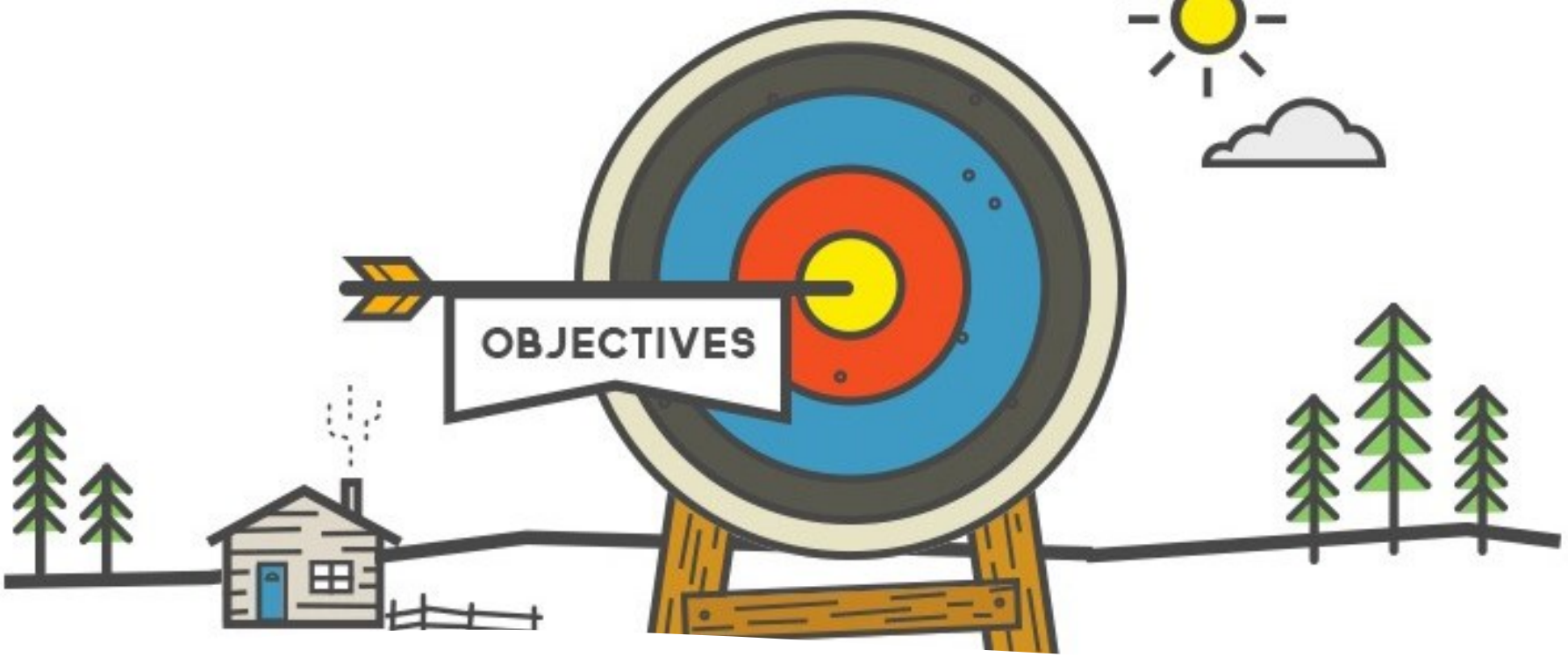
Analytical Techniques

Prof. Carlos J. Costa, PhD

Saeed Angorani, DBA



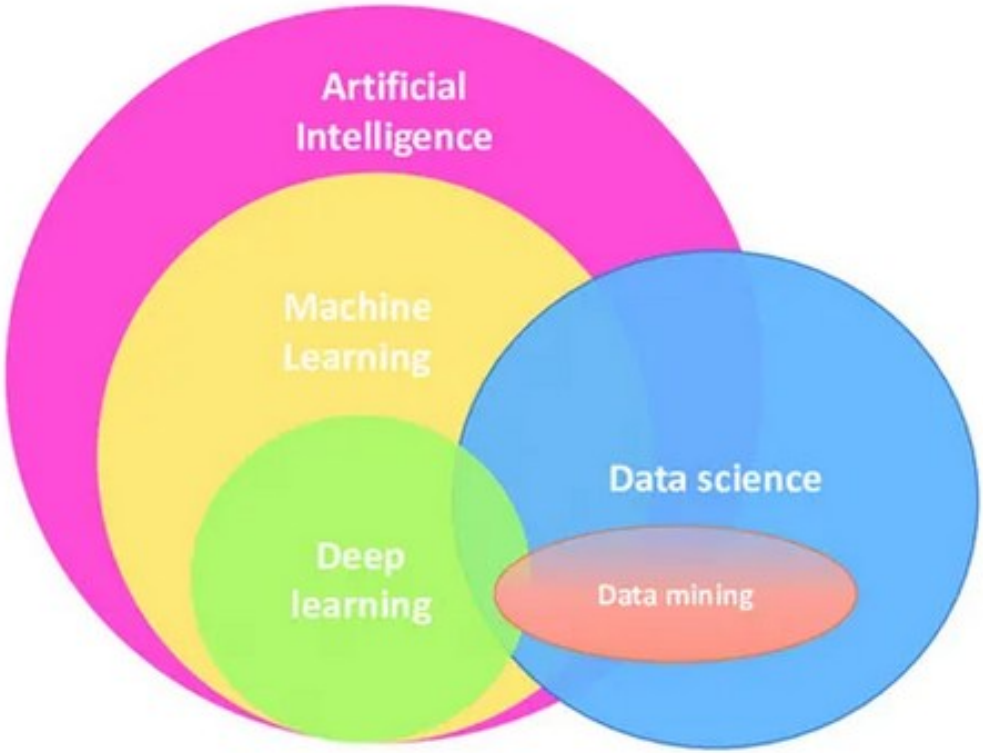
Learning Goals

- Understand key concepts in data mining and machine learning
- Identify different types of machine learning techniques
- Apply basic predictive analytics concepts
- Relate analytics to sustainability and business problems

Session Agenda

- Concepts
- Data Mining Concepts
- Machine Learning Basics
- Predictive Analytics & Applications
- Practice: Classification & Clustering

Concepts


Artificial Intelligence
Machine Learning
Deep learning
Data science
Data mining

Data Science

- Extracting Insights from Data: Data Science involves extracting actionable insights and knowledge from vast amounts of data.

What is Data Mining?

The analyst knew he was in for a rough day when his boss came in dressed like

- Process of discovering patterns in large datasets
- Combines statistics, ML, and database systems

Data Mining Tasks

- Classification
- Clustering
- Association Rules
- Anomaly Detection

Example - Retail

- Market basket analysis
- "Customers who buy X also buy Y"

Example - Sustainability

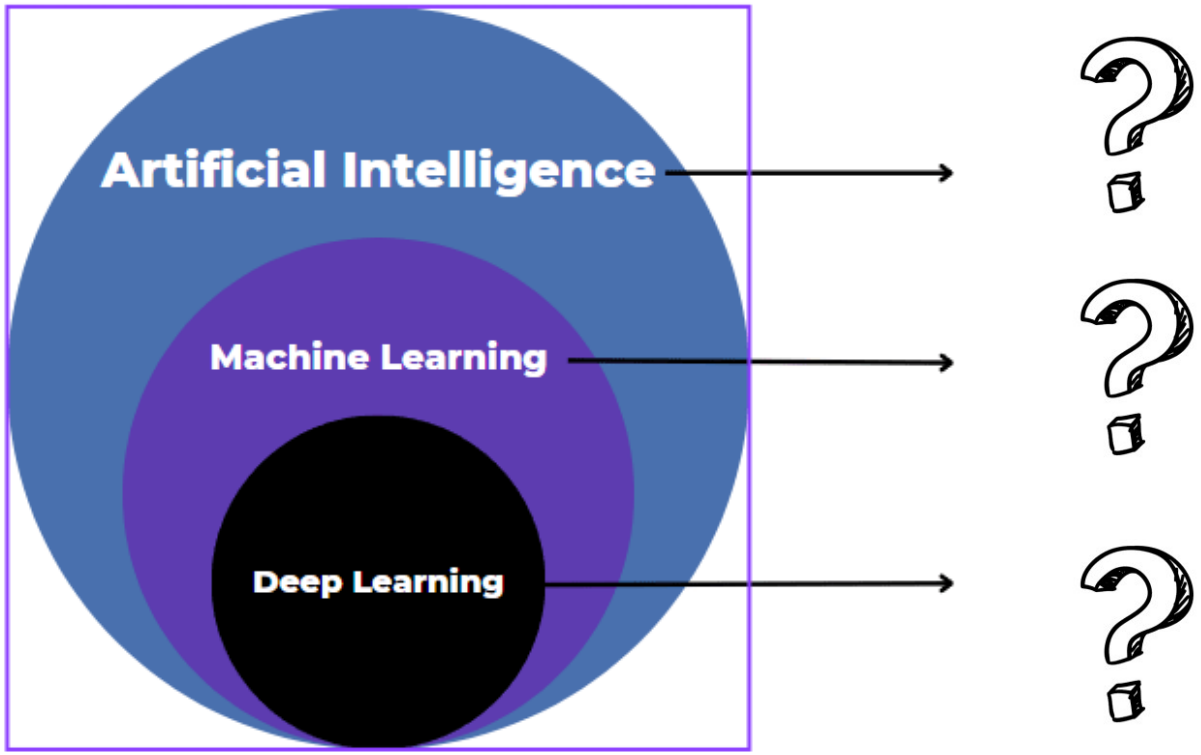
- Detecting abnormal energy consumption
- Identifying waste patterns

"I think I now know who drank our energy drinks."

Activity (5 -10 min)

- Think of a dataset
- What patterns would you try to find?

What is AI, ML, DL?



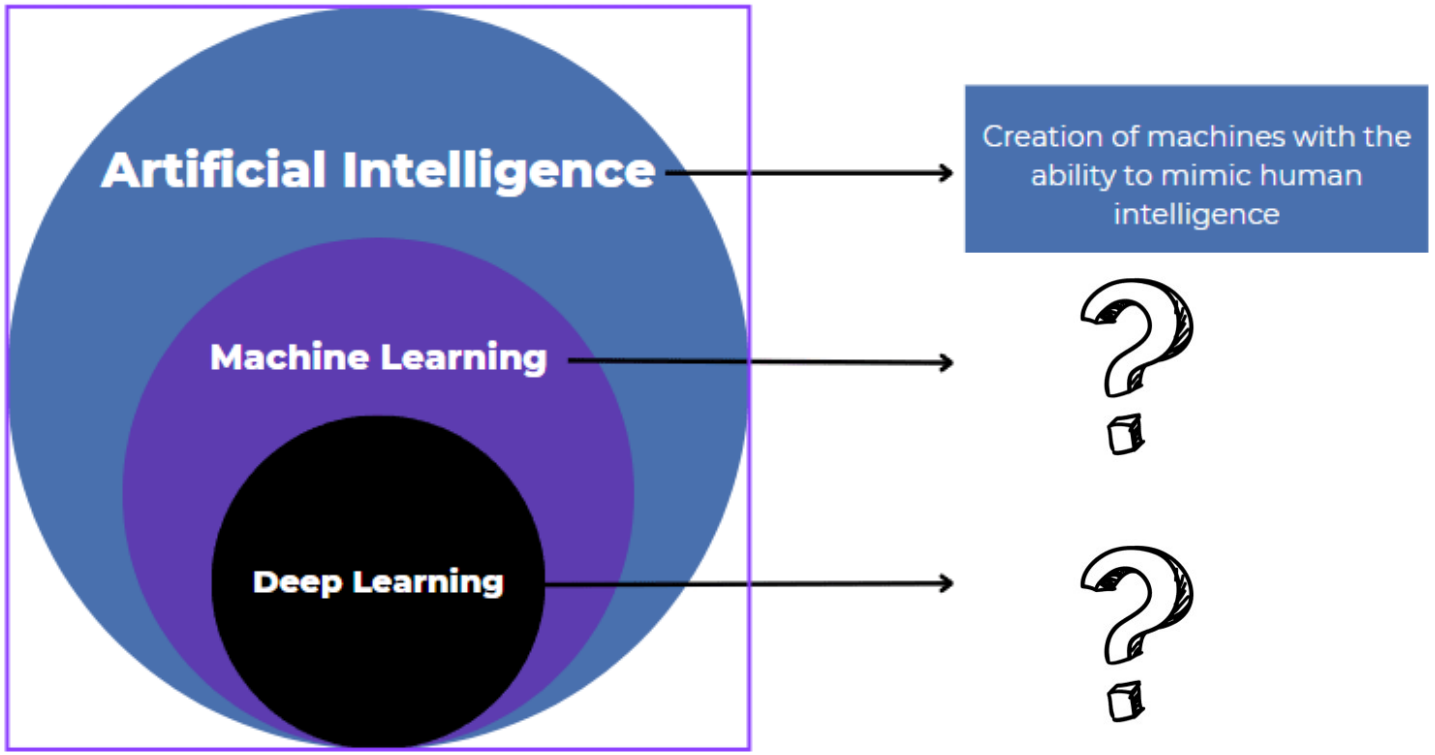
WHAT IS A.I.?

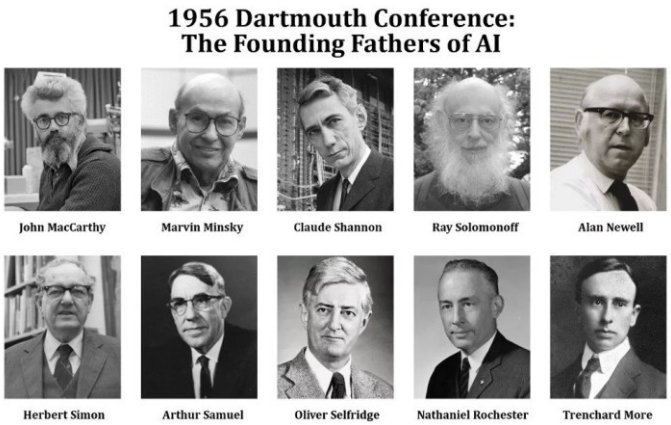
WHAT IS A.I.?

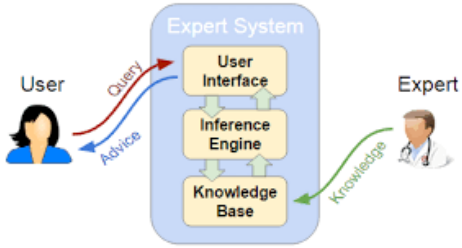
Artificial Intelligence(AI)

- Artificial intelligence refers to the development of computer-based solutions that can perform tasks which mimic human intelligence.

What is Machine Learning?







Symbolic AI

Euristic Search

Winter 1

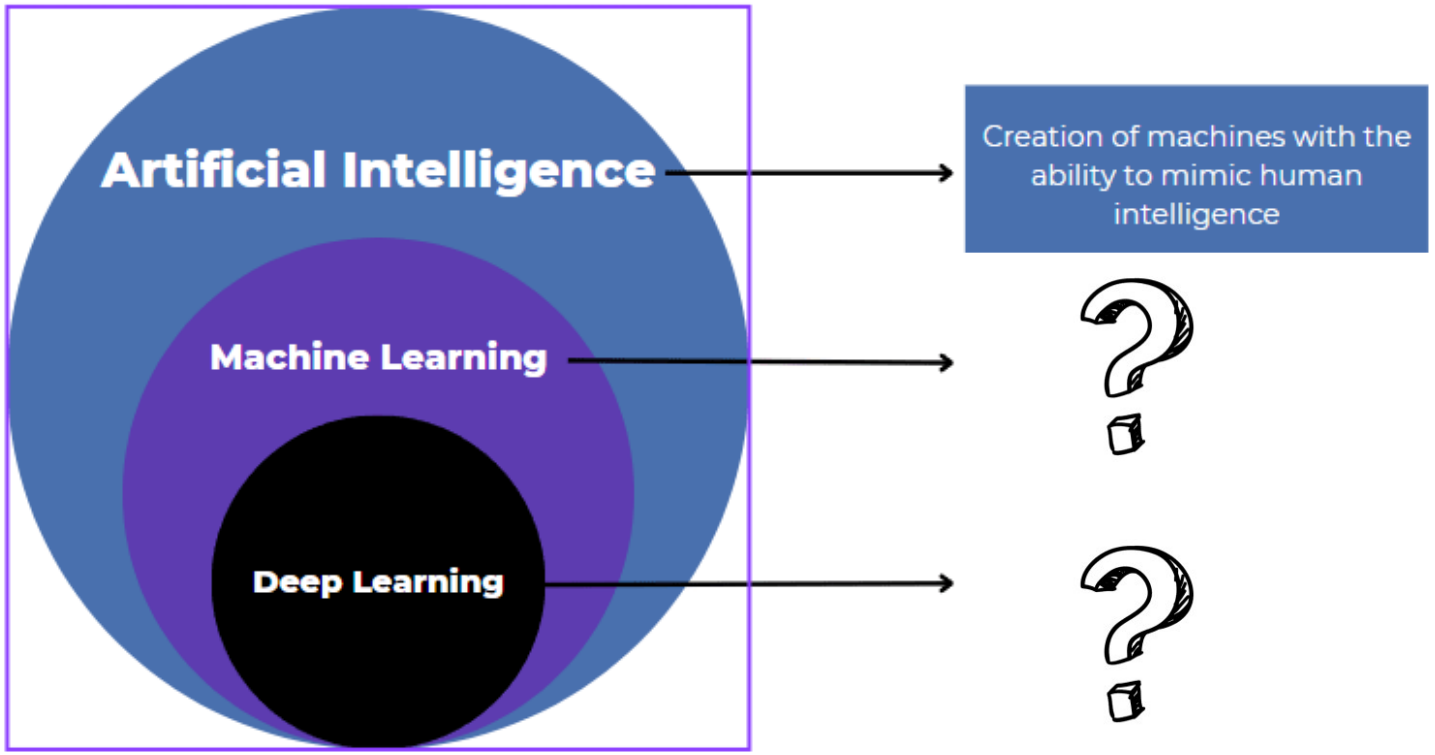
Expert Systems

Knowledge Engineering

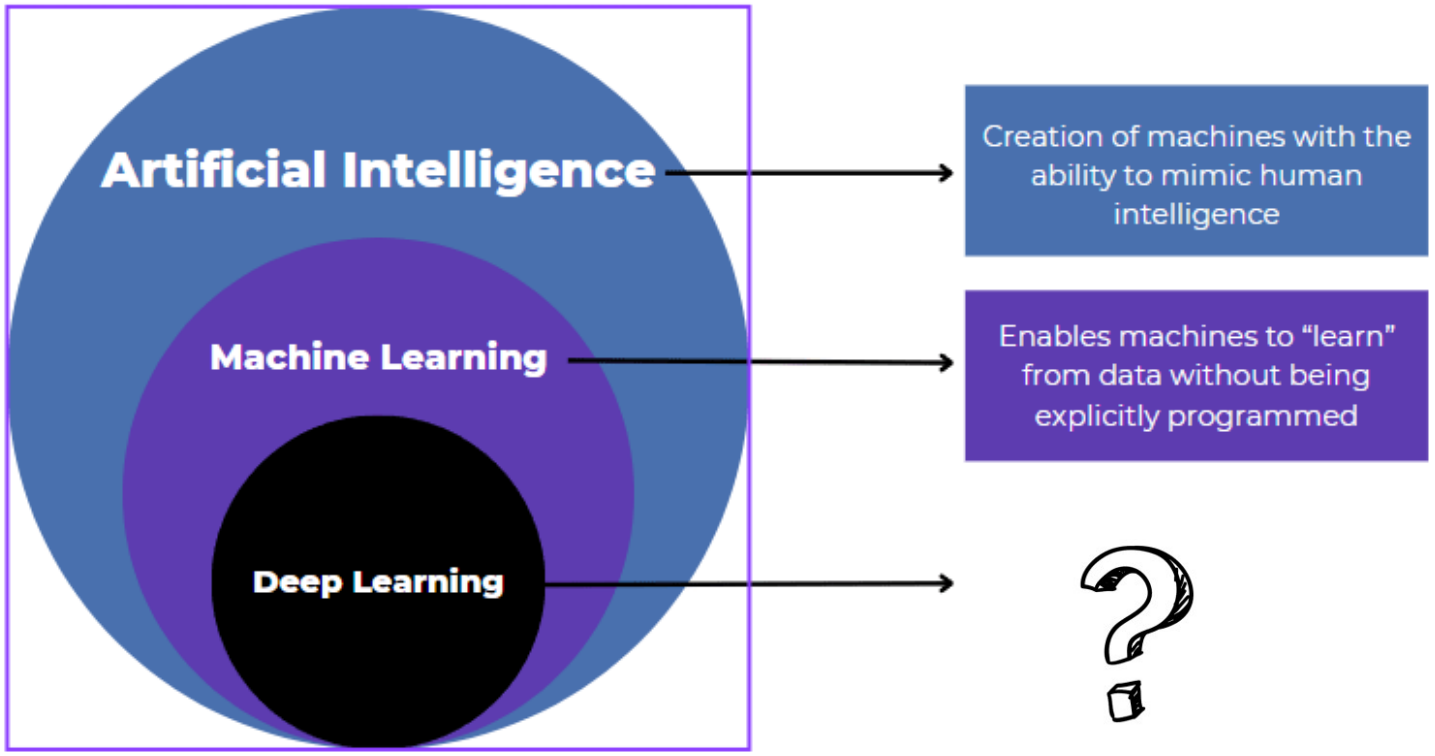
Winter 2

1950 1960 1970 1980 1990 2000 2010 2020

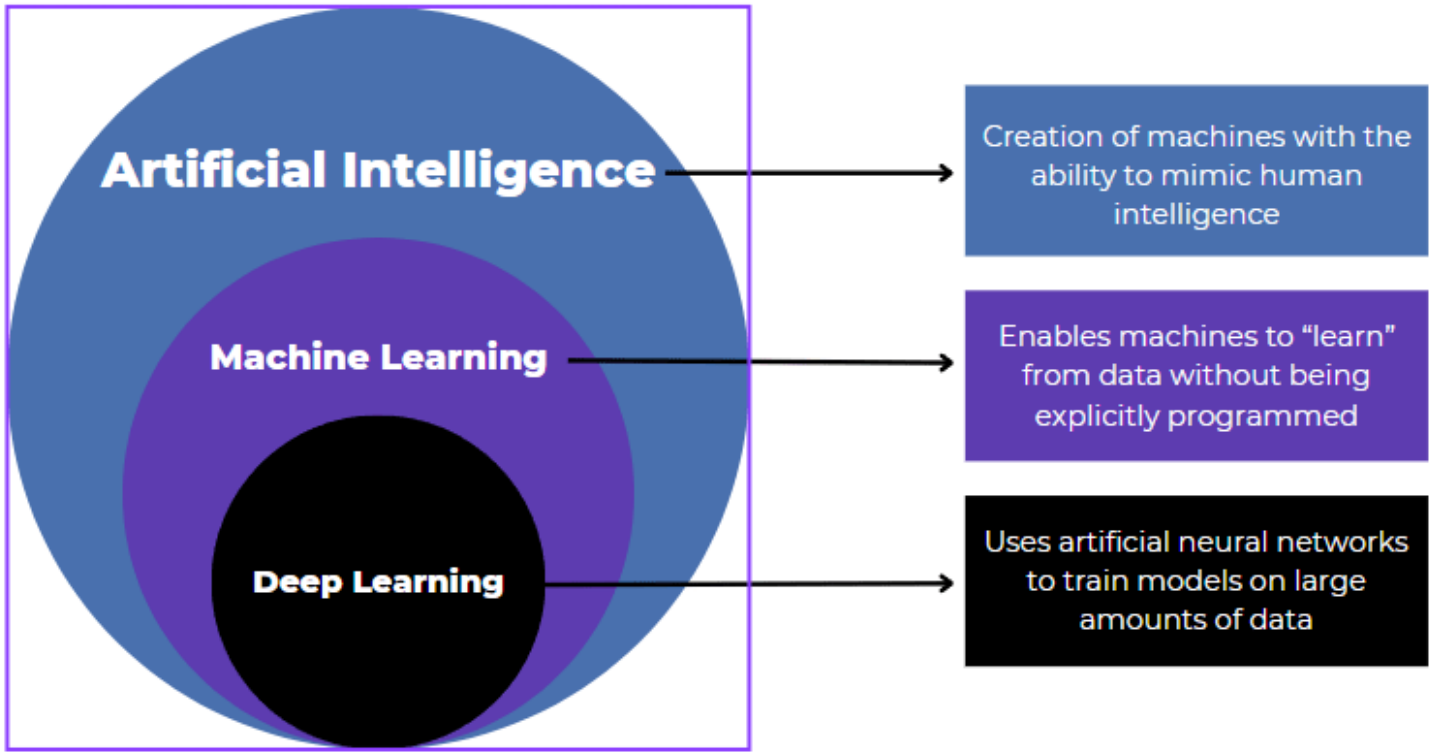
What is Machine Learning?



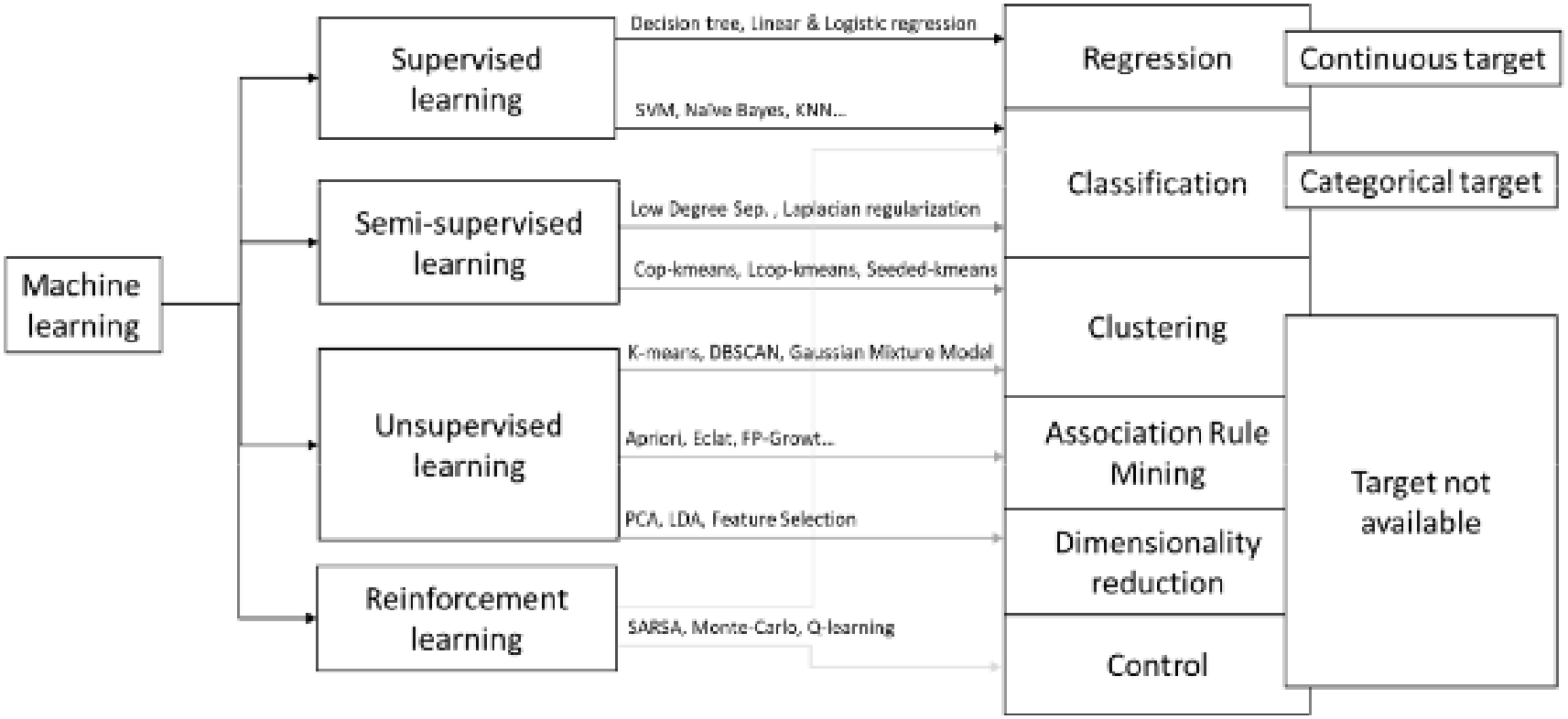
What is Deep Learning?



What is Machine Learning?



Types of Machine Learning

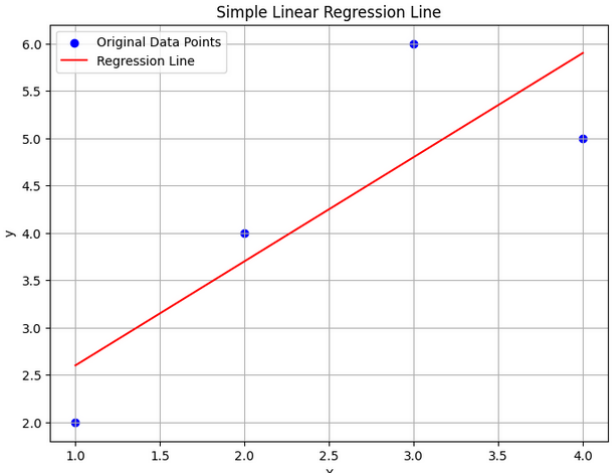


- Aparicio, Romao & Costa (2022)

Supervised Learning

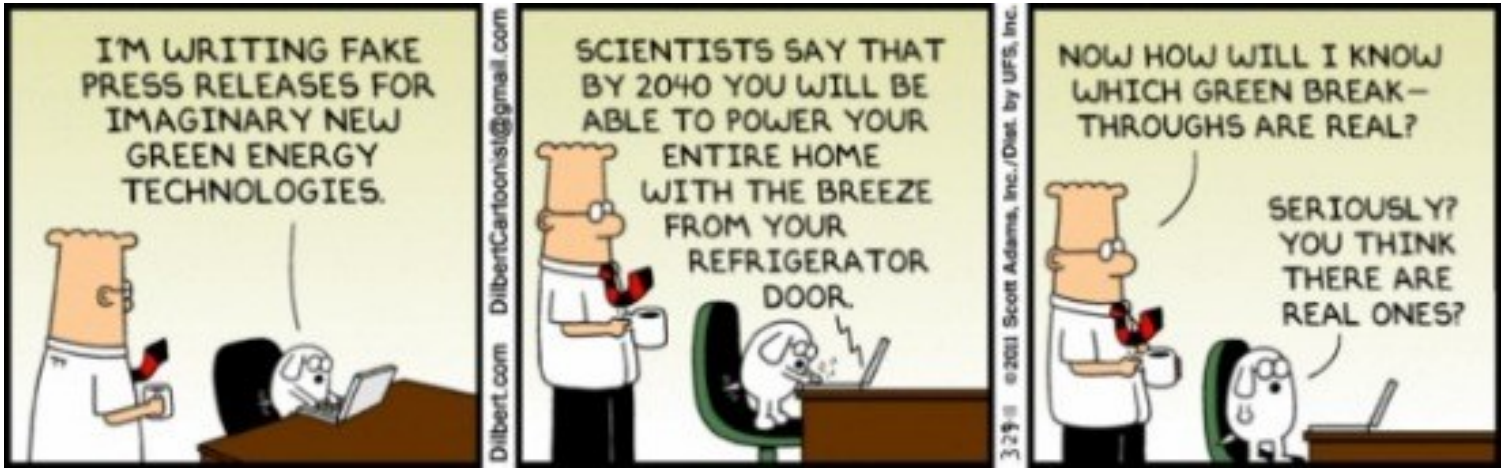
- Uses labeled data
- Tasks: classification & regression

```
from sklearn.linear_model import LinearRegression

X = [[1], [2], [3], [4]]
y = [2, 4, 6, 5]

model = LinearRegression()
model.fit(X, y)
print(model.predict([[4]]))
```



Example – Classification

- Email spam detection: Predict "spam" or "not spam“
- Fake news

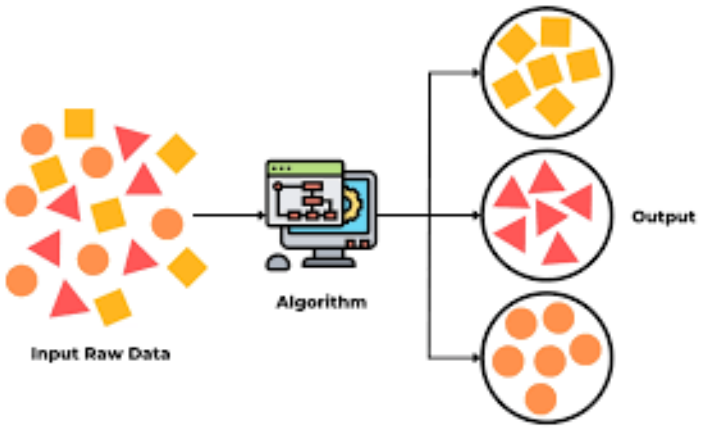


Example – Regression

- Predict house prices (e.g. Samadani & Costa, 2021)
- Predict energy consumption

Unsupervised Learning

- No labeled data
- Discover hidden patterns

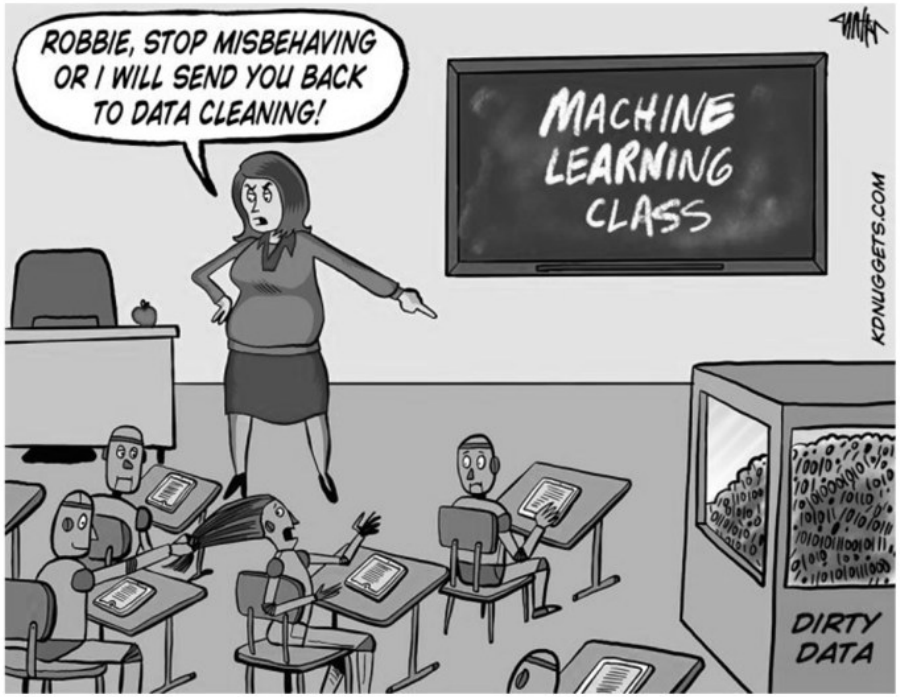


Example – Clustering

- Group customers by behavior
- Segment users

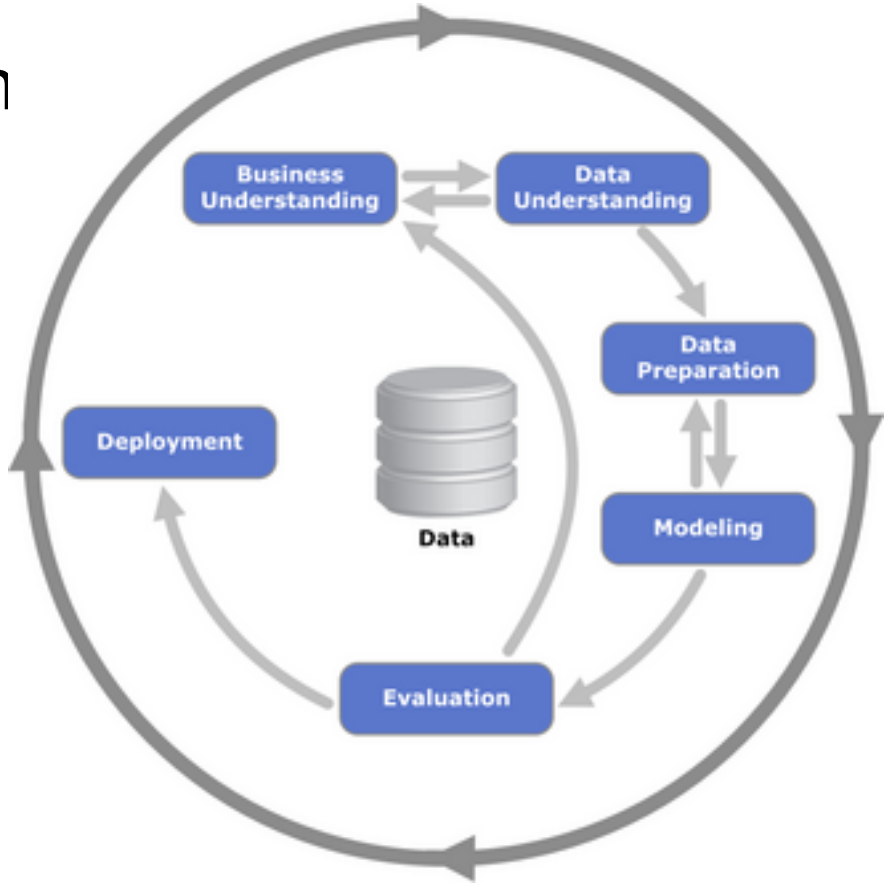
Reinforcement Learning

- Learning via rewards and penalties
- Example: recommendation systems



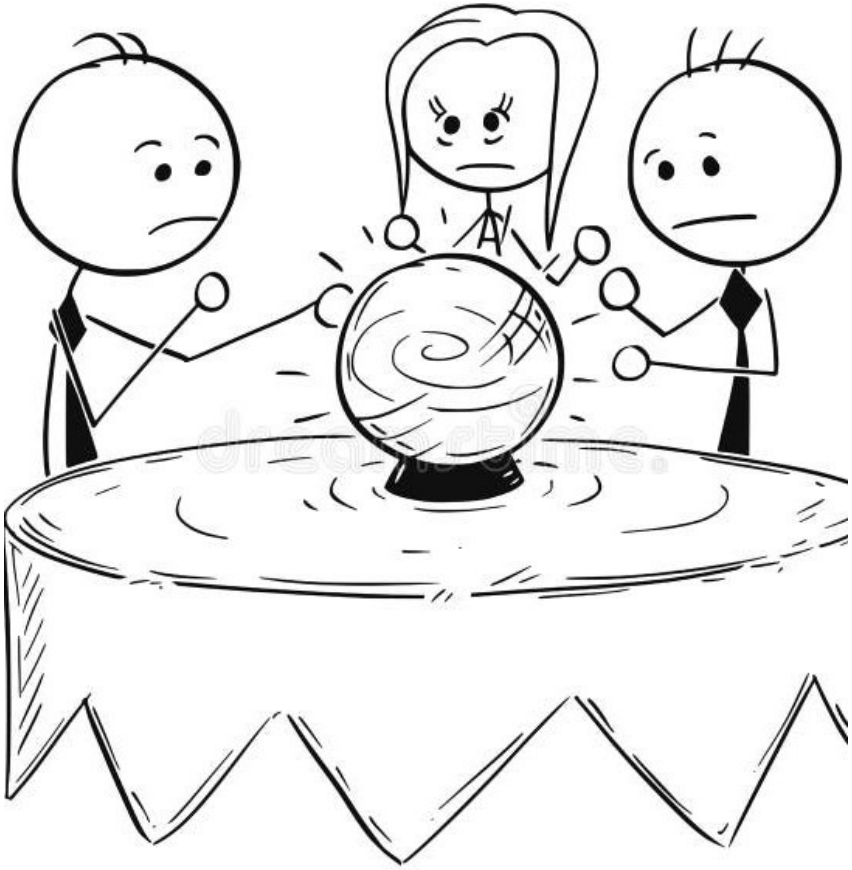
CRISP-DM

- Business Understandin
- Data Understanding
- Data Preparation
- Evaluation
- Deployment



Costa, & Aparicio (2020)

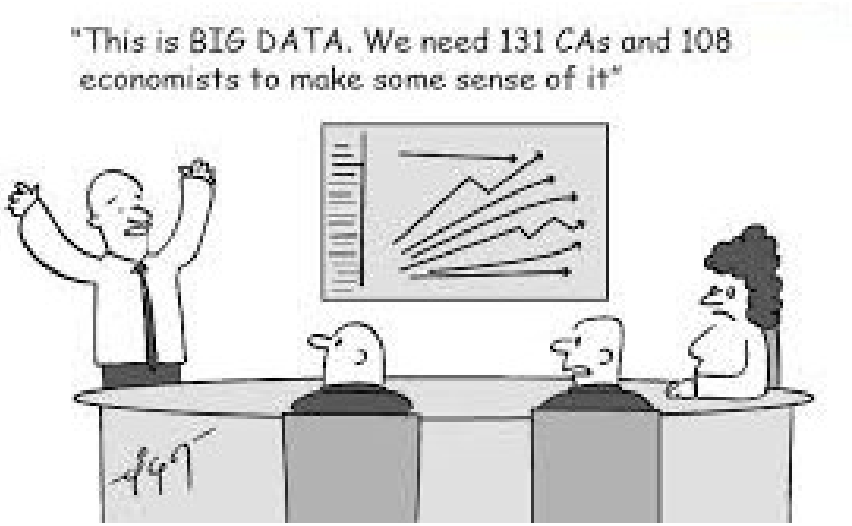
What is Predictive Analytics?



- Using data to predict future outcomes

Predictive Models

- Regression models
- Decision trees
- Time series models
- …



Industry Example

- Retail: demand forecasting
- Finance: fraud detection


CASE
STUDY

Case Study: Household Energy Consumption in Europe

Context (Eurostat data)

- Household energy use across Europe
- Trade-off: **cost vs. sustainability vs. energy security**

Problem

- High dependence on fossil fuels
- Uneven adoption of renewables
- Exposure to price volatility

Your Role

- Data analyst advising **policy/energy managers**
- Turn data into **economic & strategic insights**

Analysis Focus

- Trends: Fossil fuels vs. renewables
- Drivers of renewable consumption
- Country segmentation
- Forecasting energy patterns

Managerial Goal

- Improve **efficiency + sustainability**
- Support **investment & policy decisions**

Key Question

How can household energy consumption be optimized for a sustainable and cost-efficient future?

Data Loading

```
!pip install -q Eurostat
import eurostat
import pandas as pd
import numpy as np
data =
eurostat.get_data_df("NRG_D_HHQ")
data.head()
```

Data Preprocessing and Feature Engineering

```
NRG_BAL_LABELS = {
    "FC_OTH_HH_E":    "Total Final Energy Consumption in Households",
    "FC_OTH_HH_E_CK": "Cooking",
    "FC_OTH_HH_E_LE": "Lighting and electrical appliances",
    "FC_OTH_HH_E_OE": "Other end uses",
    "FC_OTH_HH_E_SC": "Space cooling (Air conditioning)",
    "FC_OTH_HH_E_SH": "Space heating",
    "FC_OTH_HH_E_WH": "Water heating",
}

SIEC_LABELS = {
    "TOTAL": "All energy products",
    "E7000": "Electricity",
    "G3000": "Natural gas",
    "H8000": "Heat (district heating)",
    "O4000": "Oil & petroleum products (total)",
    "O4630": "LPG",
    "O4669": "Gas oil & diesel oil",
    "O4671": "Fuel oil",
    "SFF_P1000_S2000": "Solid fossil fuels (coal)",
    "RA000": "Renewables & waste (total)",
    "RA410": "Solid biofuels",
    "RA600": "Ambient energy (heat pumps)",
    "R5300": "Renewable electricity",
    "R5110-5150_W6000RI": "Hydro/Wind/Solar/Renewable waste",
}
```

Data Preprocessing and Feature Engineering

```
SIEC_GROUPS = {
    "TOTAL": "Main Energy Carriers",
    "E7000": "Main Energy Carriers",
    "G3000": "Main Energy Carriers",
    "H8000": "Main Energy Carriers",
    "O4000": "Fossil Fuels (Oil & Coal)",
    "O4630": "Fossil Fuels (Oil & Coal)",
    "O4669": "Fossil Fuels (Oil & Coal)",
    "O4671": "Fossil Fuels (Oil & Coal)",
    "SFF_P1000_S2000": "Fossil Fuels (Oil & Coal)",
    "RA000": "Renewables & Waste",
    "RA410": "Renewables & Waste",
    "RA600": "Renewables & Waste",
    "R5300": "Renewables & Waste",
    "R5110-5150_W6000RI": "Renewables & Waste",
}
```

Data Preprocessing and Feature Engineering

```
df = data.copy()

df["nrg_bal_label"] = df["nrg_bal"].map(NRG_BAL_LABELS)
df["siec_label"] = df["siec"].map(SIEC_LABELS)
df["siec_group"] = df["siec"].map(SIEC_GROUPS)

# Rename geo column for readability in slides
df = df.rename(columns={"geo\\TIME_PERIOD": "geo"})

df[["geo", "nrg_bal_label", "siec_group"]].head()
```

Data Preprocessing and Feature Engineering

```
year_cols = [c for c in df.columns if c.isdigit()]

df_long = (
    df.drop(columns=["nrg_bal", "siec"])
      .melt(
          id_vars=["freq", "unit", "geo", "nrg_bal_label", "siec_label", "siec_group"],
          value_vars=year_cols,
          var_name="year",
          value_name="value",
      )
)

df_long["year"] = df_long["year"].astype(int)
df_long.head()
```

Data Preprocessing and Feature Engineering

```
pivoted = (
    df_long.pivot_table(
        index=["geo", "nrg_bal_label", "year"],
        columns="siec_group",
        values="value",
        aggfunc="mean",
    )
    .reset_index()
)

pivoted.head()
```

Regression Analysis - Statsmodels

```
Regression Results for 'Renewables & Waste' using Statsmodels:
                            OLS Regression Results
==============================================================================
Dep. Variable:     Renewables & Waste   R-squared:                       0.940
Model:                            OLS   Adj. R-squared:                  0.940
Method:                 Least Squares   F-statistic:                 1.478e+04
Date:                Mon, 06 Apr 2026   Prob (F-statistic):               0.00
Time:                        18:00:03   Log-Likelihood:                -32445.
No. Observations:                2813   AIC:                          6.490e+04
Df Residuals:                    2809   BIC:                          6.492e+04
Df Model:                           3
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
const                        -1.97e+06   2.36e+05     -8.342      0.000   -2.43e+06   -1.51e+06
year                          976.5906    117.097      8.340      0.000     746.985    1206.196
Fossil Fuels (Oil & Coal)       0.8455      0.051     16.626      0.000       0.746       0.945
Main Energy Carriers            0.1638      0.005     34.430      0.000       0.154       0.173
==============================================================================
Omnibus:                     1984.195   Durbin-Watson:                   0.471
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           214018.439
Skew:                           2.518   Prob(JB):                         0.00
Kurtosis:                      45.434   Cond. No.                     2.14e+08
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.14e+08. This might indicate that there are
strong multicollinearity or other numerical problems.
```

```
import statsmodels.api as sm

y_col = "Renewables & Waste"
X_cols = ["year", "Fossil Fuels (Oil & Coal)", "Main Energy Carriers"]

df_reg = pivoted.dropna(subset=[y_col] + X_cols).copy()

X = sm.add_constant(df_reg[X_cols])
y = df_reg[y_col]

res = sm.OLS(y, X).fit()
print(res.summary())
```

Regression Analysis – Scikit-learn

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_squared_error

X = df_reg[X_cols]
y = df_reg[y_col]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

lr = LinearRegression().fit(X_train, y_train)
pred = lr.predict(X_test)

print("R²:", round(r2_score(y_test, pred), 4))
print("RMSE:", round(np.sqrt(mean_squared_error(y_test, pred)), 2))
```

```
R²: 0.9406
RMSE: 27711.89
```

Creating Binary Target for Classification

```
df_clf = pivoted.sort_values(["geo", "nrg_bal_label", "year"]).copy()

df_clf["yoy_change"] = df_clf.groupby(["geo", "nrg_bal_label"])[y_col].diff()

# Keep only rows where change exists and is non-zero (optional but cleaner)
df_clf = df_clf.dropna(subset=["yoy_change"])
df_clf = df_clf[df_clf["yoy_change"] != 0].copy()

df_clf["target"] = np.where(df_clf["yoy_change"] > 0, "Increase", "Decrease")
df_clf[["geo", "nrg_bal_label", "year", y_col, "yoy_change", "target"]].head()
```

Logistic Regression Classifier

```
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

X_cols_clf = ["year", "Fossil Fuels (Oil & Coal)", "Main Energy Carriers", "Renewables & Waste"]

X = df_clf[X_cols_clf].fillna(0)
y = df_clf["target"]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

pipe_lr = Pipeline([
    ("scaler", StandardScaler()),
    ("clf", LogisticRegression(max_iter=2000, random_state=42)),
])

pipe_lr.fit(X_train, y_train)
pred = pipe_lr.predict(X_test)

print("Accuracy:", round(accuracy_score(y_test, pred), 4))
print(classification_report(y_test, pred))
print(confusion_matrix(y_test, pred))
```

```
Accuracy: 0.6077
              precision    recall  f1-score   support

    Decrease       1.00      0.02      0.04       136
    Increase       0.60      1.00      0.75       203

    accuracy                           0.61       339
   macro avg       0.80      0.51      0.40       339
weighted avg       0.76      0.61      0.47       339

[[  3 133]
 [  0 203]]
```

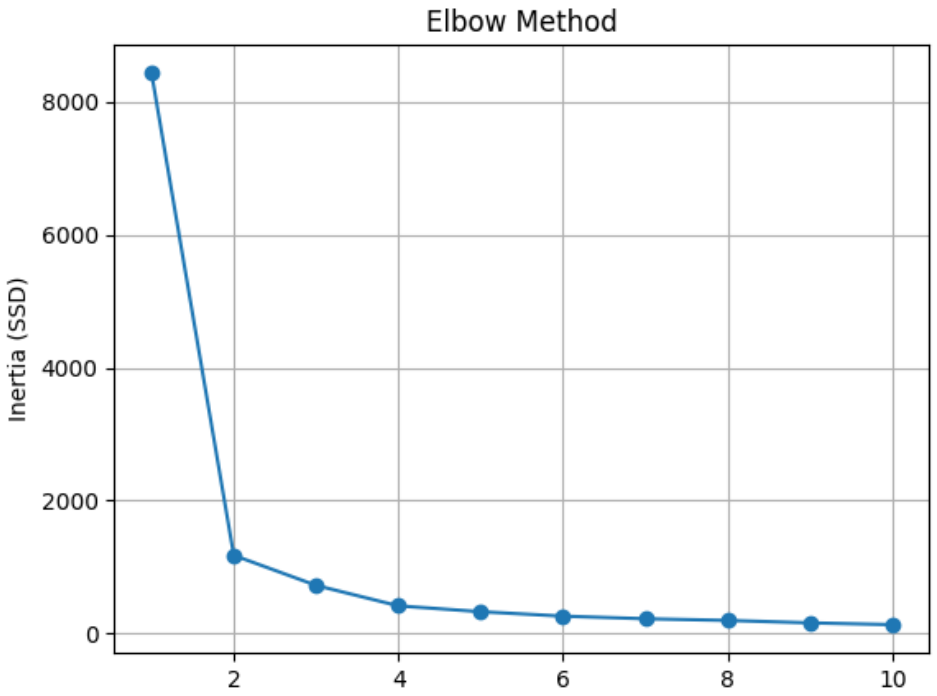
Decision Tree and Random Florest Classifiers

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier

dt = DecisionTreeClassifier(random_state=42).fit(X_train, y_train)
rf = RandomForestClassifier(random_state=42).fit(X_train, y_train)

pred_dt = dt.predict(X_test)
pred_rf = rf.predict(X_test)

print("DT accuracy:", round(accuracy_score(y_test, pred_dt), 4))
print("RF accuracy:", round(accuracy_score(y_test, pred_rf), 4))
```

```
DT accuracy: 0.649
RF accuracy: 0.6519
```

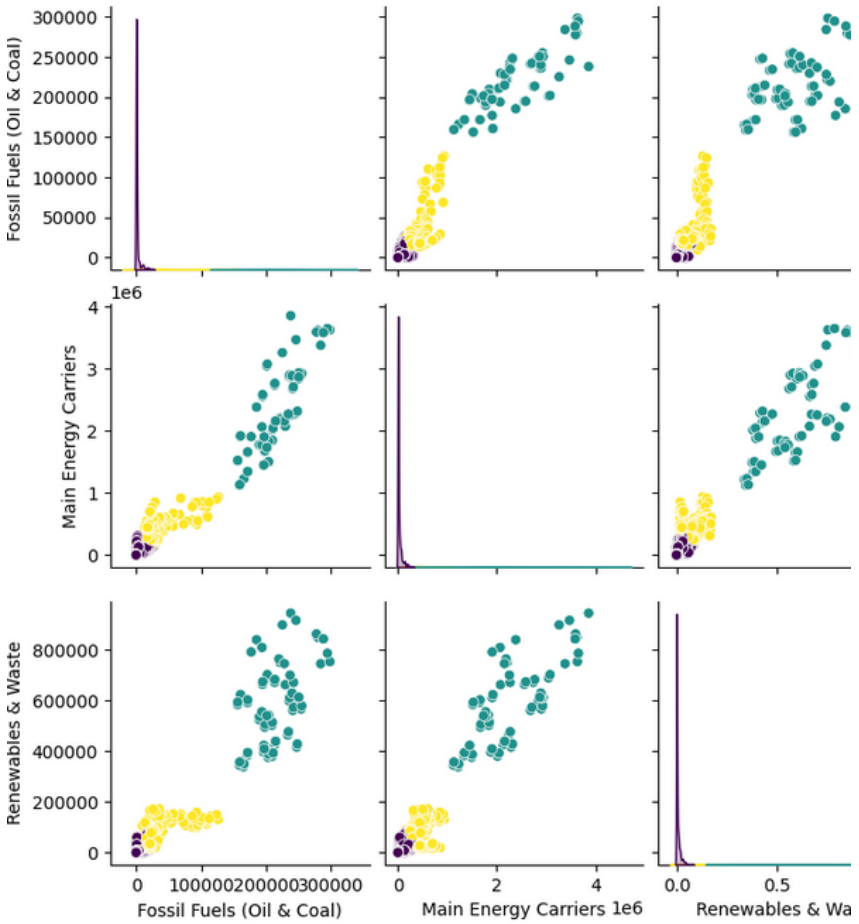
Cluster Analysis - Elbow Method

```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

features = ["Fossil Fuels (Oil & Coal)", "Main Energy Carriers", "Renewables & Waste"]

cluster_df = pivoted.dropna(subset=features).copy()
X = cluster_df[features].values

X_scaled = StandardScaler().fit_transform(X)

ssd = []
K = range(1, 11)
for k in K:
    km = KMeans(n_clusters=k, random_state=42, n_init=10).fit(X_scaled)
    ssd.append(km.inertia_)

plt.plot(list(K), ssd, marker="o")
plt.xlabel("K")
plt.ylabel("Inertia (SSD)")
plt.title("Elbow Method")
plt.grid(True)
plt.show()
```



K-Means Clustering and Visualization

```
import seaborn as sns


k = 3
km = KMeans(n_clusters=k, random_state=42,
n_init=10)
cluster_df["cluster"] = km.fit_predict(X_scaled)


sns.pairplot(cluster_df, vars=features, hue="cluster",
palette="viridis")
plt.show()
```



- Plot the trend
- Try simple prediction using regression

Summary

- Data mining finds patterns
- Machine learning builds models
- Predictive analytics forecasts outcomes
- Applications support sustainability

- Next class:
- Apache Spark ecosystem
- Hadoop ecosystem
- Databricks
- Create account in

References

Aparicio, J. T., de Sequeira, J. S., & Costa, C. J. (2021). Emotion analysis of portuguese political parties communication over the covid-19 pandemic. In 2021 16th Iberian Conference on Information Systems and Technologies (CISTI) (pp. 1-6). IEEE.

Aparicio, J. T., Romao, M., & Costa, C. J. (2022). Predicting Bitcoin prices: The effect of interest rate, search on the internet, and energy prices. In 2022 17th Iberian Conference on Information Systems and Technologies (CISTI) (pp. 1-5). IEEE.

Aparicio, S., Aparicio, J. T., & Costa, C. J. (2019,). Data Science and AI: trends analysis. In 2019 14th Iberian Conference on Information Systems and Technologies (CISTI) (pp. 1-6). IEEE.

Arriaga, A., & Costa, C. J. (2023, May). Modeling and Predicting Daily COVID-19 (SARS-CoV-2) Mortality in Portugal: The Impact of the Daily Cases, Vaccination, and Daily Temperatures. In Proceedings of International Conference on Information Technology and Applications: ICITA 2022 (pp. 275-285). Singapore: Springer Nature Singapore.

Costa, C. J., & Aparicio, J. T. (2020). POST-DS: A methodology to boost data science. In 2020 15th Iberian Conference on Information Systems and Technologies (CISTI) (pp. 1-6). IEEE.

Costa, C. J., & Aparicio, M. (2023). Applications of Data Science and Artificial Intelligence. Appl. Sci, 13, 9015.

Costa, C., Aparicio, M., & Aparicio, J. (2021). Sentiment analysis of portuguese political parties communication. In Proceedings of the 39th ACM International Conference on Design of Communication (pp. 63-69).

Custódio, J. P. G., Costa, C. J., & Carvalho, J. P. (2020). Success prediction of leads–A machine learning approach. In 2020 15th Iberian Conference on Information Systems and Technologies (CISTI) (pp. 1-6). IEEE.

Hajishirzi, R., & Costa, C. J. (2021). Artificial Intelligence as the core technology for the Digital Transformation process. In 2021 16th Iberian Conference on Information Systems and Technologies (CISTI) (pp. 1-6). IEEE.

Samadani, S., & Costa, C. J. (2021). Forecasting real estate prices in Portugal: A data science approach. In 2021 16th Iberian Conference on Information Systems and Technologies (CISTI) (pp. 1-6). IEEE.