Introduction to Data Science

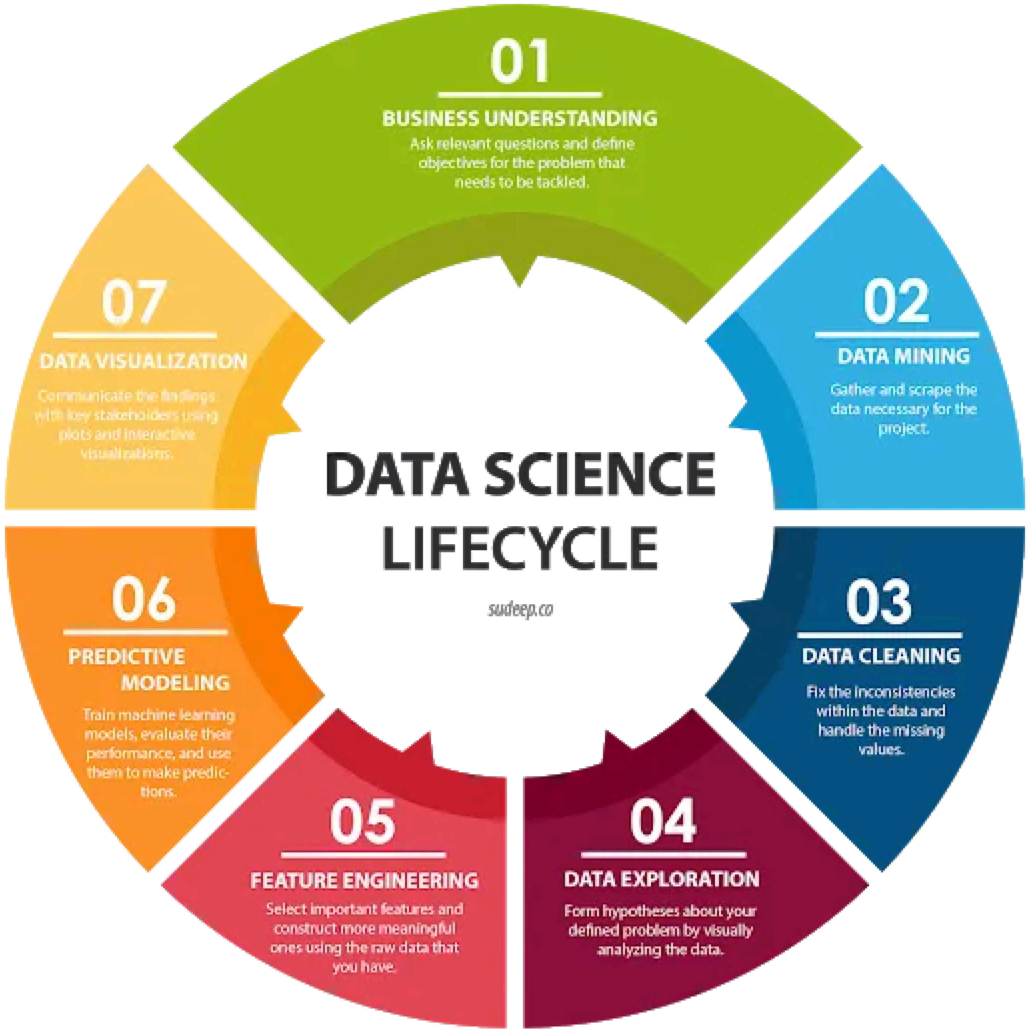
An easy guide to simple data science and machine learning

Daniel Lee

Contents

1. What is Data Science?
2. What is Machine Learning?
3. Principal Component Analysis (PCA)
4. Linear Regression
5. Logistic Regression

What is Data Science?



01 BUSINESS UNDERSTANDING
Ask relevant questions and define objectives for the problem that needs to be tackled.

02 DATA MINING
Gather and scrape the data necessary for the project.

03 DATA CLEANING
Fix the inconsistencies within the data and handle the missing values.

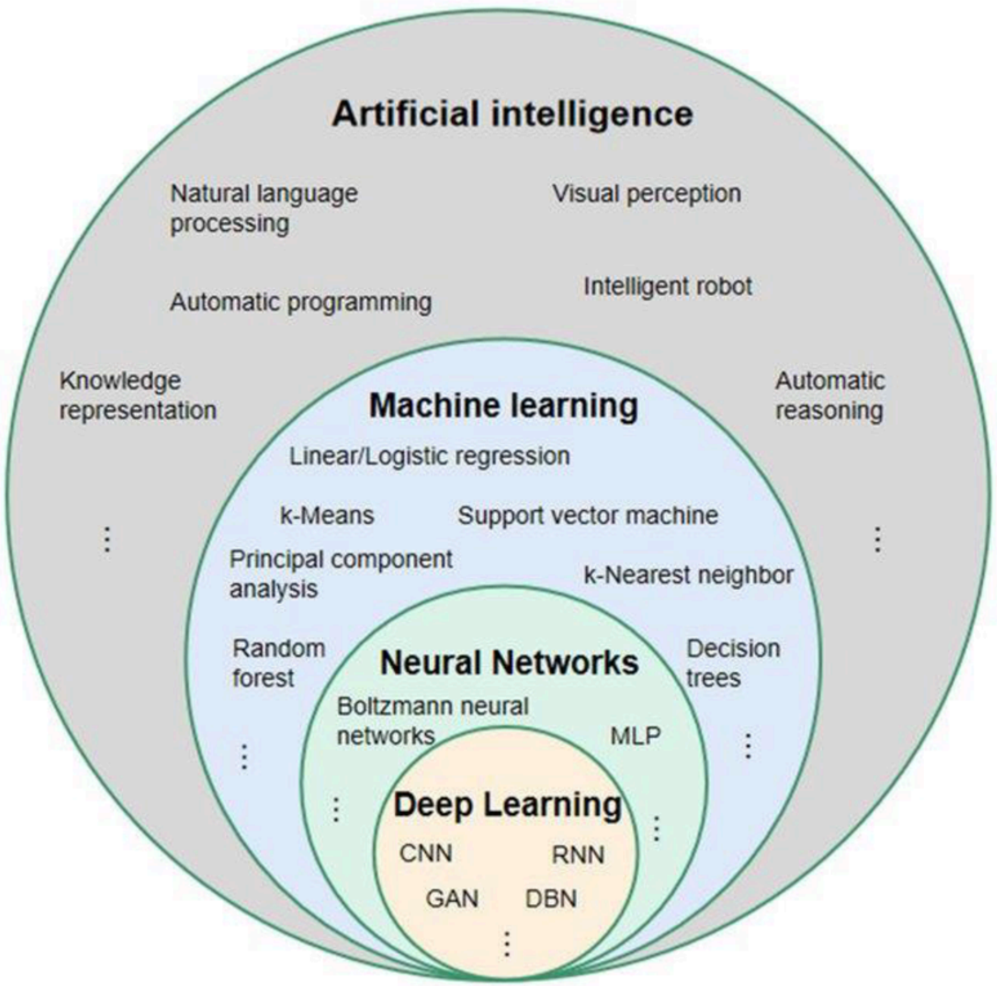
04 DATA EXPLORATION
Form hypotheses about your defined problem by visually analyzing the data.

05 FEATURE ENGINEERING
Select important features and construct more meaningful ones using the raw data that you have.

06 PREDICTIVE MODELING
Train machine learning models, evaluate their performance, and use them to make predictions.

07 DATA VISUALIZATION
Communicate the findings with key stakeholders using plots and interactive visualizations.

What is Machine Learning?



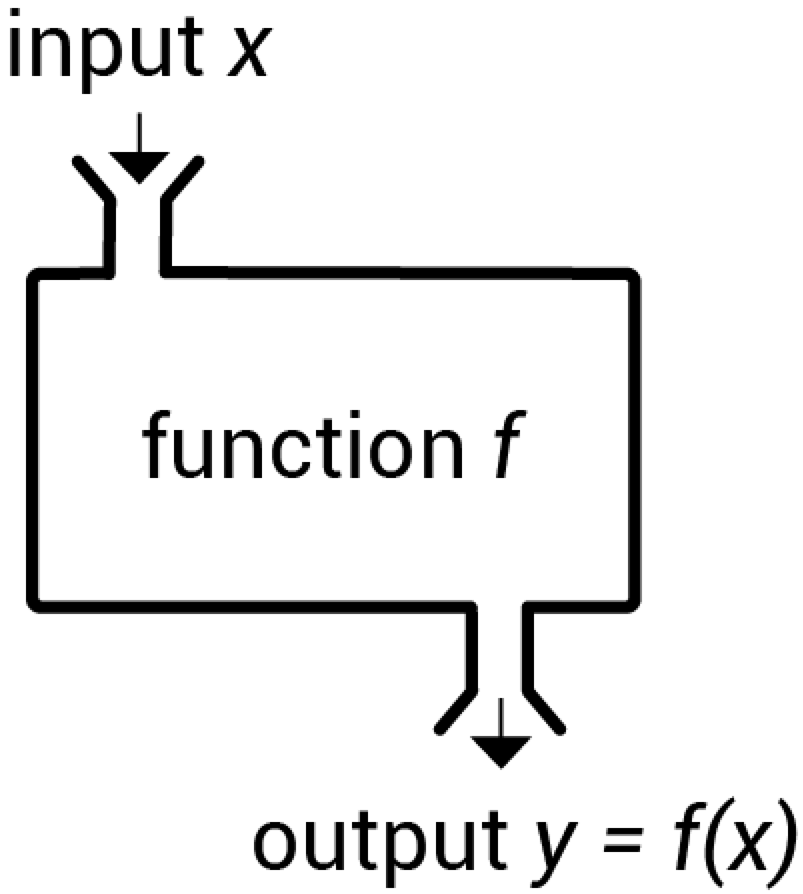
What is a Feature (X)?

A feature is an input variable used by a machine learning model.
Features are also called "independent variables" or "predictors".

Example: In predicting Titanic survival,

- Features: class, sex, age, fare, etc.
- Target (Y): survived or not

Machine Learning as a Function



input x

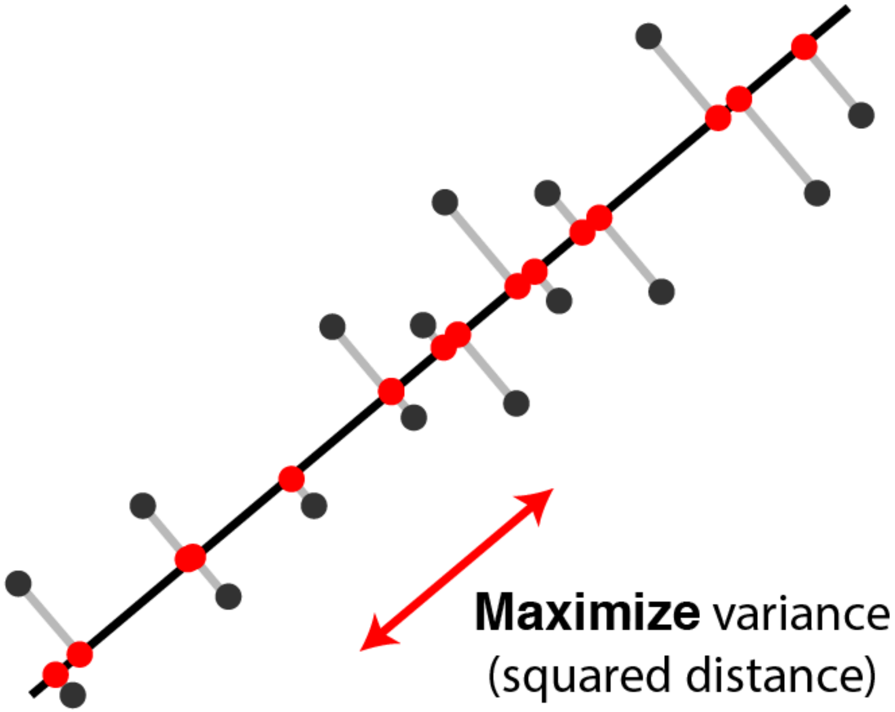
function f

output $y = f(x)$

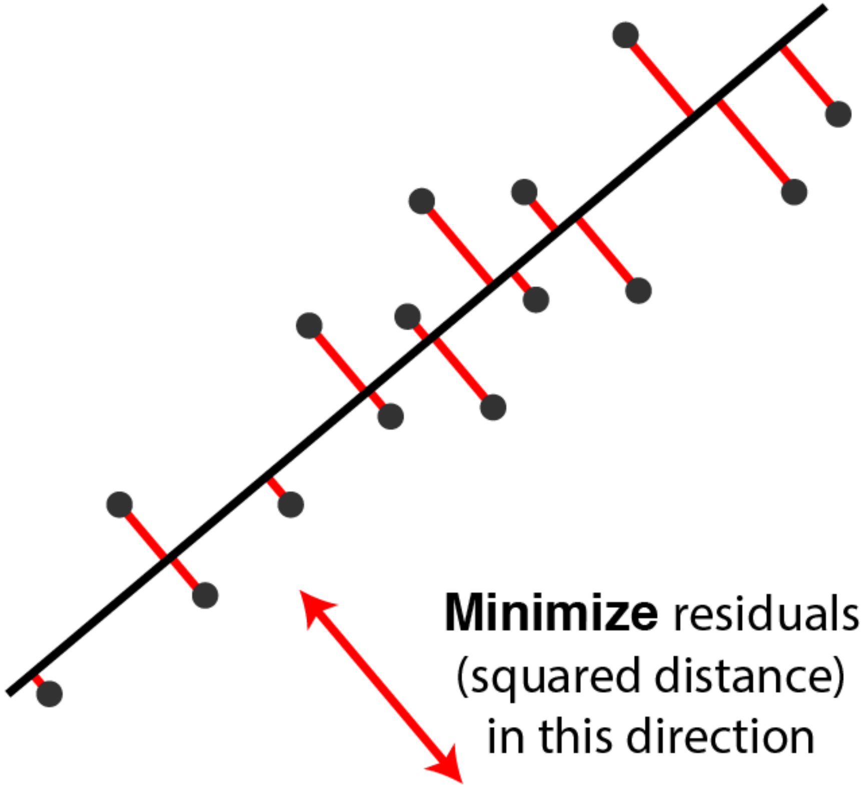
PCA (Principal Component Analysis)

PCA is a technique for reducing the dimensionality of data while preserving as much information as possible.

It finds new axes (principal components) that capture the largest **variance** in the data.

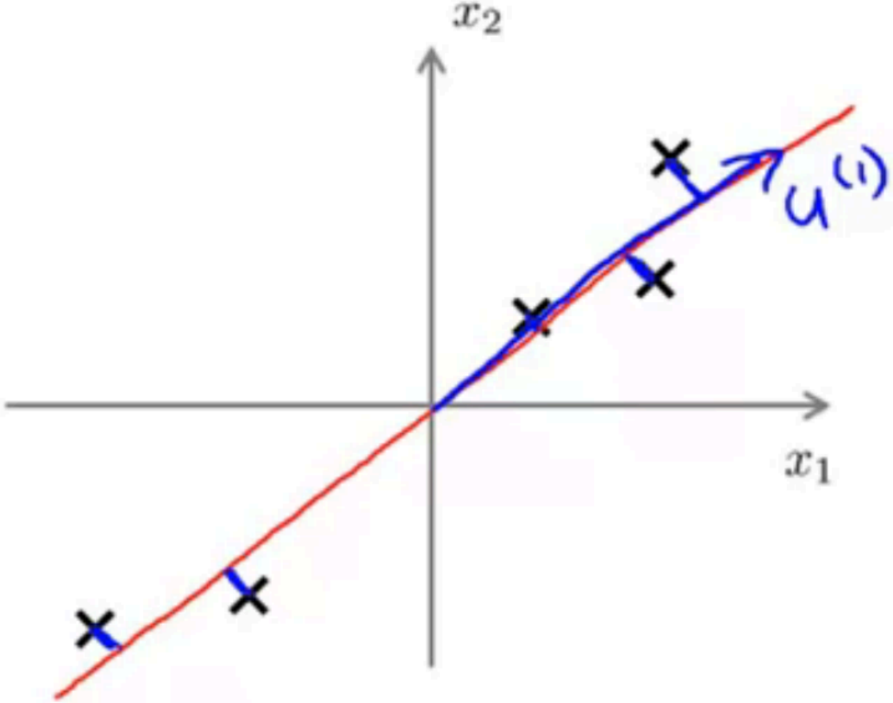


Maximize variance (squared distance) of red dots in this direction

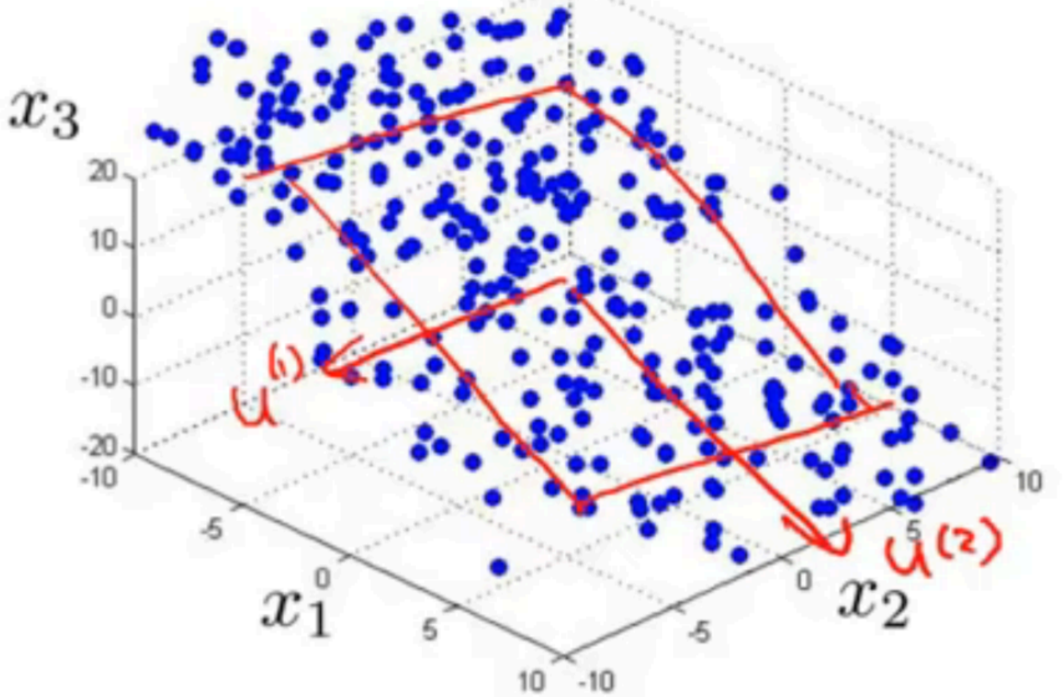


Minimize residuals (squared distance) in this direction

Principal Component Analysis (PCA) algorithm



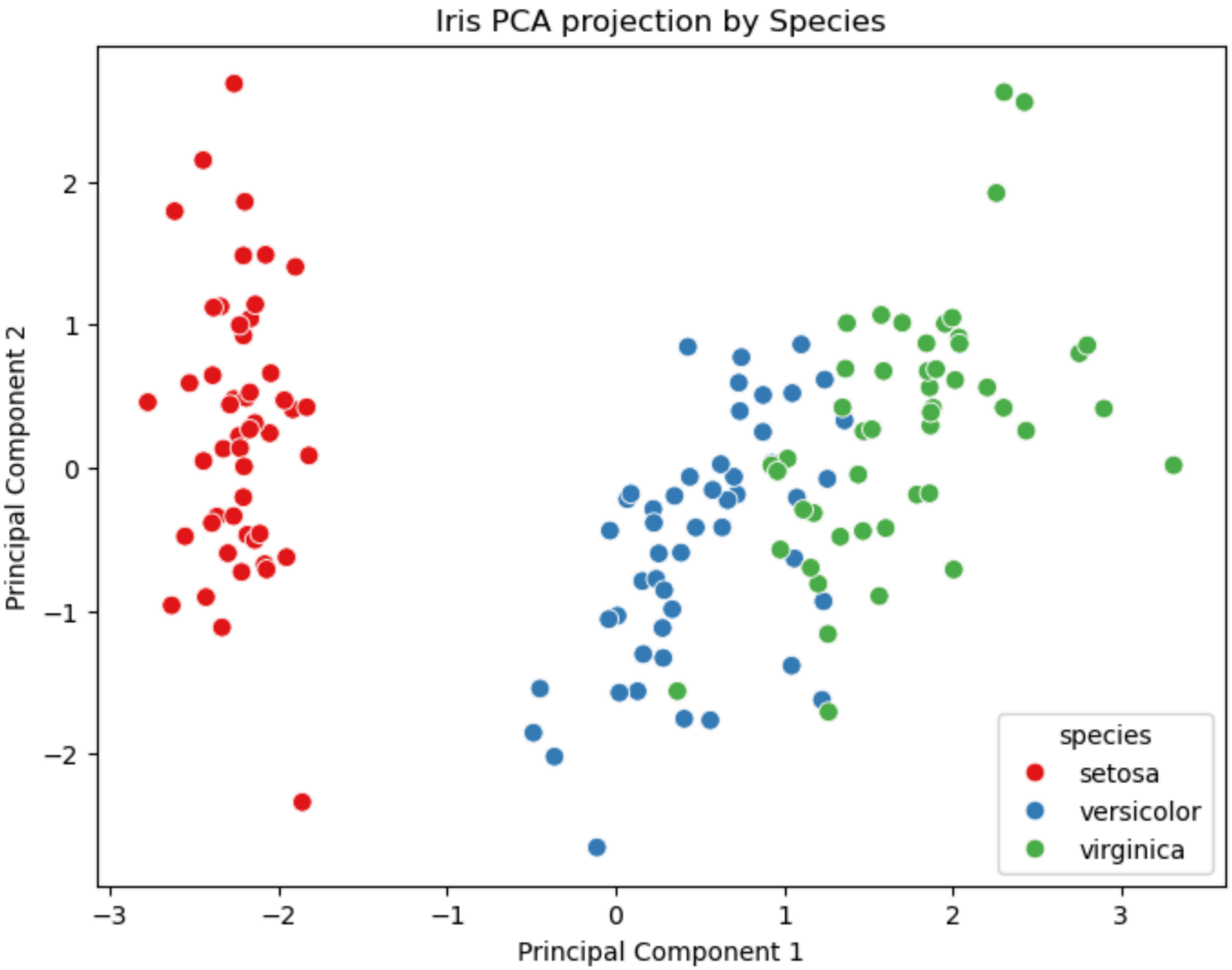
Reduce data from 2D to 1D



Reduce data from 3D to 2D

PCA Example (Iris)

```python
import seaborn as sns
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

iris = sns.load_dataset("iris")
X = iris.drop("species", axis=1)
y = iris["species"]

X_scaled = StandardScaler().fit_transform(X)

pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

iris['PC1'] = X_pca[:,0]
iris['PC2'] = X_pca[:,1]

plt.figure(figsize=(8,6))
sns.scatterplot(
    x='PC1', y='PC2',
    hue='species',
    data=iris,
    palette='Set1',
    s=60
)
plt.title('Iris PCA projection by Species')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.show()
```

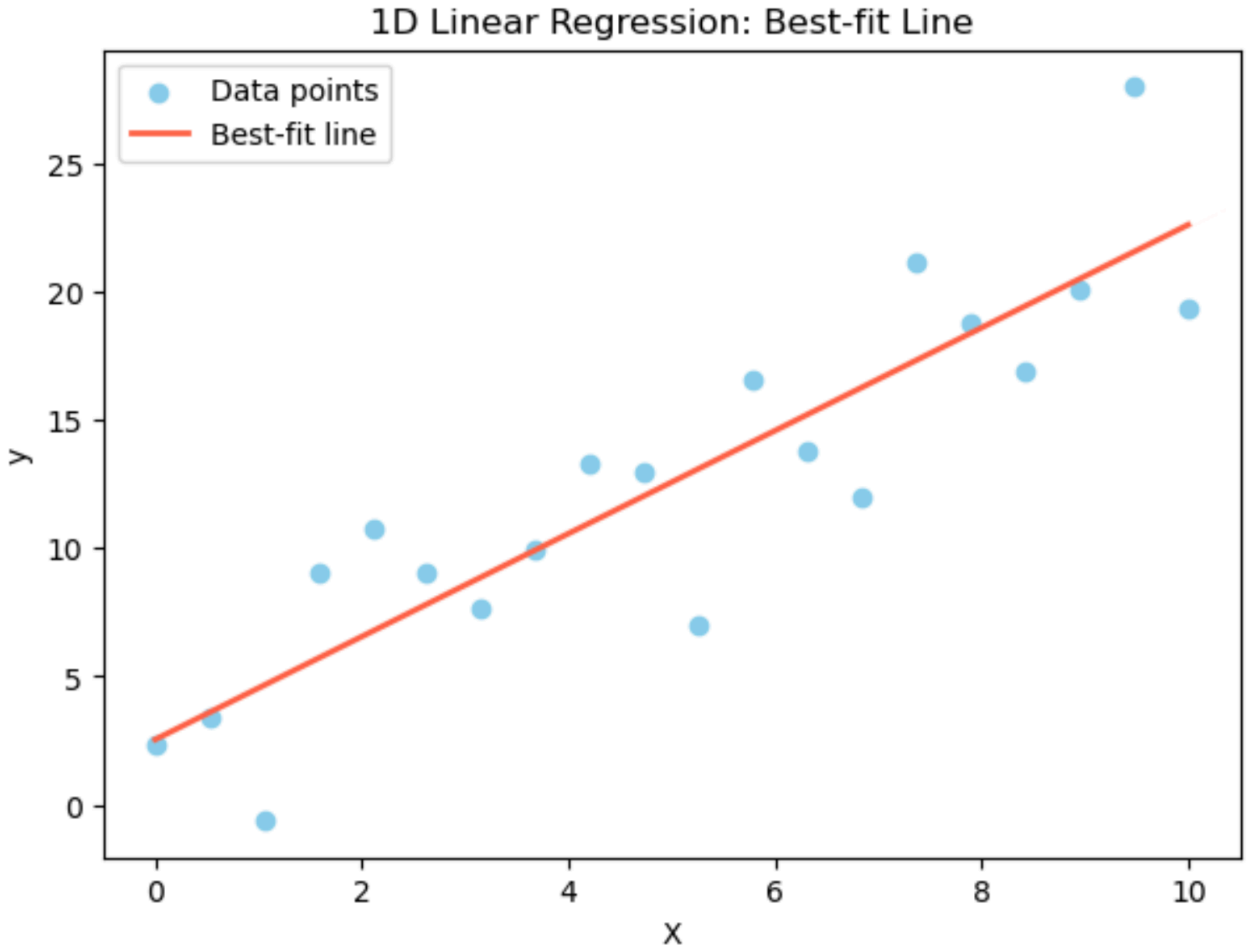

Iris PCA projection by Species
Principal Component 2
2
1
0
−1
−2
Principal Component 1
−3
−2
−1
0
1
2
3
species
● setosa
● versicolor
● virginica

Linear Regression

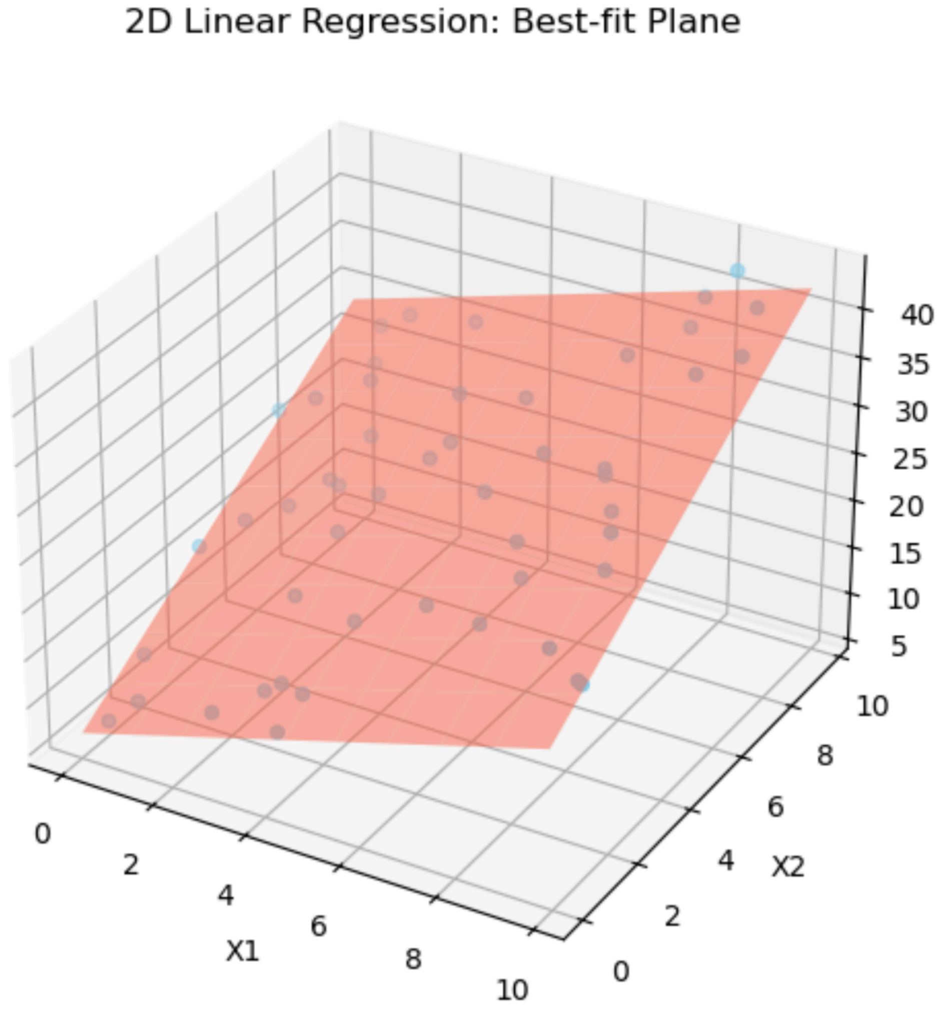
Linear regression is a basic machine learning method that finds the best straight line to describe the relationship between two (or more) variables.

- It answers: “If X changes, how does Y change?”
- The line allows us to **predict** Y for any value of X.
- The best-fit line is as close as possible to all data points.

Linear Regression(1D)



Linear Regression(2D)



Get Correaltions From Titanic

```python
import seaborn as sns

df = sns.load_dataset("titanic")
df["sex"] = df["sex"].map({"male": 0, "female": 1})
corr = df.corr(numeric_only=True)
print(corr["survived"])
```

```
pclass       -0.338481
sex           0.543351
age          -0.077221
sibsp        -0.035322
parch         0.081629
fare          0.257307
adult_male   -0.557080
alone        -0.203367
Name: survived, dtype: float64
```

Titanic Example: Linear Regression

Linear regression models the relationship between features and a continuous target variable.
For Titanic, you might use regression to predict fare or age.

```python
import seaborn as sns
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

df = sns.load_dataset("titanic")
df = df[["survived", "pclass", "sex", "age", "fare", "sibsp", "parch", "alone"]].dropna()
df["sex"] = df["sex"].map({"male": 0, "female": 1})

X = df[["pclass", "sex", "age", "fare", "alone"]]
y = df["survived"]

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

model = LinearRegression()
model.fit(X_train, y_train)

y_pred_reg = model.predict(X_test)

y_pred_class = (y_pred_reg >= 0.5).astype(int)

acc = accuracy_score(y_test, y_pred_class)
print("Linear Regression classification accuracy:", acc)
```

Titanic Example: Logistic Regression

Logistic regression is used for classification, such as predicting survival (yes/no) on the Titanic.

The model outputs the probability of the target being 1.

Logistic Regression Example

```python
import seaborn as sns
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

df = sns.load_dataset("titanic")
df = df[["survived", "pclass", "sex", "age", "fare", "sibsp", "parch"]].dropna()
df["sex"] = df["sex"].map({"male": 0, "female": 1})
df["alive"] = df["alive"].map({"yes": 1, "no": 0})


X = df[["pclass", "sex", "age", "fare", "alone"]]
y = df["alive"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = LogisticRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
print("Test Accuracy:", accuracy_score(y_test, y_pred))
```