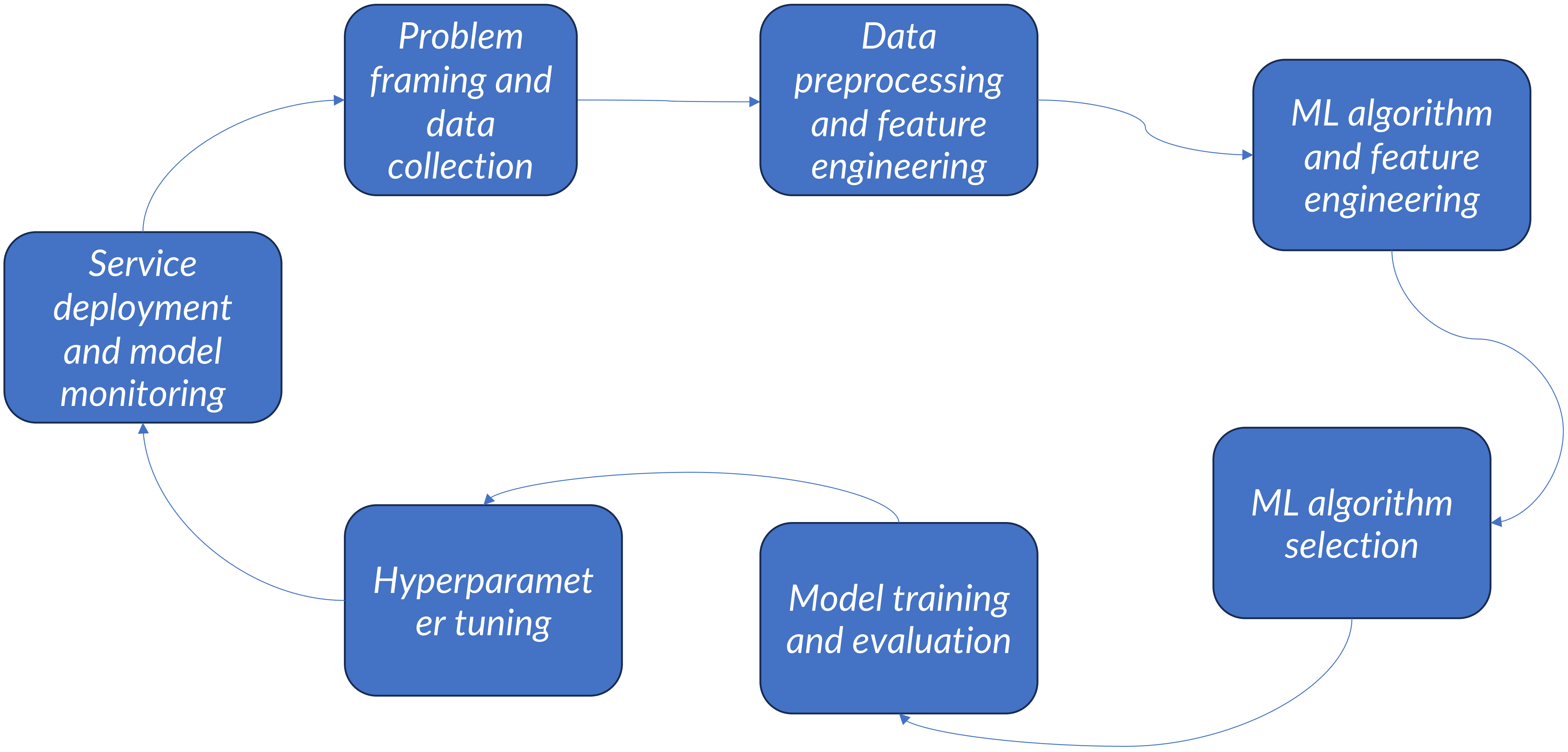
Background: „Resident”, a real estate owners association is considering aggresive investment campaign in one of Europe cities, but would like to select a city based on proper analyses. Short-rental (up to 3 days).

Goal: provide a report on how AirBnB prices in each city are shaped by internal and external factors; Recommend city where „Resident” should invests first

Audience: „Resident” board of directors

Data source: http://insideairbnb.com/get-the-data.html

Machine Learning Lifecycle



Machine Learning buzz words

Supervised vs **Unsupervised learning**

classification

„spam”, „not-spam”

regression

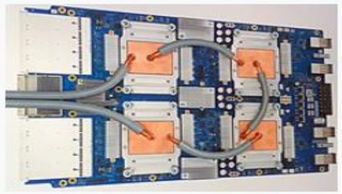
predicted value for sample is y

clustering

which data is similar to each other (assign label)

dimension reduction

describe data without lower numer ff dimension



Tensor Processing Unit 3.0

- Parallel processing
- Floating point arithmetics
- Memory bandwith

Graphics Processing Unit (GPU)

vs

Central Processing Unit (CPU)

Machine Learning buzz words

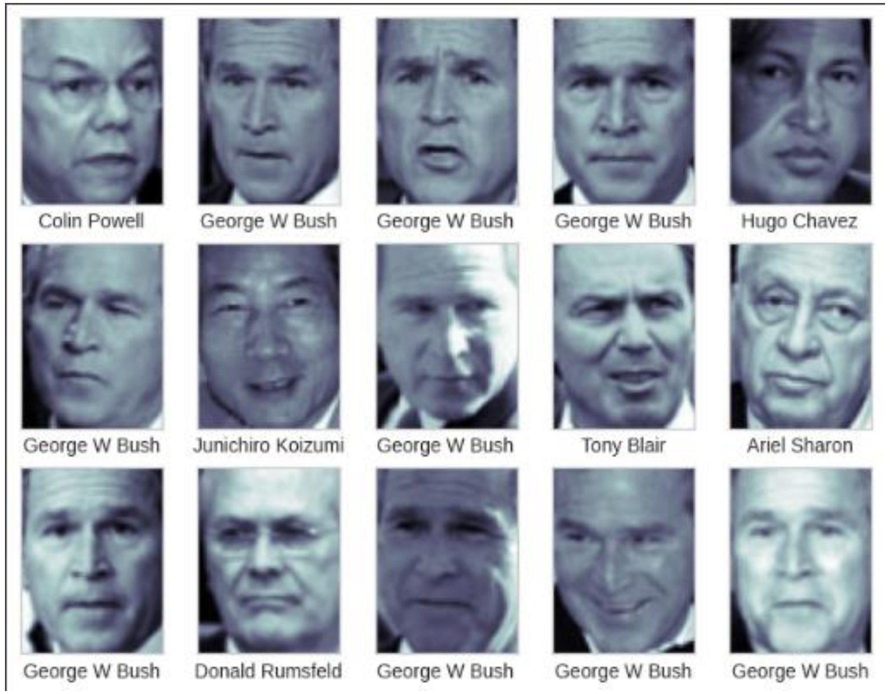
Noise Reduction

Dimension reduction

(Using Principal Component Analysis Component algorithm)

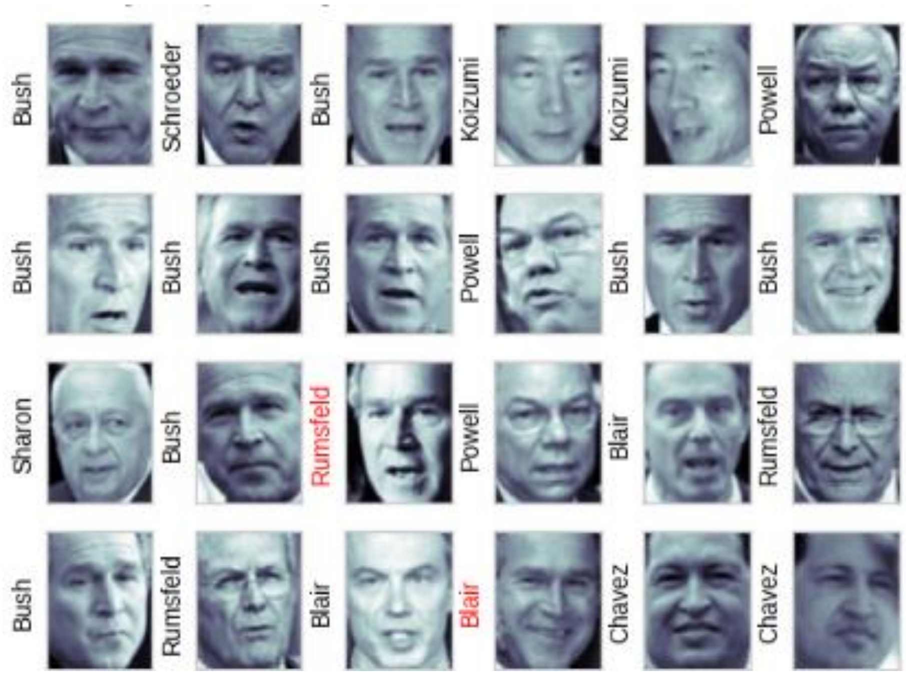
https://vis-www.cs.umass.edu/lfw/

Machine Learning buzz words



Training data

Classification



New data

Machine Learning buzz words

Clustering

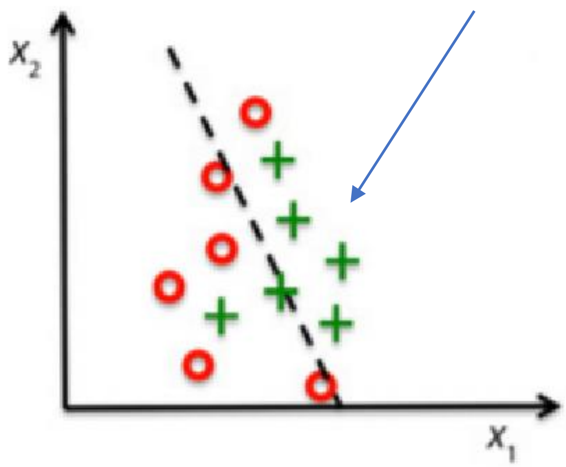
16 milions colors

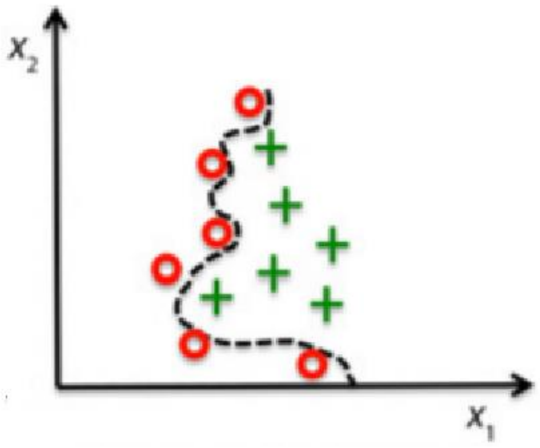
16 colors

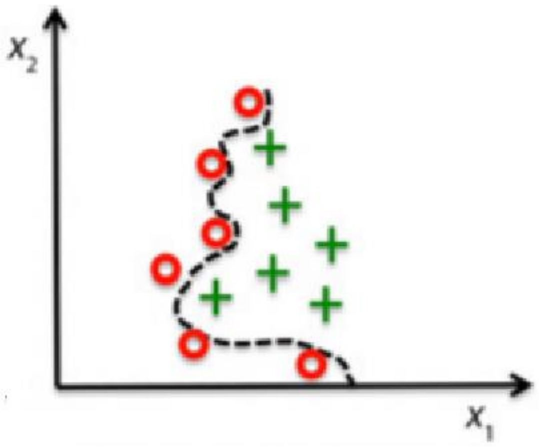
Compression ratio: 1mln !!!

Machine Learning – overfitting

Decision Line







Too little suited to the training data. Low prediction power for new data and training set.

Too tightly suited to the training data. High prediction accuracy for training data, but low for new data.

Good prediction accuracy for training data and new data.

Under-training

Overtraining

Life Example
During exam: Professor, on wich lecture slide was this topic discussed?

Good Fit

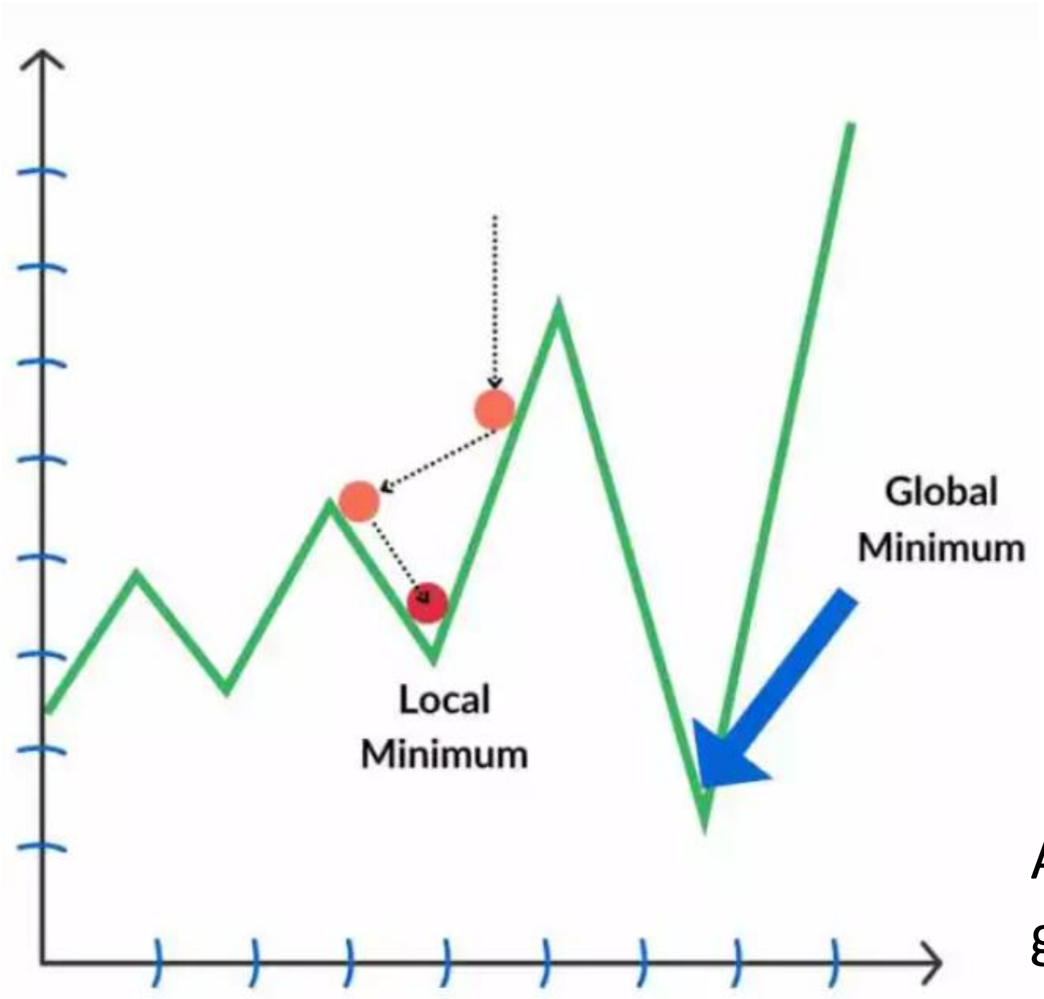
Global vs Local minimum during training

Important during training phase

Training Goal

Minimize loss function

Loss value



Iteration

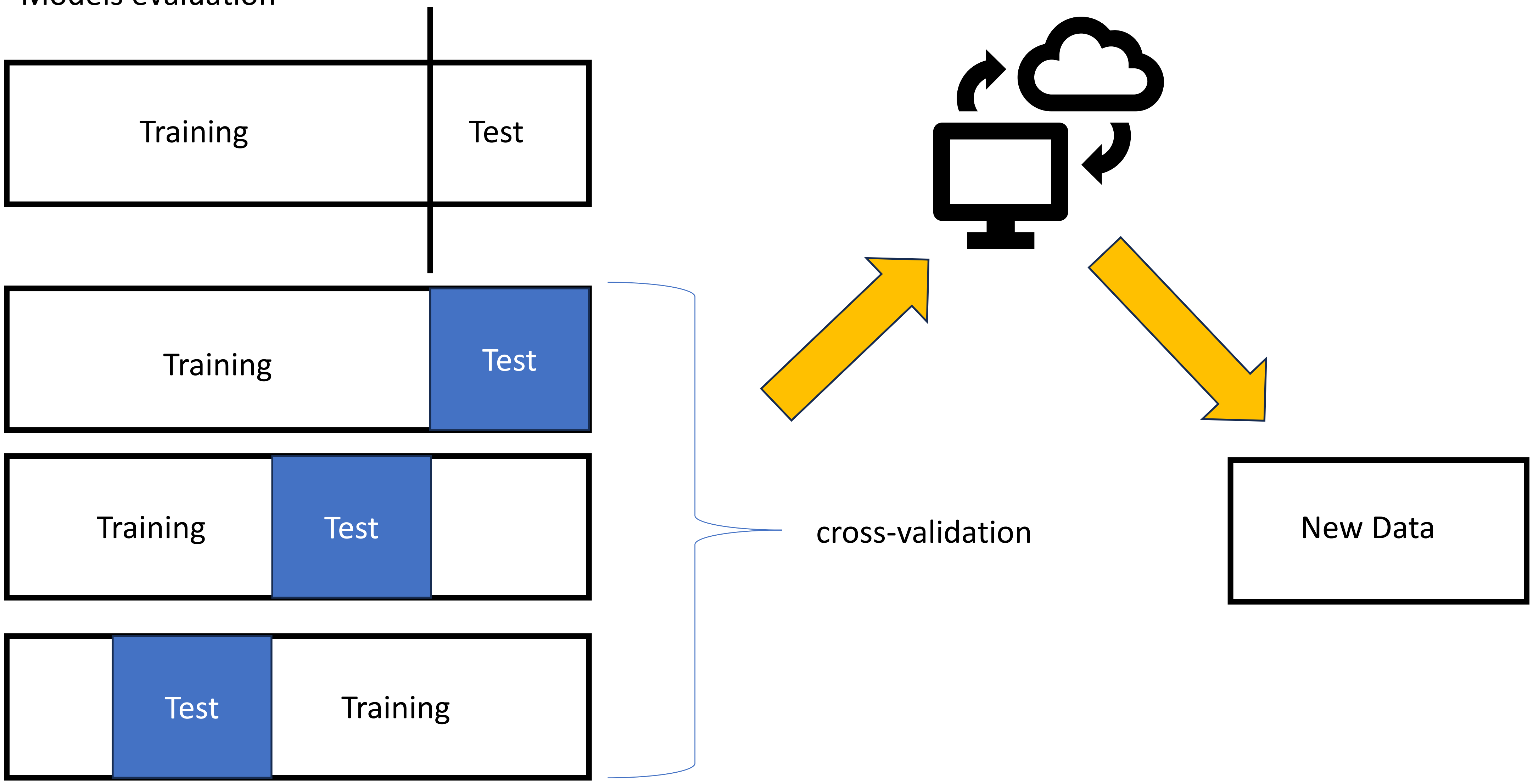
Solution

Stochastic Gradient Descend

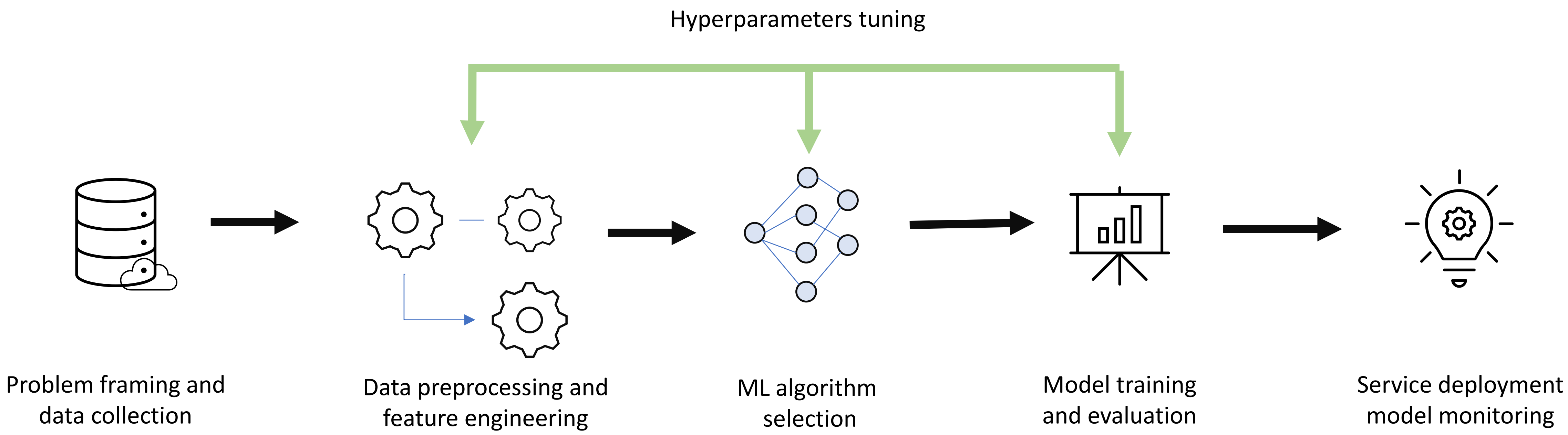
Add noise to gradient !

Machine Learning buzz words

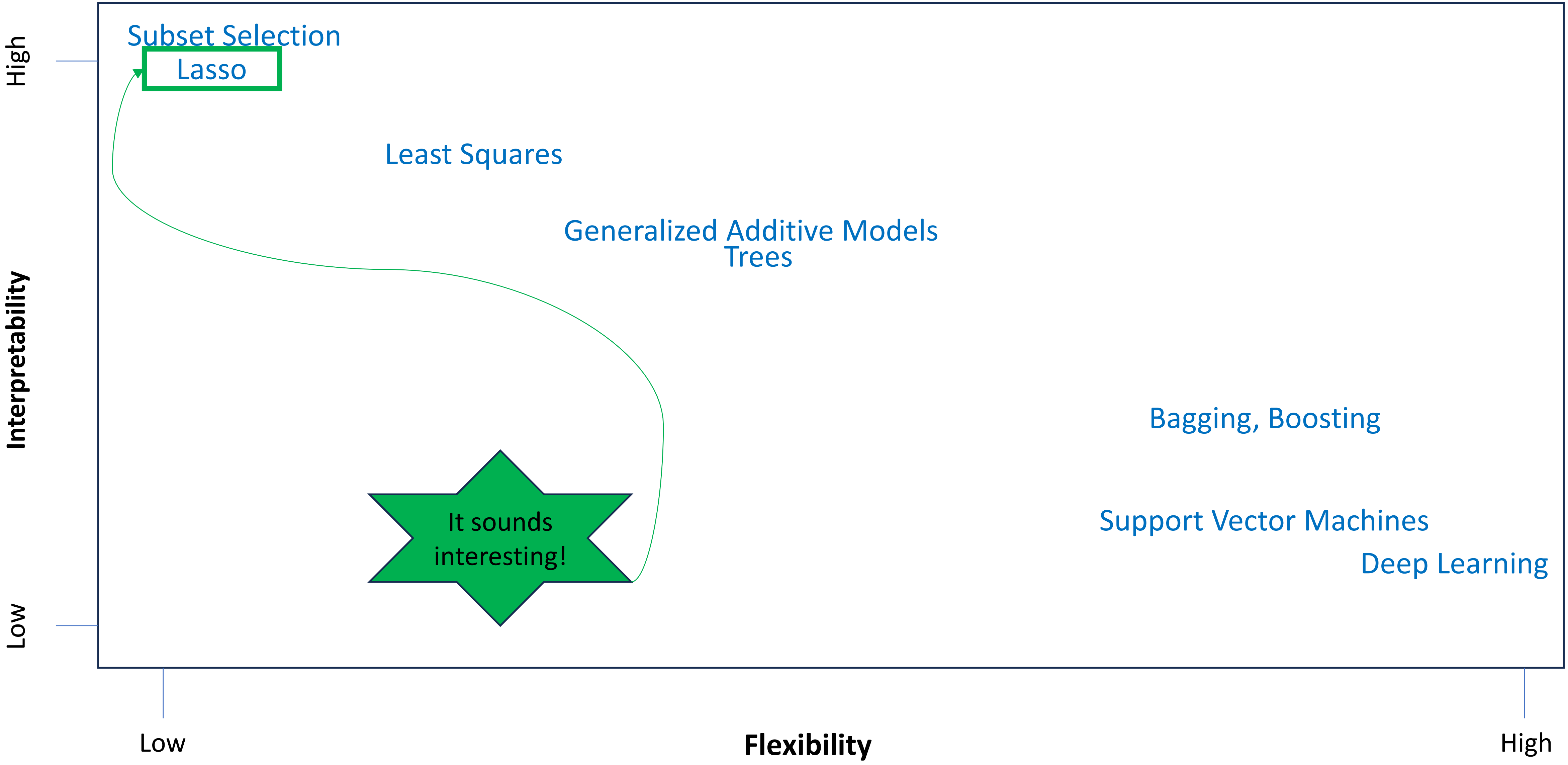
Models evaluation



Machine Learning Pipeline



Interpretability vs Flexibility



AutoML Goals

Automated Hyperparameters Tuning

Automated Pipeline Search

Automated Feature Engineering

Beneficial for newcomers

Fast prototyping ML models

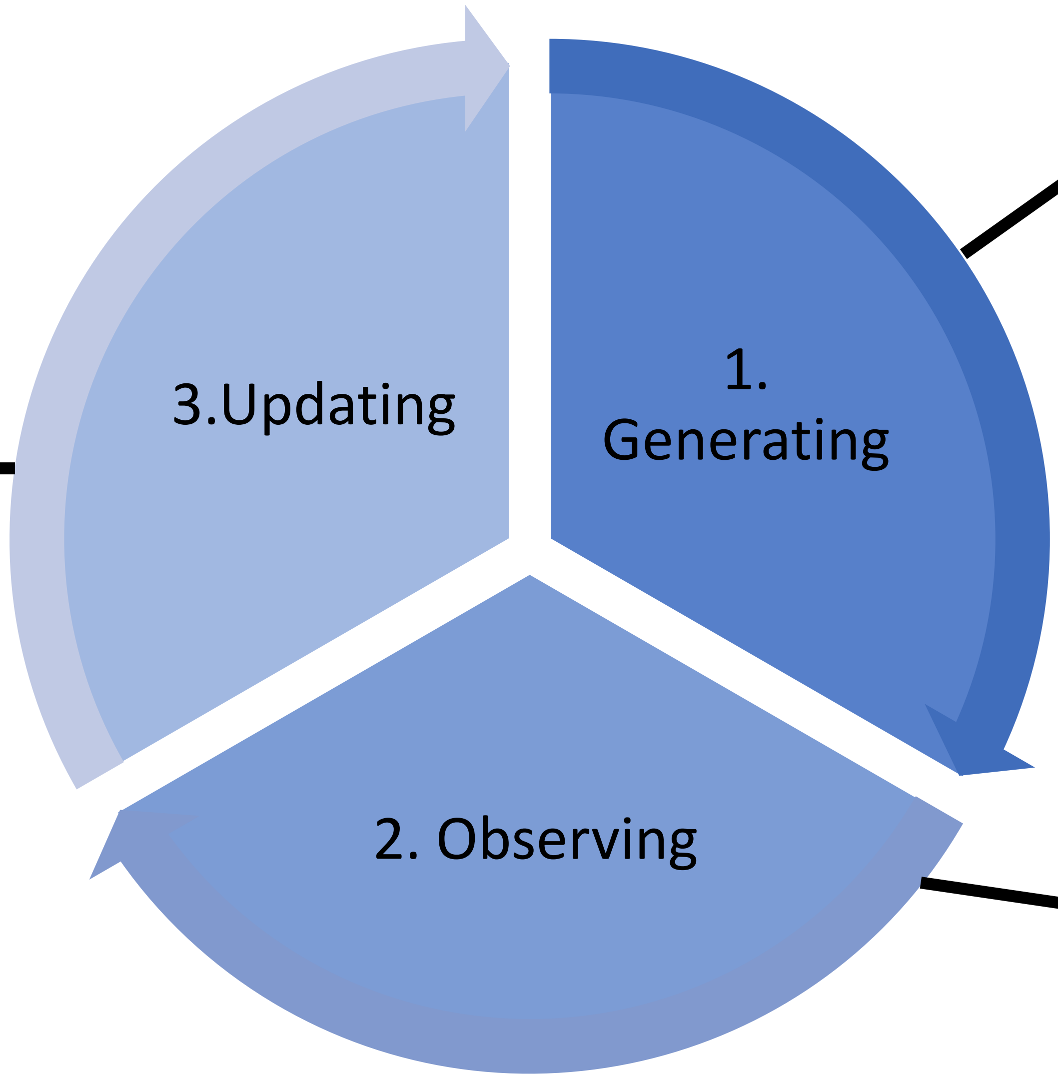
Limit the burden of ML models configuration

AutoML process

Select the next ML pipeline from the search space for observation based on the search algorithm

Search Strategy = oracle ☺

(Optional)
Update the search algorithm based on the evaluation if needed



Train and evaluate the selected ML pipeline on the dataset.

AutoML limitations

- There is no generic Tuner – differences in the model learning and validation process make it quite hard to combine shallow and deep models together
- Automatic Feature Engineering works better for shallow models where automatic hyperparameter tuning is not crucial for deep models
- No one perfect search strategy for find fast and the best hyperparameters sets.
- Warm-starting the search space could speed up Search Space but requires domain-specific knowledge
- Accuracy metric is only possible metric (edge devices or limiting memory consumption)
- Quite problematic interpretation and transparency
- Reproducibility – there are „hyper-hyperparameters" to control the search algorithm
- No free lunch – if you don't have typical ML task you have to perform extra work

OCI AutoMLx – What's new vs AutoML frameworks

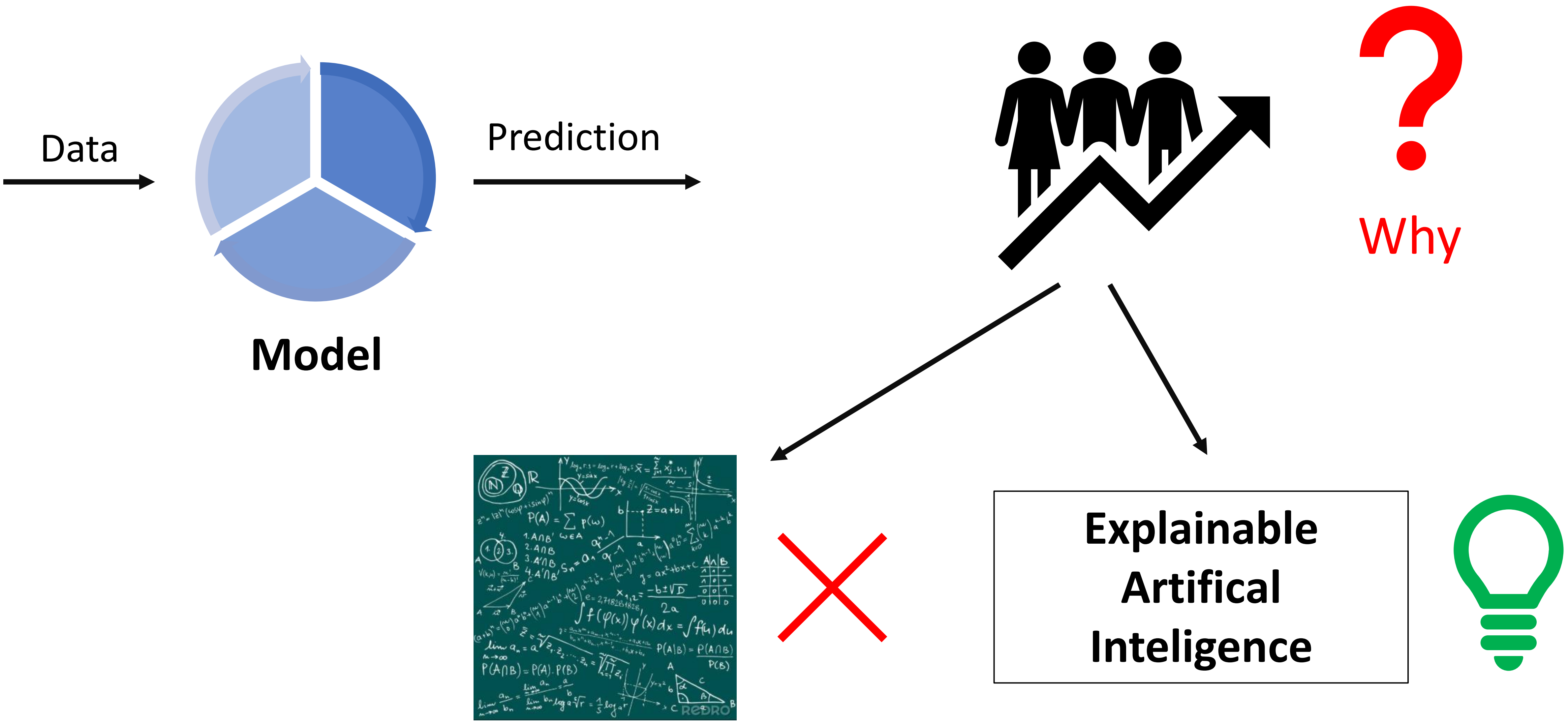
MLExplainer

- Visual and interactive explanations
- Quite hard to use

Fairness module

- Add addtional contraints to model (bias)
- Check againt bias and generate recommentation

Model Interpretability



Model Interpretability - SHAP

- XAI implementations for Python is ***shap*** package (by Scott Lundberg)
- Based on **Game Theory** („Beautiful Mind" – story of John Nash played by Russell Crowe)
- What is the **effect** on the model's prediction for each feature?

Training data Rows

Temperature	Relative humidity	Heat Index	Moisture Code	Area burned
23.1	97	22.95	3.0	57.9
11.3	33	18.37	106.7	0.0
8.2	51	22.59	26.2	0.0

Replace values from the dataset

Temperature	Relative humidity	Heat Index	Moisture Code
23.1	97	22.95	3.0
23.1	51	22.95	3.0
23.1	33	22.95	3.0

Predict

Predict

Predict

68.7
92.3
117.9

|68.7 – 92.3| = **23.6**

|68.7 – 117.9| = **49.2**

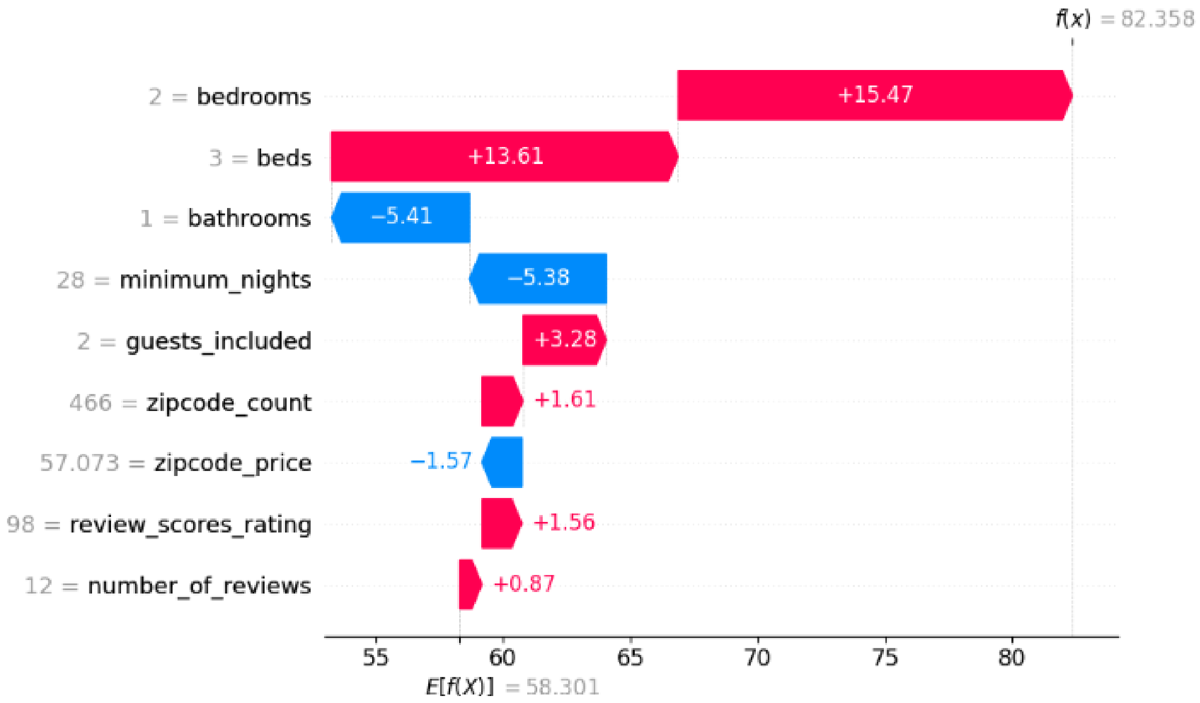
Marginal contribution calculations

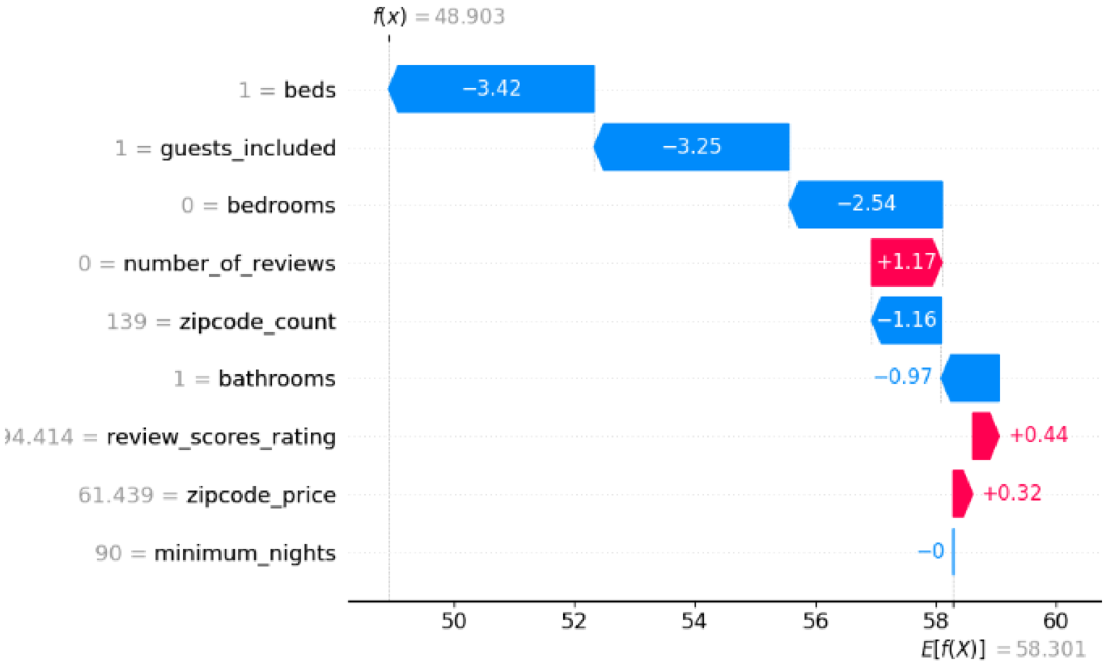
36.4

Relative contribution

Temperature	**Relative humidity**	**Heat Index**	**Moisture Code**
42	26	22	10

SHAP – Waterfall plot





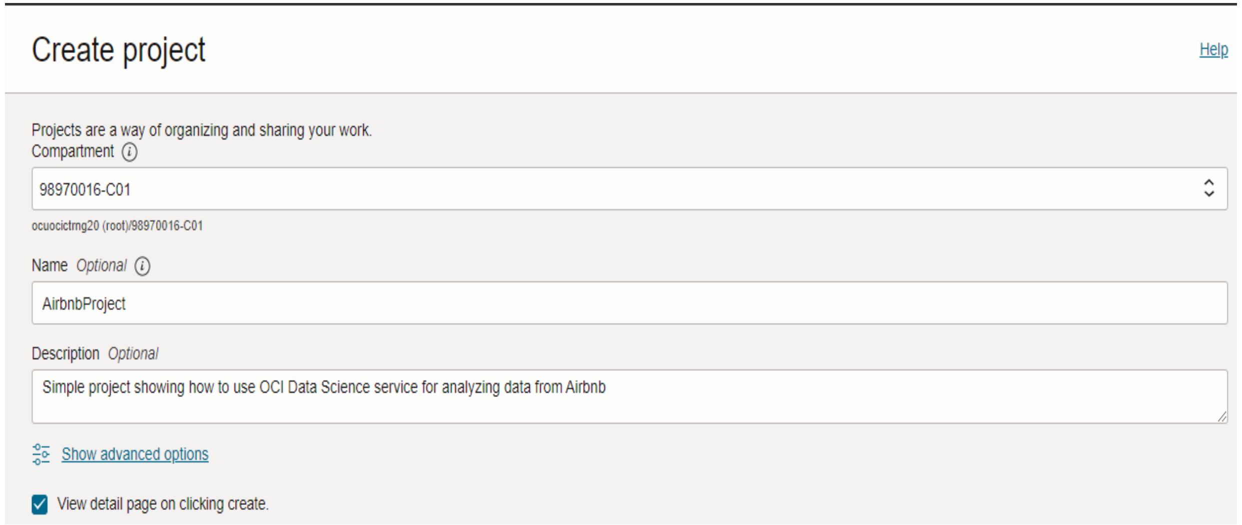
Lot of computing time to generate (1h for 30000x12 dataframe)

Use for find the pattern for top samples (here maximum price)

Not easy to find feature importance for entire dataset

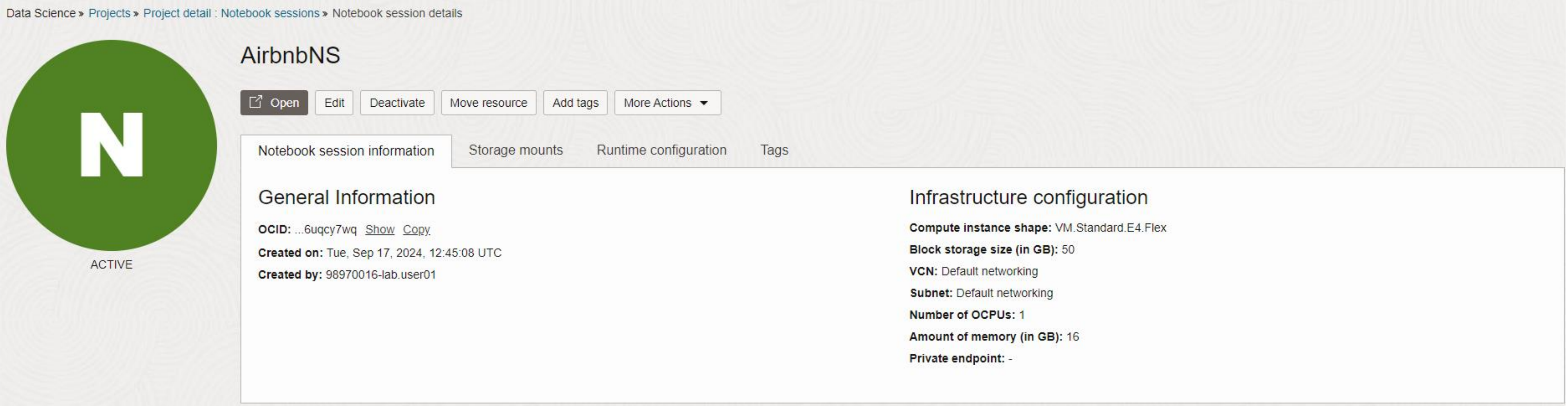
You need access to all data for new samples too

OCI Data Science Components

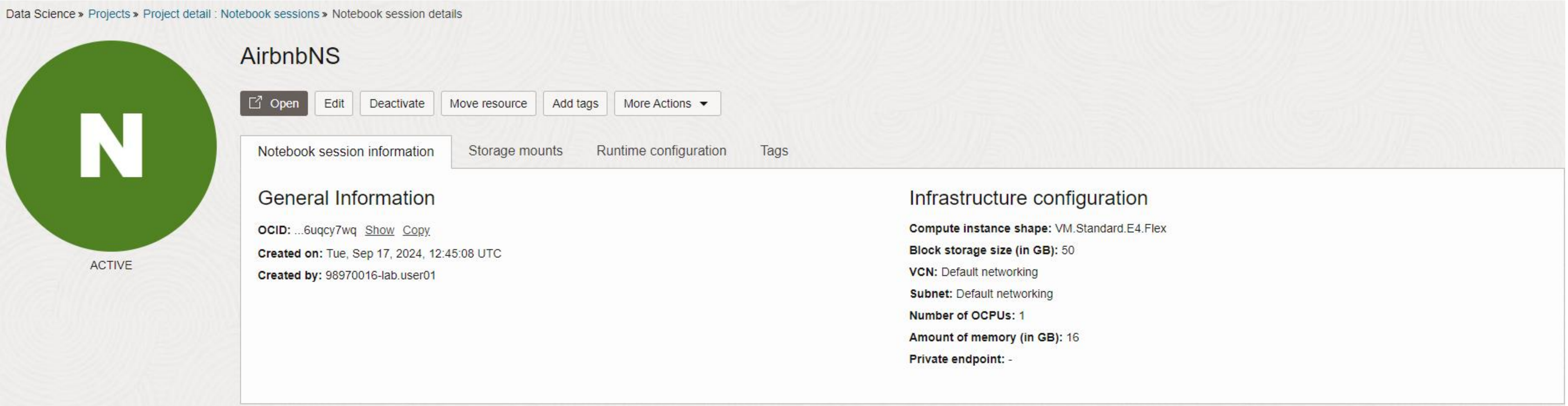


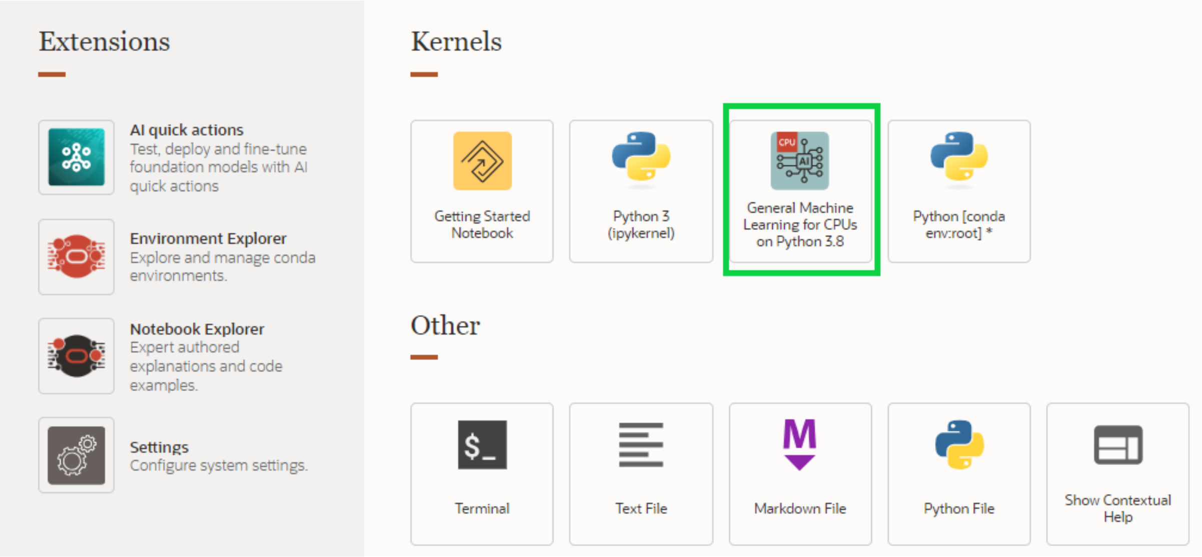
Project

Notebook Session



OCI Data Science Components

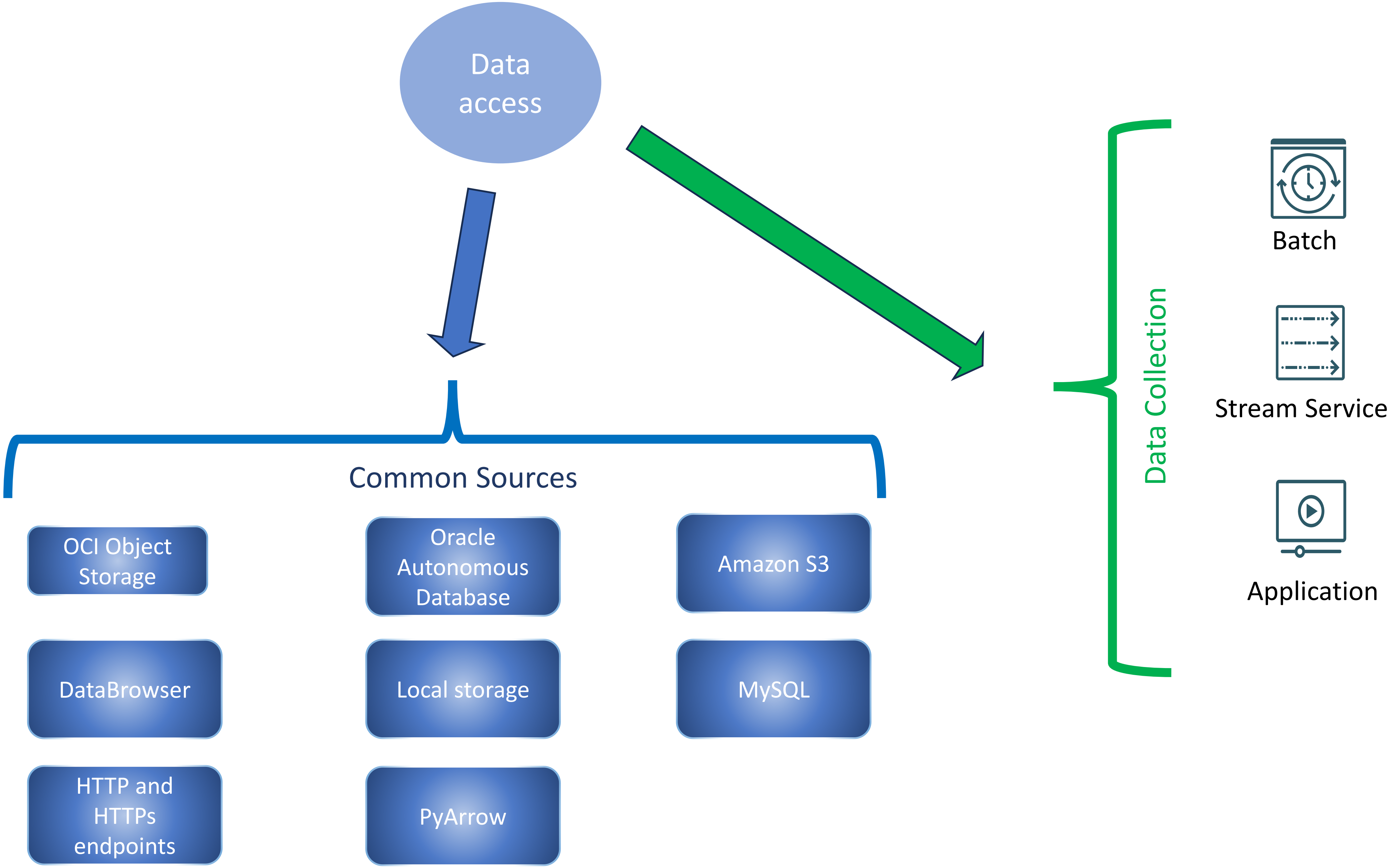




Notebook Session

Conda environment

Data Access


Data access
Common Sources
OCI Object Storage
Oracle Autonomous Database
Amazon S3
DataBrowser
Local storage
MySQL
HTTP and HTTPs endpoints
PyArrow
Data Collection
Batch
Stream Service
Application

Data Access

OCI Object Storage Buckets

Objects

Upload | More Actions

	Name	Last Modified	Size
	listings_berlin_11_2019.csv	Tue, Sep 17, 2024, 18:20:52 UTC	70.12 MiB
	listings_munich_11_2019.csv	Tue, Sep 17, 2024, 18:19:50 UTC	35.27 MiB
	listings_prague_11_2019.csv	Tue, Sep 17, 2024, 18:18:57 UTC	16.58 MiB

Buckets - most popular in practice (easy to integration, flexibily in price)

OCIFS – Oracle library to access OCI bucket in Python

OCI ADS library offers special class ADSDataSet – enabled recommended transformation and quicker data exploration features

```
berlinsrcfile = "listings_berlin_11_2019.csv"
munichsrcfile = "listings_munich_11_2019.csv"
praquesrcfile = "listings_prague_11_2019.csv"

data_berlin = pd.read_csv(f"oci://{bucket}@{namespace}/{berlinsrcfile}",
                          storage_options=default_signer())
print(f"Source file for Berlin: {berlinsrcfile}")
print(f"size {data_berlin.shape}")

data_munich = pd.read_csv(f"oci://{bucket}@{namespace}/{munichsrcfile}",
                          storage_options=default_signer())
print(f"Source file for Munich: {munichsrcfile}")
print(f"size {data_munich.shape}")

data_praque = pd.read_csv(f"oci://{bucket}@{namespace}/{praquesrcfile}",
                          storage_options=default_signer())
print(f"Source file for Praque: {praquesrcfile}")
print(f"size {data_praque.shape}")
```

pandas

oracle/accelerated-data-science

Data Exploration and Preparation

Data exploration and preparation

Very time consuming task in ML workflow

What to do with missing values, strongly correlated data, imbalanced data

Data visualization used for understanding data better

Is there a way to enrich the data?

Oracle-ADS library helps includes two magic functions which helps a lot with phase in ML lifecycle.
One-liners ☺☺☺

Data Exploration and Preparation (EDA)

"The only way humans can do BETTER than computers is to take a chance of doing WORSE than them."

80 – 90%

Why does this stage matter? → Avoiding the GIGO symptom

Goal and strategy

→ Clean and transform data for optimal learning algorithm

→ Understanding datasets by using summary statistics and visualization

EDA subtasks

- Complete observations or mark missing cases by appropriate features
- Transform text or categorical variables
- Create new features based on domain knowledge of the data problem
- Have at hand a numeric dataset where rows are observations and columns are variables
- Describe of your data
- Closely explore data distributions
- Understand the relations between variables
- Notice unusual or unexpected situations
- Place the data into groups
- Notice unexpected patterns within groups
- Eliminate outliners

Airbnb project - Data Enrichment

- Enrich data by calculate zip code relation to price
- **amenities_len** -> value for comparing number of amenities across offers
- **zip_count** -> number of offers in close location
- **zip_price** -> average price for offers in the same location

5%

Airbnb project - EDA

```
# Enrich data by calculate zip code relation to price
# amenities_len -> value for comparing number of amenities across offerts
# zip_count -> number of offerts in close location
# zip_price -> average price for offerts in the same location

temp_zipcode = data_berlin.zipcode.copy()
data_berlin['zipcode2'] = temp_zipcode.str.replace("\D+", "", ).copy()
data_berlin.zipcode2.fillna(0, inplace=True)
x_count = data_berlin.groupby('zipcode2')['id'].nunique()
x_mean = data_berlin.groupby('zipcode2')['price'].mean()

x_count_dict = x_count.to_dict()
x_mean_dict = x_mean.to_dict()

a1 = np.zeros((len(data_berlin), 6))
print(a1)
for i in range(0,len(data_berlin)):
    val = data_berlin.zipcode2[i]
    a1[i][0] = data_berlin.id[i]
    a1[i][1] = x_count_dict[val]
    a1[i][2] = x_mean_dict[val]
    a1[i][3] = val
    a1[i][4] = len(data_berlin.amenities[i])

data_berlin['amenities_len'] = a1[:,3]
data_berlin['zipcode_count'] = a1[:,1]
data_berlin['zipcode_price'] = a1[:,2]
print(data_berlin.head())
```

Create new variables => data enrichment

Airbnb project - EDA

```
# Preprocessing - replace NaN values with mean from column
# With checking before and after replacement

print(data_berlin[cols].isna().sum())
data_berlin.fillna((data_berlin[cols].mean()), inplace=True)
print(data_berlin[cols].isna().sum())
```

<u>Replace NaN values with means</u>

Airbnb project - EDA

```
# Preprocessing - get rid of outliers
print("99.7% properties have a price lower than {0: .2f}".format(np.percentile(data_berlin.price, 99.7)))
data_berlin = data_berlin[(data_berlin.price <= np.percentile(data_berlin.price, 99.7)) & (data_berlin.price > 0)]
```

<u>Get rid of outliners</u>

Data Exploration and Preparation

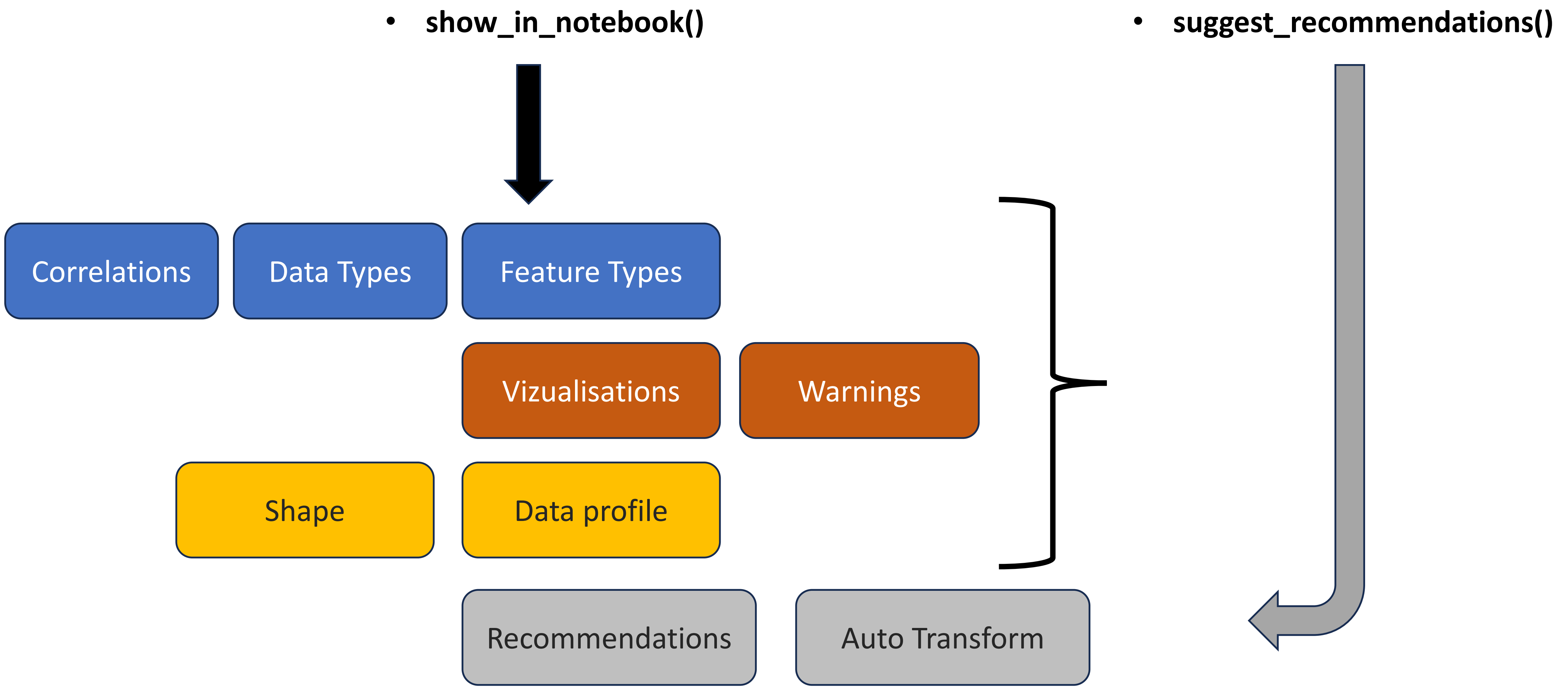
- **show_in_notebook()**
- **suggest_recommendations()**

- Oracle-ADS helps with automation – suggested recommendations and applying if needed
- one-liners – do more with less code
- Data profiling and visualisation

- Loading/Converting a dataset with Oracle-ADS DataFactory can be much slower than simple reading Pandas dataframe
- You are blocked if Oracle-ADS estimates allocated resources in OCI for processing are not enough

Not attempting to calculate correlations, too few cores (2) for wide dataset (96 columns) berlin_ds.suggest_recommendations()

Data Exploration and Preparation



Data Exploration and Preparation

- show_in_notebook()

```
[*]: berlin_ds.show_in_notebook()
```

▸ Summary

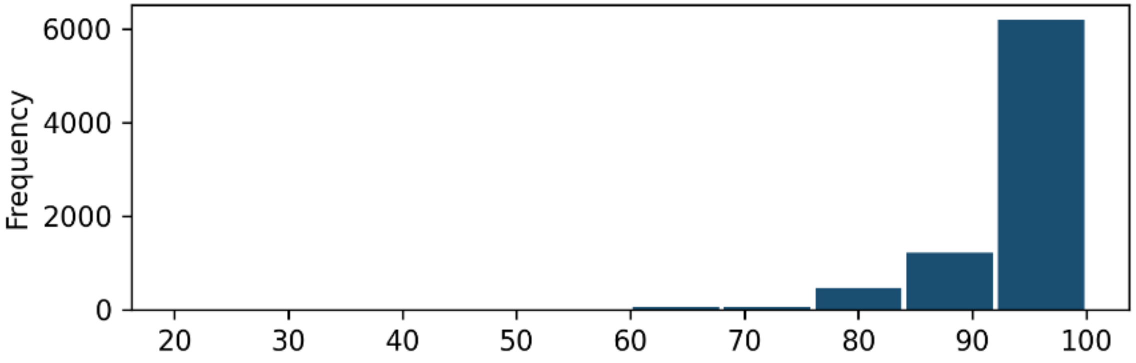
▾ Features (9)

• **Note** these are computed on the entire dataset.

	count	unique	top	freq	mean	std	min	25%	50%	75%	max	missing	skew
host_is_superhost	22526	2	f	19515								26	
accommodates	22552				2.64	1.51	1	2	2	3	16	0	2.6024944
bathrooms	22520				1.09	0.33	0	1	1	1	8.5	32	6.3385291
bedrooms	22534				1.16	0.65	0	1	1	1	12	18	2.3882602
beds	22512				1.62	1.17	0	1	1	2	22	40	3.7055501
price	22552				67.14	220.27	0	30	45	70	9000	0	26.733229
minimum_nights	22552				7.16	40.67	1	2	2	4	5000	0	85.888045
number_of_reviews	22552				17.84	36.77	0	1	5	16	498	0	4.3829585
review_scores_rating	18163				94.41	7.64	20	92	97	100	100	4389	-3.2790174

review_scores_rating

- type: ordinal (float64)
- missing_percentage: 19.3%
- ordinal statistics:
  - unique percentage: 0.521%
  - x_min: 20
  - x_max: 100
  - mode: 100
  - count: 8,066
  - unique: 42
  - top: 100
  - freq: 2,700



Data Exploration and Preparation

- show_in_notebook()

Warnings (4)

4 WARNING(S) found

missing

`review_scores_rating` has 4389.0 (19.5%) missing values. Consider remove the column or replace null values.

`price` has skew of 26.733

skew

`minimum_nights` has skew of 85.888

skew

`number_of_reviews` has 3890 (17.25%) zeros)

zeros

Data Exploration and Preparation

- suggest_recommendations()

```
[15]: berlin_ds.suggest_recommendations()
```

[15]:

Message	Variables	Suggested	Action	Code
Contains missing values(12)	host_is_superhost	Fill missing values with frequent	Drop	.drop_columns(["host_is_superhost"])
			Fill missing values with frequent	.fillna({"host_is_superhost": "f"})
			Fill missing values with constant	.fillna({"host_is_superhost": "constant"})
			Do nothing	
Contains missing values(9)	bathrooms	Fill missing values with mean	Drop	.drop_columns(["bathrooms"])
			Fill missing values with mean	.fillna({"bathrooms": 1.0876})
			Fill missing values with median	.fillna({"bathrooms": 1.0})
			Fill missing values with frequent	.fillna({"bathrooms": 1.0})
			Fill missing values with constant	.fillna({"bathrooms": "constant"})
			Do nothing	
Contains missing values(8)	bedrooms	Fill missing values with frequent	Drop	.drop_columns(["bedrooms"])
			Fill missing values with frequent	.fillna({"bedrooms": 1.0})
			Fill missing values with constant	.fillna({"bedrooms": "constant"})
			Do nothing	
Contains missing values(12)	beds	Fill missing values with frequent	Drop	.drop_columns(["beds"])
			Fill missing values with frequent	.fillna({"beds": 1.0})
			Fill missing values with constant	.fillna({"beds": "constant"})
			Do nothing	
Contains missing values(19.34%)	review_scores_rating	Fill missing values with frequent	Drop	.drop_columns(["review_scores_rating"])
			Fill missing values with frequent	.fillna({"review_scores_rating": 100.0})
			Fill missing values with constant	.fillna({"review_scores_rating": "constant"})
			Do nothing	
Strongly correlated with beds(79.06%.)	accommodates	Drop beds	Drop accommodates	.drop_columns(["accommodates"])
			Drop beds	.drop_columns(["beds"])
			Do nothing	
Imbalanced Target(0.16%)	price	Do nothing	Do nothing	
			Down-sample	.down_sample()
			Up-sample	.up_sample(sampler='default') \n `pip install imbalanced-learn` to use default up-sampler.

Data Exploration and Preparation

- **suggest_recommendations()**

Manual

- ➢ We could choose which recommendations to implement
- ➢ Code provided for implementation

Automatic

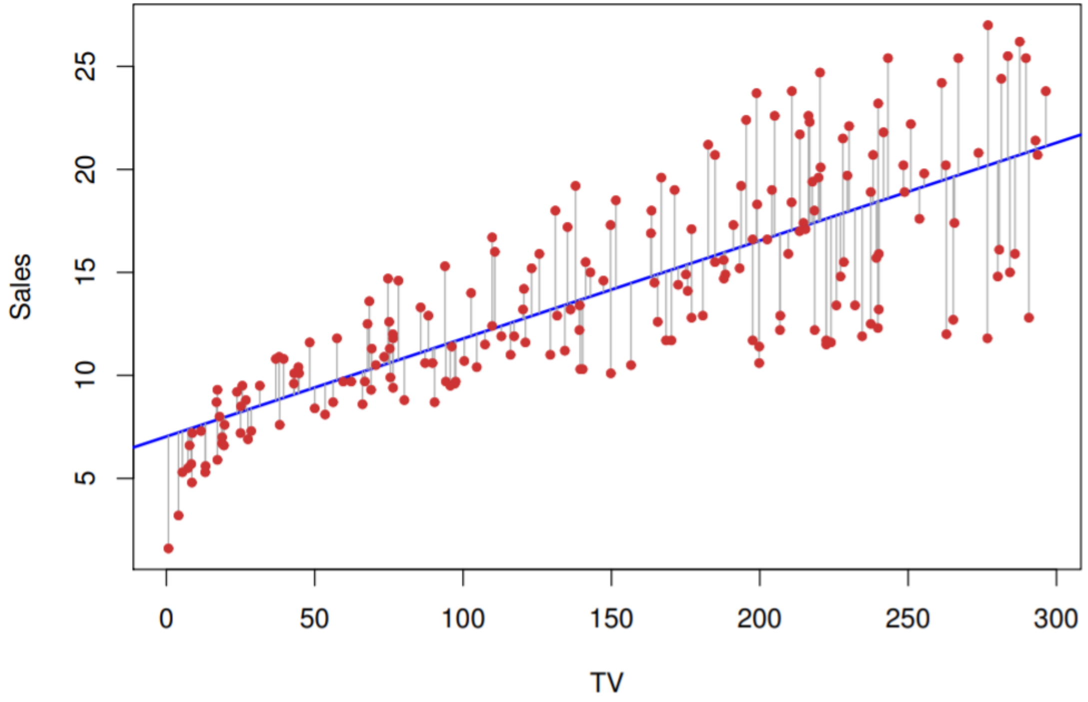
- ➢ Oracle-ADS provides auto_transform function
- ➢ All recommendation implemented (actions suggested)
- ➢ Pick up recommendations – not possible

Modeling

Validation

Modeling & Validation

Training Process



Find the best algorithms – maximize prediction / interpretaion / performance / resource usage / time to response

Find the best parameters for algorithm – they are called hyperparameters

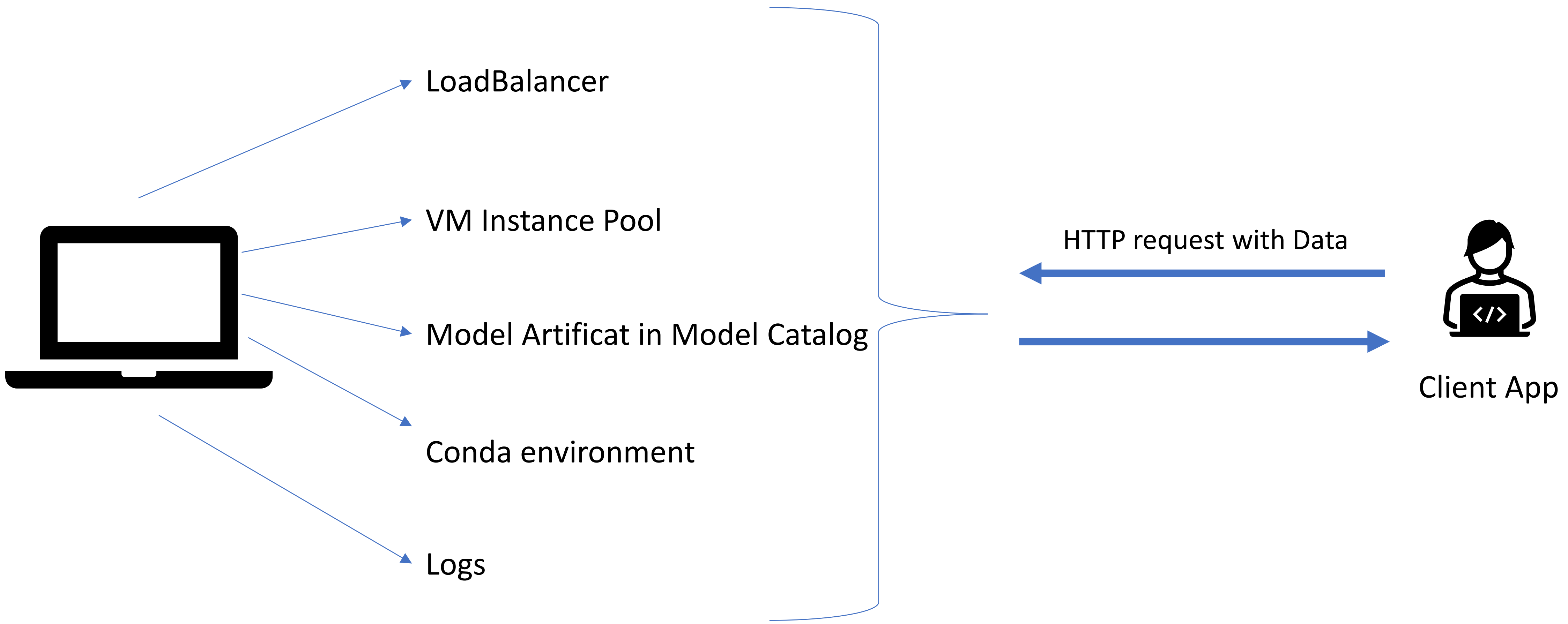
Interactive (within Notebook Session)

Batch/Job

Deployment

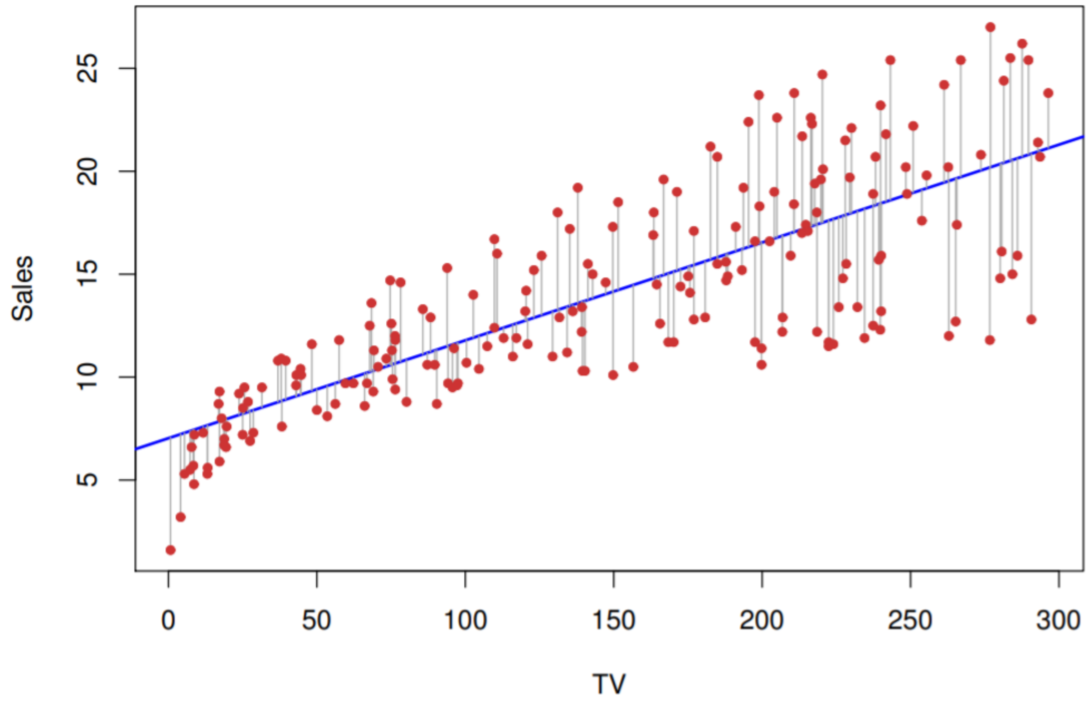
Deployment & Monitoring

Monitoring

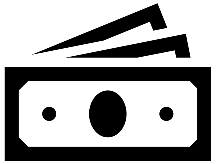


Regression analysis – Quick Theoretical Guide

Belongs to supervised category of machine learning algorithm.
Unlike classification (think email -> *spam or not spam,* categories predefined, binary) – the second subcategory of supervised learning – regressive analysis is used to predict data continuously rather than through categorization class labels.



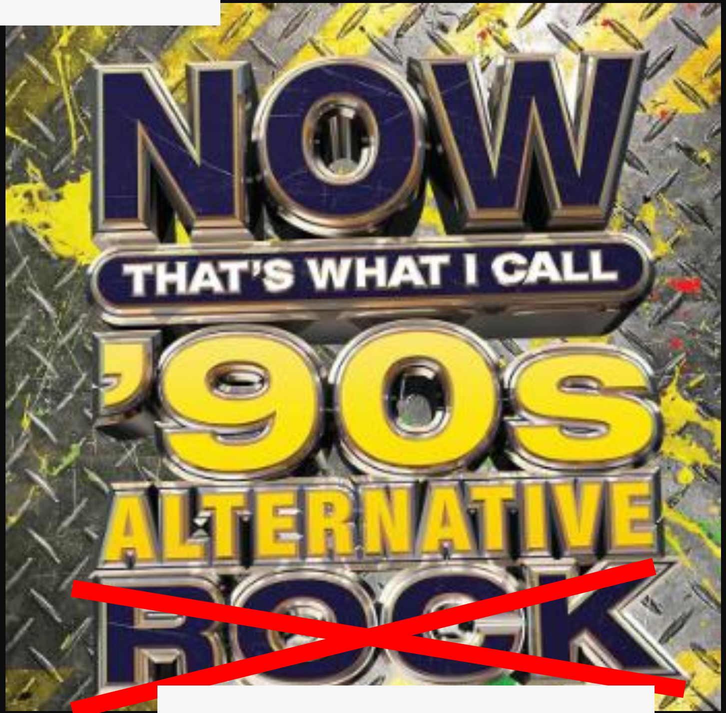
What will be the sales volume if we invest $400000 in television advertising?



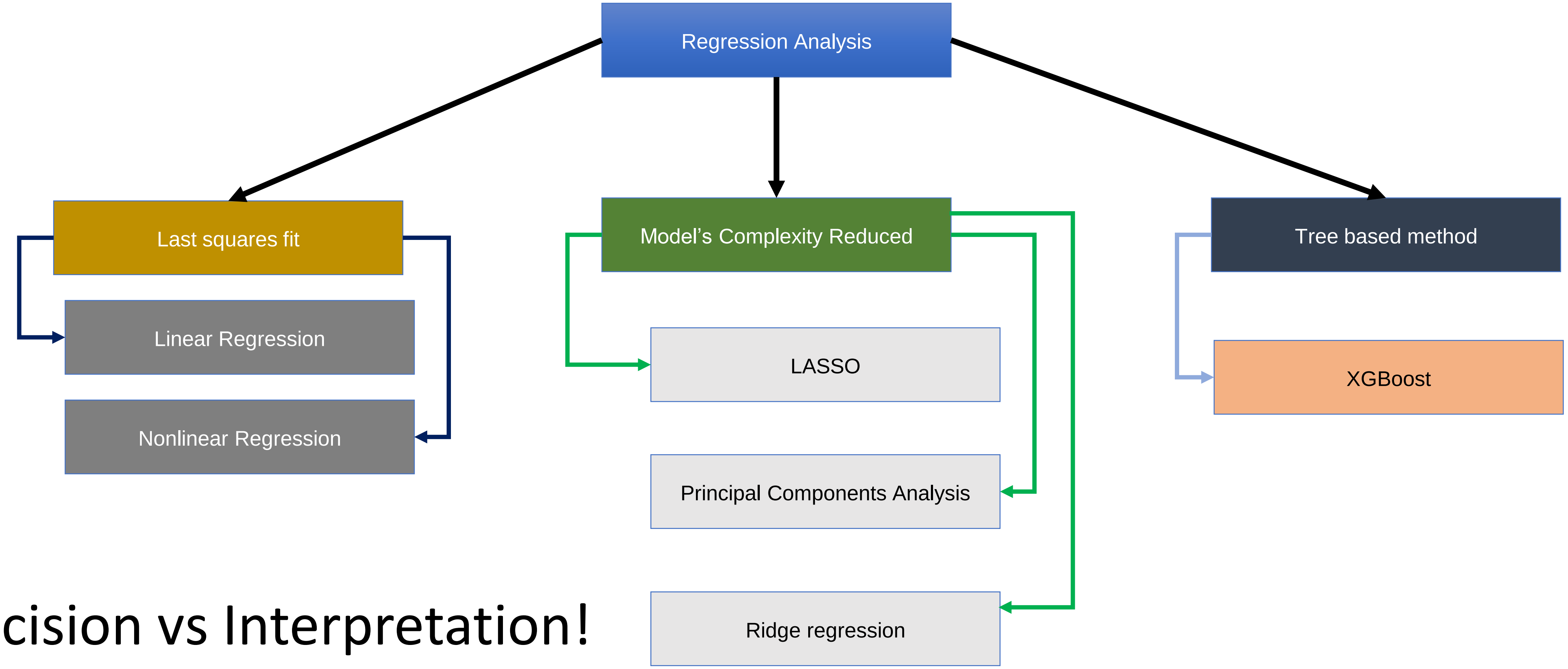
Regression analysis – Quick Theoretical Guide

Find values of coefficients which fit best for training data and achieve best accuracy for new data

Find coefficient(s) that has the greatest impact on the explained variable = change of its value change explained variable the most



Regression Analysis – Algorithms by category



Precision vs Interpretation!

LASSO algorithm – that's we used

$$J(w)_{LASSO} = \sum_{i=1}^{n} \left(y^{(i)} - \hat{y}^{(i)} \right)^2 + \lambda \|w\|_1$$

Goal: Minimize Cost Function

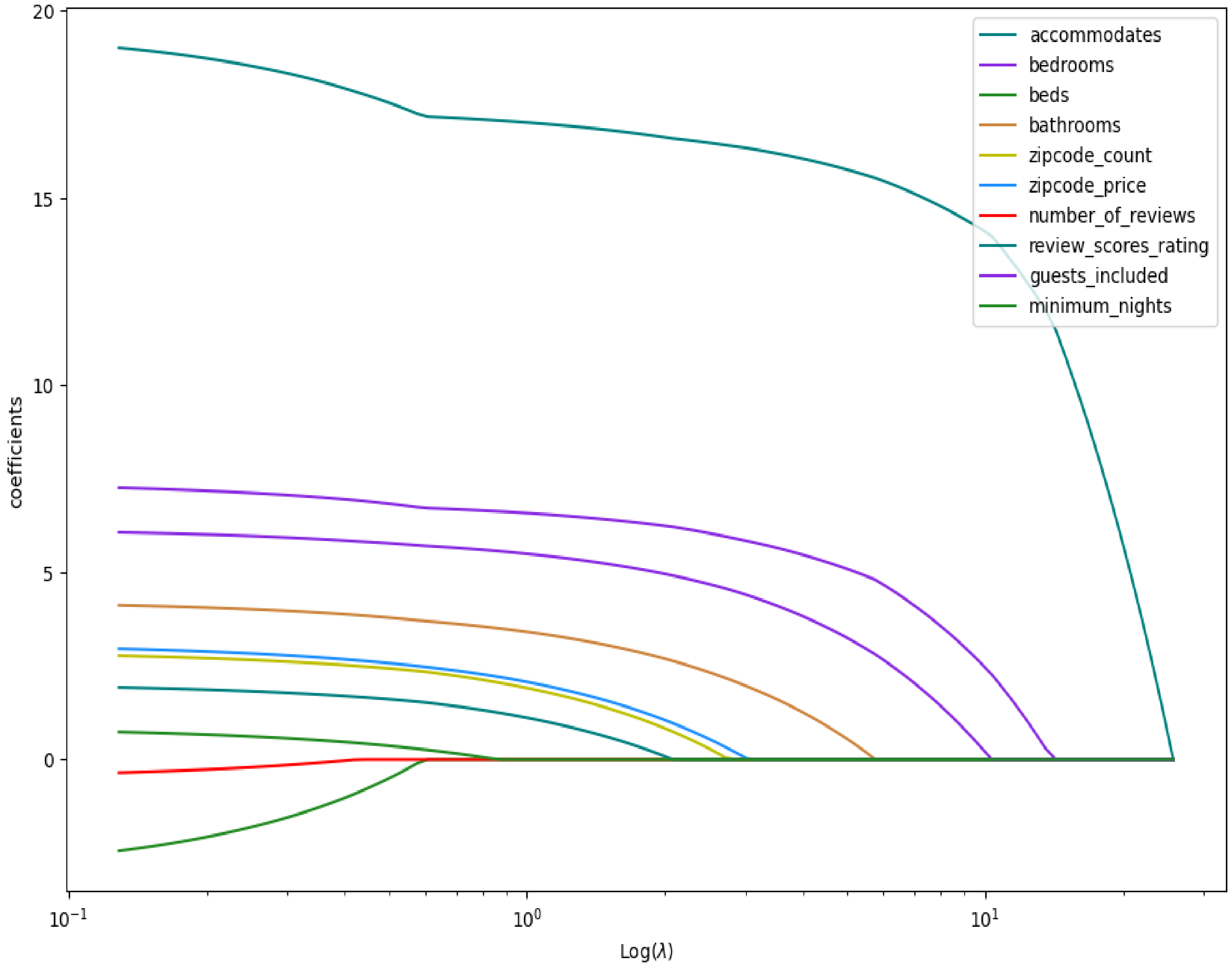
$$\text{L1:} \quad \lambda \|w\|_1 = \lambda \sum_{j=1}^{m} |w_j|$$

- Includes regularization part as response to overfitting problem
- Regularization introduces penalty for complex model (mathematically = it tries to avoid large values of coefficients)
- We use it when we have a lot of variables, we don't really know which variables we want to use (for example - no domain knowledge)

LASSO algorithm – that’s we used

- **Lasso** - newer and better method than “classical” linear regression (origins in XVII century)
  - we are performing linear regression, but we add additional element -> penalty added to list of coefficients
  - optimization goal is minimization of cost function J; LASSO enforces that absolute values of coefficients must be less than LAMBDA parameter
  - lower LAMBDA value then more coefficient will be zero so they will be eliminated from model
  - If coefficient is sooner eliminated from model, then less “impact” it has on explained variable
  - LASSO results could be very nicely visualized (“positive” impact and “negative” impact) are possible

LASSO algoritm – practice



It shows how short-term price rent is related with various variables and value of LAMBDA parameter
(here is log(LAMBDA) is used on X axis for better visualization experience’)

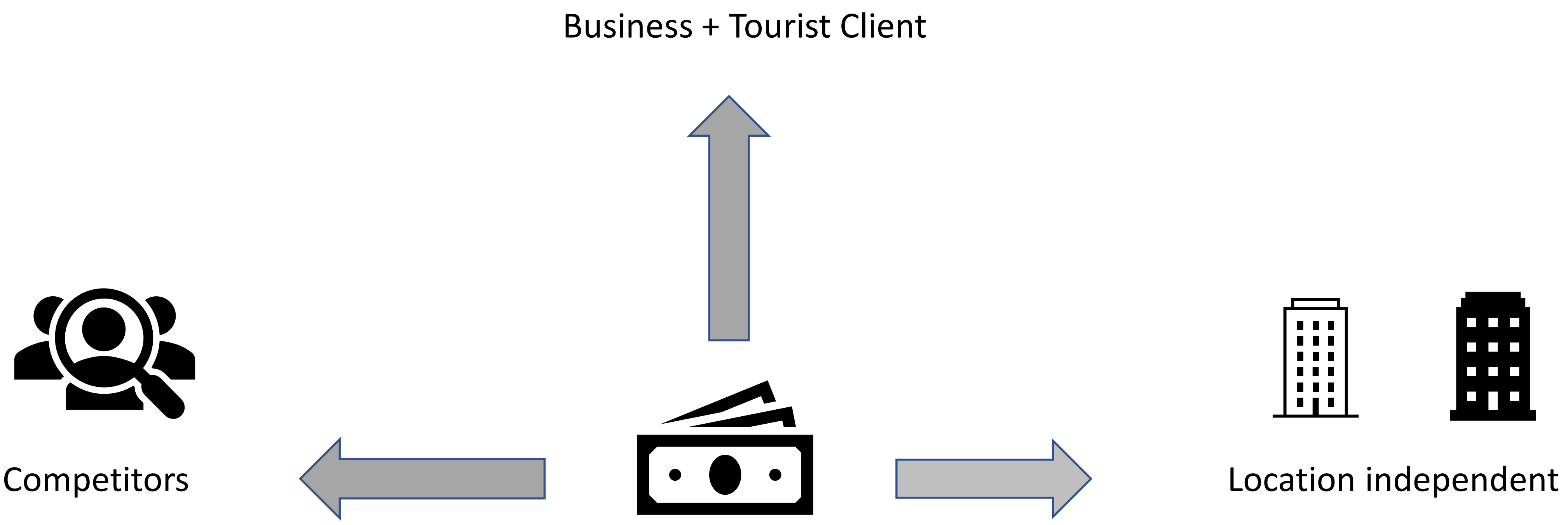
Lower LAMBDA value then sum of coefficients. The coefficient with larger impact on price dies last.

Here for example, we see ***number of bedrooms*** has most impact on price and rating has negative impact on price (lower rating then lower price), but it has least impact on price among all variables provided.

Airbnb project -> Data Analysis

- Comparing 3 cities -> Berlin, Monachium i Praga
- A small percentage of rejected values in each city (0.03%)
- Praque – significant impact of the numer of comments on the rental price
- Munich + Prague – greate influence of the location on the rental price
- "Superhost" badge – marginal impact on rent price
- Slightly affected by the number of bathrooms in the apartment

Airbnb project -> Recommended city - Berlin



Airbnb -> Recommendations

- Berlin is the best place for investments

- Strategy „Buy metres, rent rooms”

- We propose that the next step should be to enrich the analysis with economic information (inflation, prices of repair services, purchase prices, maintenance costs, taxes, possible length of long-term rental)

Airbnb project -> Used methods and algorithms

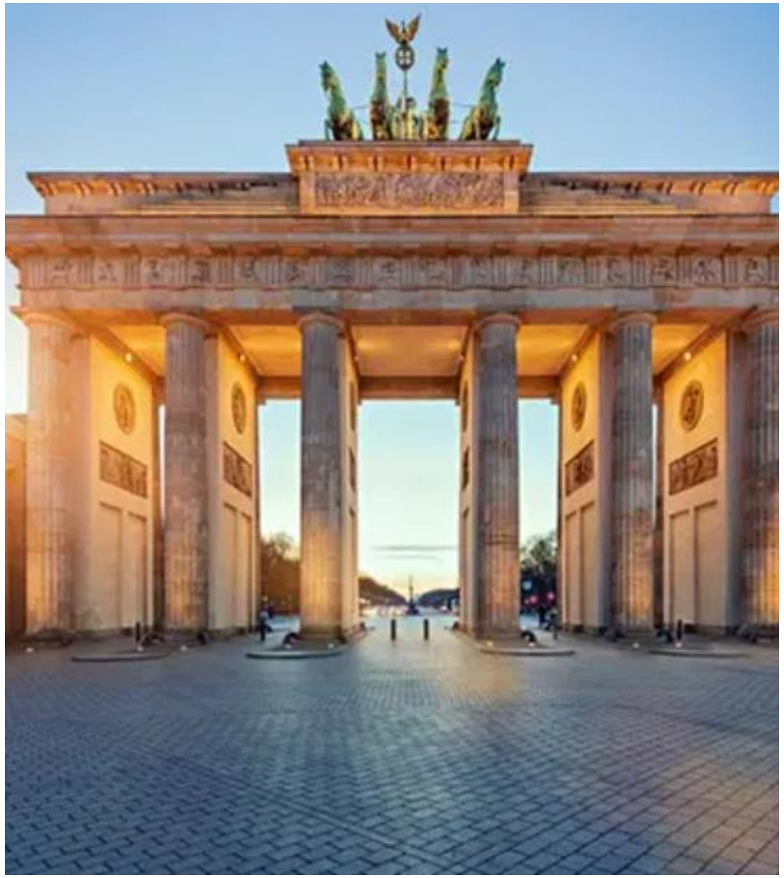
- Metrics for comparing regression models: P-value, R2, AIC
- Regulated regression algorithms : LASSO

Airbnb project -> Executive Summary

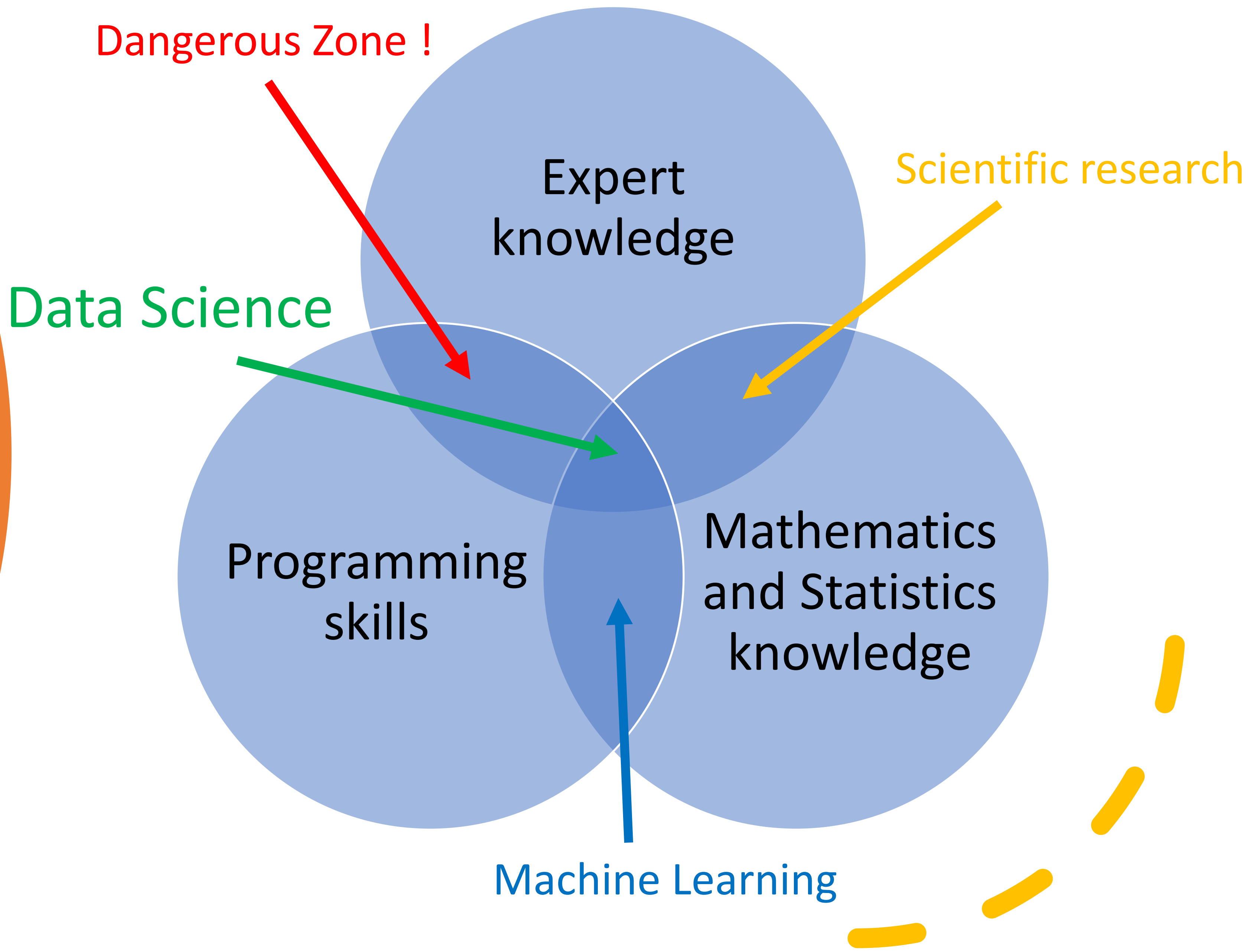
- Berlin is the best city for investments from among the selected ones (Berlin, Munich and Prague)

- Maximizing sleeping places in a good strategy to increase profits

- Berlin has the most diffuse competition, allowing smaller players entering the market



Summary I


Dangerous Zone !
Expert knowledge
Scientific research
Data Science
Programming skills
Mathematics and Statistics knowledge
Machine Learning

Summary II

- OCI ADS framework helps with time consuming task in Machine Learning Lifecycle
- Integrates easily with OCI services
- Data Science is interdisciplinary domain
- Machine Learning is not only for prediction cases; sometimes it turns into interpretation cases
- OCI benefit - total cost was 8.24 USD for whole work

Thank you ☺ !