LECTURE 11

Machine Learning

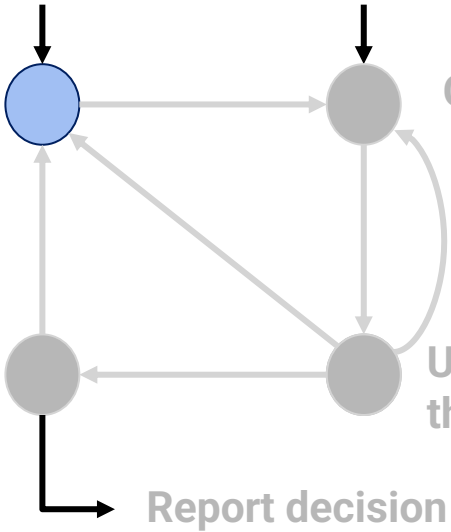
Understanding the usefulness of models and the simple linear regression model

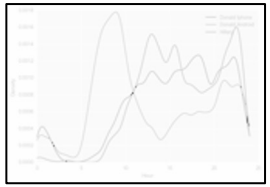
Data Science, Spring 2024 @ Knowledge Stream

Sana Jabbar

Plan for Next Few Lectures: Modeling

Ask a Question

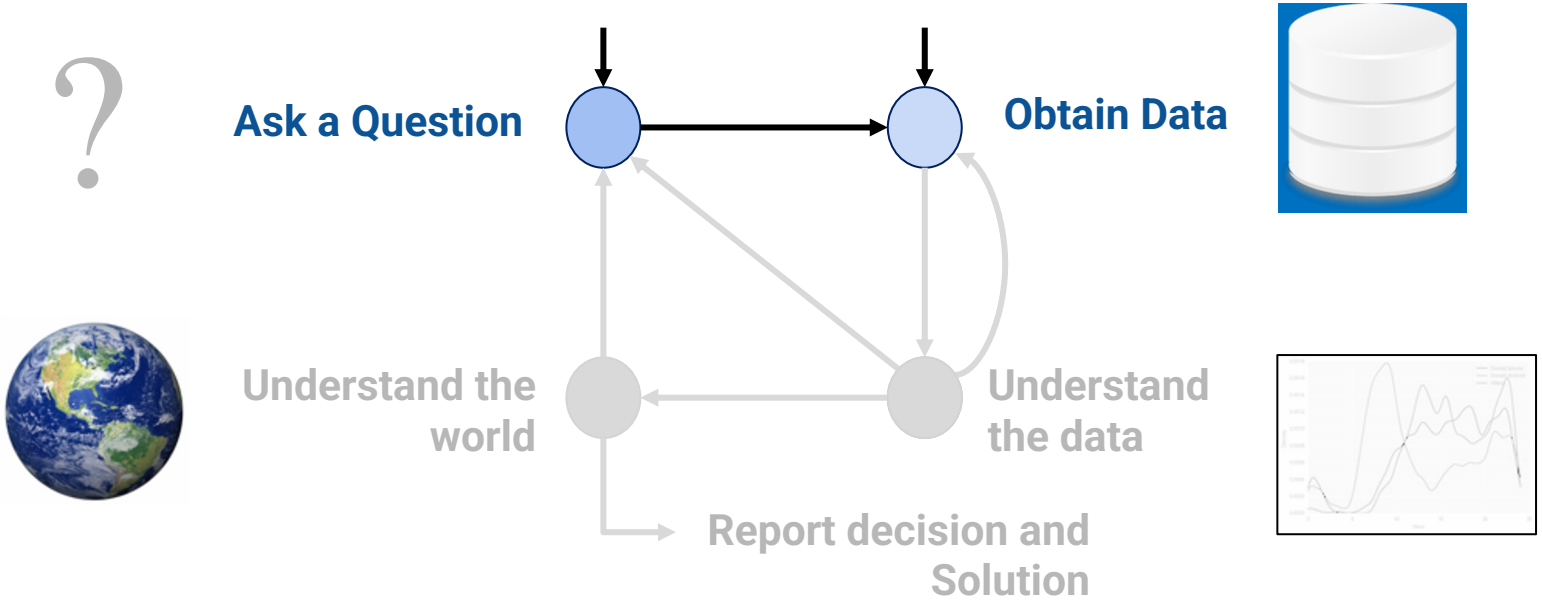




(today)

Modeling I: Intro to Modeling, Simple Linear Regression	→	Modeling II: Different models, loss functions,	→	Modeling III: Multiple Linear Regression

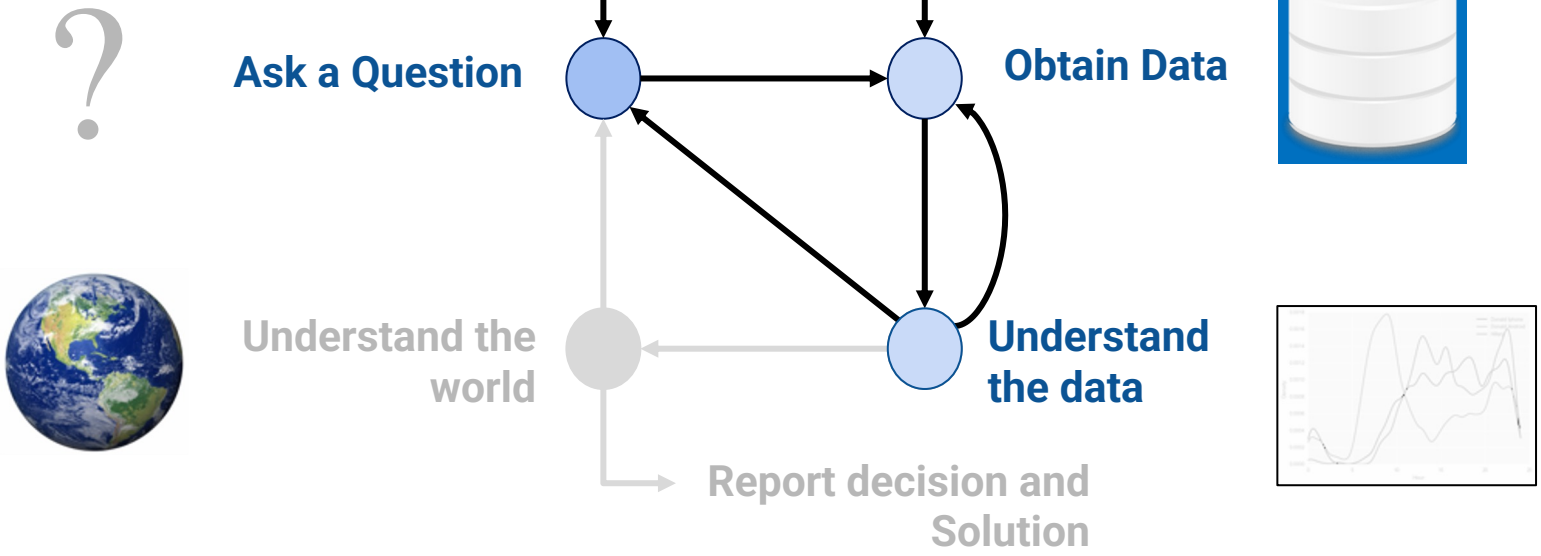
Plan for Next Few Lectures: Modeling



(today)

Modeling I: Intro to Modeling, Simple Linear Regression	→	Modeling II: Different models, loss functions,	→	Modeling III: Multiple Linear Regression

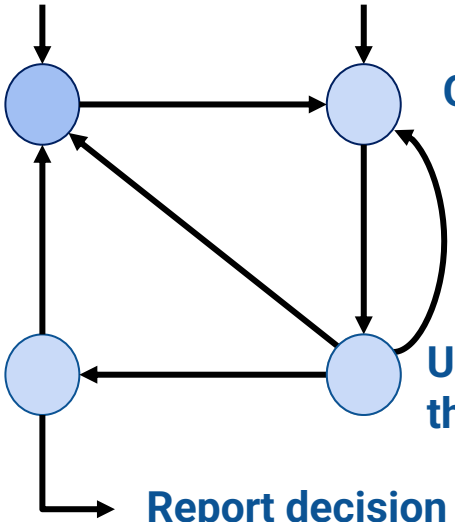
Plan for Next Few Lectures: Modeling

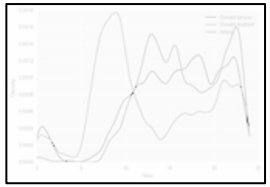


(today)

Modeling I: Intro to Modeling, Simple Linear Regression	→	Modeling II: Different models, loss functions,	→	Modeling III: Multiple Linear Regression

Plan for Next Few Lectures: Modeling





(today)

Modeling I: Intro to Modeling, Simple Linear Regression	→	Modeling II: Different models, loss functions,	→	Modeling III: Multiple Linear Regression

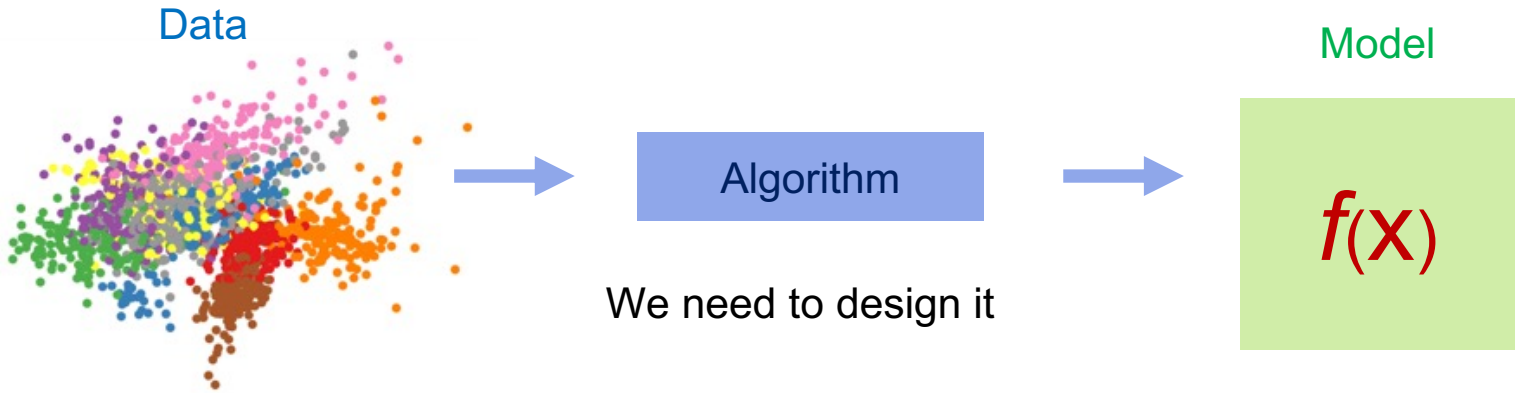
Goals for this Lecture

Lecture 12

- **What is a Model?**
- The Modelling Process
  - Choose a Model
  - Choose a Loss Function
  - Fit the Model
  - Evaluate the Model

Machine learning framework

Given examples (training data), develop a machine learning system to discover patterns



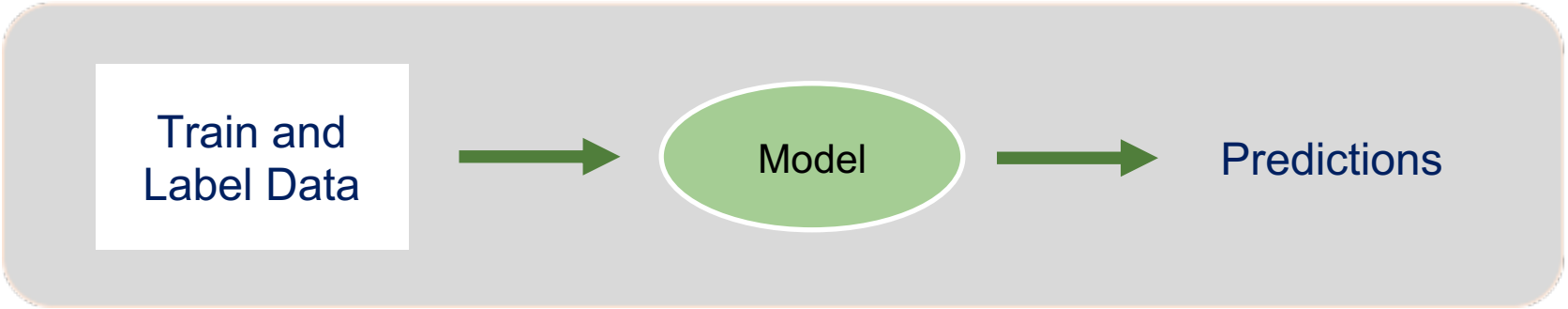
Algorithms vs Model

Linear regression algorithm produces a model, that is, a vector of values of the coefficients of the model.

Neural network along with backpropagation + gradient descent: produces a model comprised of a trained (weights assigned) neural network.

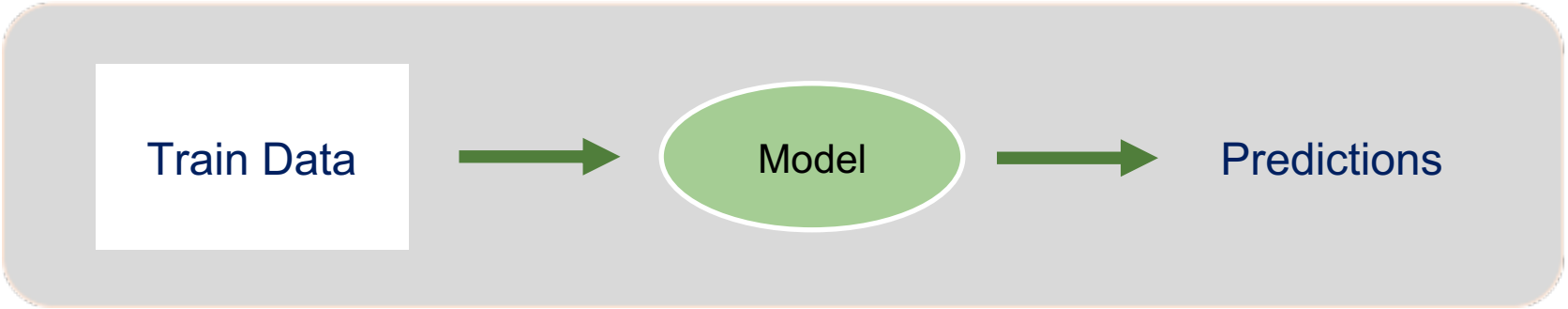
Supervised Learning

The learning algorithm would receive a set of inputs along with the corresponding correct to train a model



Unsupervised Learning

The learning algorithm would receive only a set of inputs to train a model

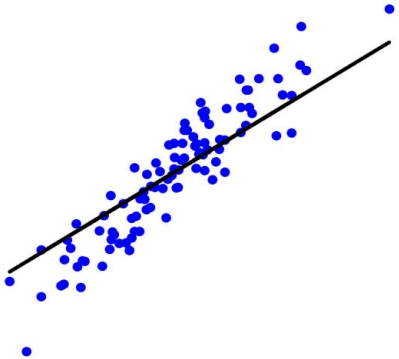


Regression

Quantitative Prediction on a continuous scale

Examples: Prediction of

1. Age of a person from his/her photo
2. Price of 10 Marla, 5-bedroom house in 2050
3. USD/PKR exchange rate after one week
4. Efficacy of any vaccine
5. Average temperature/Rainfall during monsoon
6. Cumulative score in ML course
7. Probability of a decrease in the electricity prices in Pakistan



What do all these problems have in common?

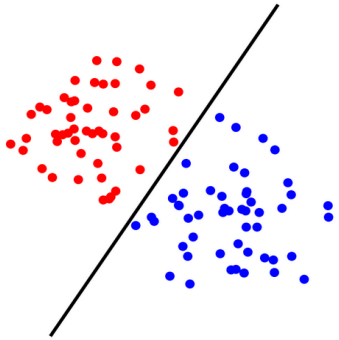
Continuous outputs

Classification:

Predicting a categorical output is called classification

Examples: Prediction of

1. Gender of a person using his/her photo or hand-writing style
2. Spam filtering
3. Temperature/Rainfall normal or abnormal during monsoon
4. Letter grade in a course
5. Decrease expected in electricity prices in Pakistan next year
6. More than 10000 Steps taken today



What do all these problems have in common?
Discrete outputs: Categorical Yes/No (Binary Classification)
Multi-class classification: multiple classes

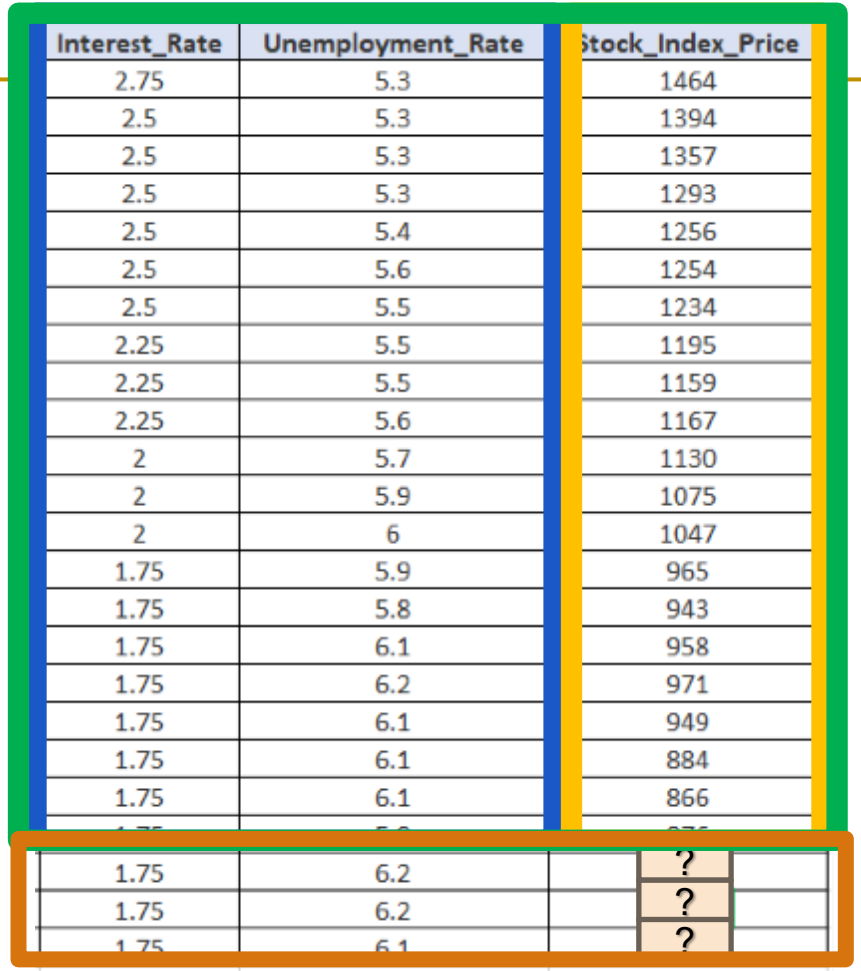
Supervised Learning Setup

- In these regression or classification problems, we have
- Inputs – referred to as Features
- Output – referred to as Label
- Training data – (input, output) for which the output is known and is used for training a model by the ML algorithm
- A Loss, an objective, or a cost function – determines how well a trained model approximates the training data
- Test data – (input, output) for which the output is known and is used for the evaluation of the performance of the trained model

Supervised Learning Setup

Predict Stock Index Price

- Features (Input)
- Labels (Output)
- Training data
- Validation data



Interest_Rate	Unemployment_Rate	Stock_Index_Price
2.75	5.3	1464
2.5	5.3	1394
2.5	5.3	1357
2.5	5.3	1293
2.5	5.4	1256
2.5	5.6	1254
2.5	5.5	1234
2.25	5.5	1195
2.25	5.5	1159
2.25	5.6	1167
2	5.7	1130
2	5.9	1075
2	6	1047
1.75	5.9	965
1.75	5.8	943
1.75	6.1	958
1.75	6.2	971
1.75	6.1	949
1.75	6.1	884
1.75	6.1	866
1.75	6.2	?
1.75	6.2	?
1.75	6.1	?

Using the adopted notation, we can formalize the supervised machine learning setup. We represent the entire training data as

$$D = \{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_n}, y_n)\} \subseteq \mathcal{X}^d \times \mathcal{Y}$$

Here $\mathcal{X}^d$ - d dimensional feature space and $\mathcal{Y}$ is the label space.

Regression: $\mathcal{Y} = \mathbf{R}$ (prediction on continuous scale)

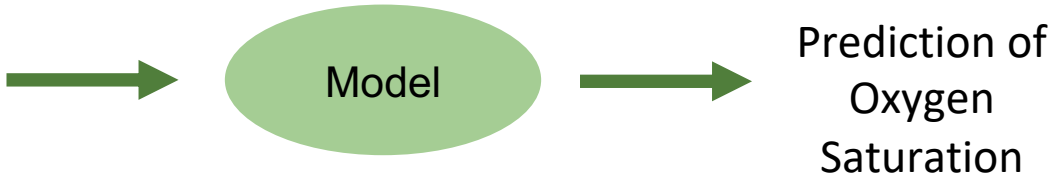
Classification: $\mathcal{Y} = \{0, 1\}$ or $\mathcal{Y} = \{-1, 1\}$ or $\mathcal{Y} = \{1, 2\}$ (Binary classification)

$\mathcal{Y} = \{1, 2, \ldots, M\}$ (M-class classification)

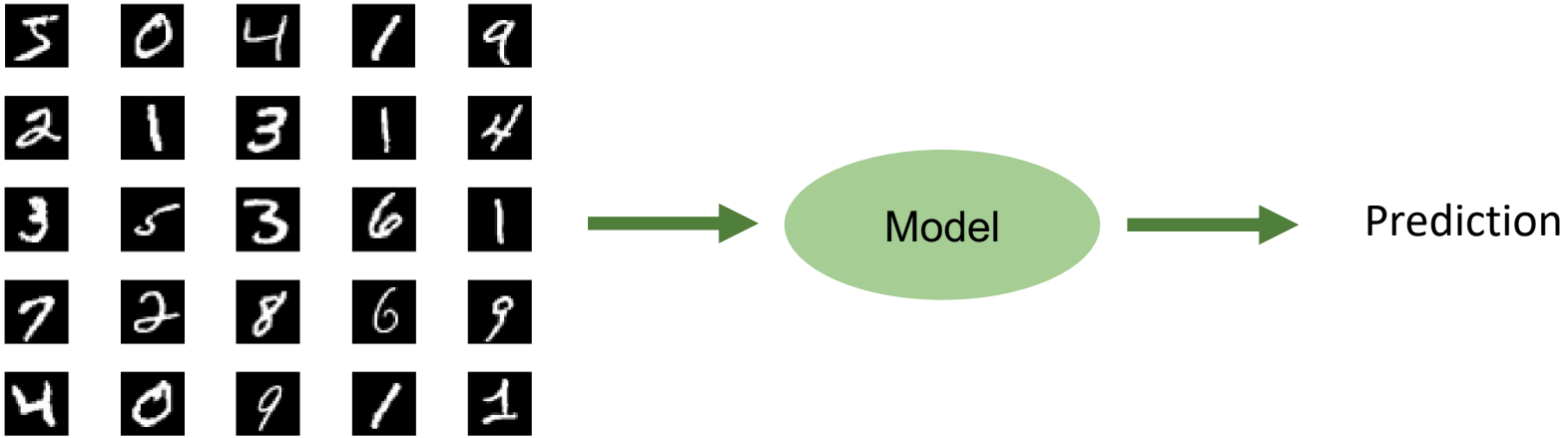
Example:

Data of 200 Patients:

- Age of the patient
- Cholesterol levels
- Glucose levels
- BMI
- Height
- Heart Rate
- Calories intake
- No. of steps taken



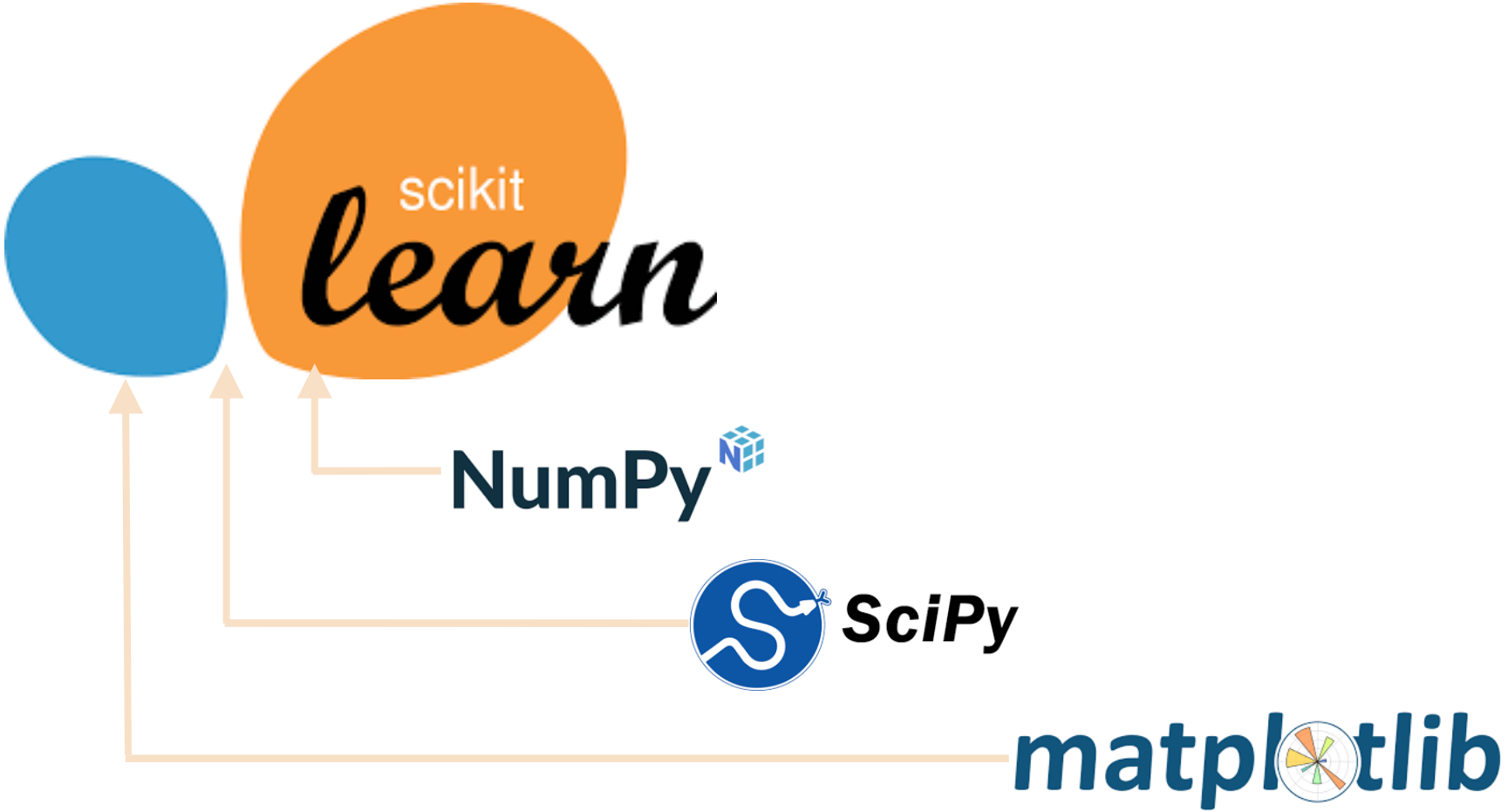
Example:



MNIST Data:

- Each sample 28x28 pixel image
- 60,000 training data
- 10,000 testing data

Scikit-learn



Regression:

from sklearn.linear_model import LinearRegression

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \epsilon$$

- y is the target variable.
- $x_1, x_2, \ldots, x_n$ are the feature variables.
- $\beta_0, \beta_1, \ldots, \beta_n$ are the coefficients to be
- learned.
- ϵ. is the error term.

$$\text{Minimize} \left(\sum_{i=1}^{n} (y_i - \sum_{j=0}^{p} \beta_j x_{ij})^2 \right)$$

Regression:

```
from sklearn.linear_model import LinearRegression

from sklearn.linear_model import Ridge
```

$$y = \beta_0 + \beta_1 x_1 + \beta_2\, x_2 + \cdots + \beta_n x_n + \epsilon$$

- y is the target variable.
- $x_1, x_2, \ldots, x_n$ are the feature variables.
- $\beta_0, \beta_1, \ldots, \beta_n$ are the coefficients to be
- learned.
- ϵ. is the error term.

$$\text{Minimize} \left(\sum_{i=1}^{n} (y_i - \sum_{j=0}^{p} \beta_j x_{ij})^2 + \alpha \sum_{j=1}^{p} \beta_j^2 \right)$$

Regression:

```
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
```

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \epsilon$$

- y is the target variable.
- $x_1, x_2, \ldots, x_n$ are the feature variables.
- $\beta_0, \beta_1, \ldots, \beta_n$ are the coefficients to be
- learned.
- ϵ. is the error term.

$$\text{Minimize } \left(\sum_{i=1}^{n} (y_i - \sum_{j=0}^{p} \beta_j x_{ij})^2 + \alpha \sum_{j=1}^{p} |\beta_j| \right)$$

Regression:

```
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
from sklearn.linear_model import ElasticNet
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
```

Regression

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split


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

# Linear regression
lr = LinearRegression()
lr.fit(X_train, y_train)
y_pred = lr.predict(X_test)
```

Today's task is to repeat the previously provided notebooks with LinearRegression, Ridge, Lasso, ElasticNet

Fit the Model

Lecture 12

- What is a Model?
- **The Modeling Process**
  - Choose a Model
  - Choose a Loss Function
  - **Fit the Model**
  - Evaluate the Model

Minimizing MSE for the SLR Model

Recall: we wanted to pick the **regression line** $\hat{y} = \hat{\theta}_0 + \hat{\theta}_1 x$

To minimize the (sample) **Mean Squared Error**: $MSE(\theta_0, \theta_1) = \frac{1}{n}\sum_{i=1}^{n}(y_i - (\theta_0 + \theta_1 x_i))^2$

To find the best values, we set derivatives equal to zero to **obtain the optimality conditions:**

$$\frac{\partial}{\partial \theta_0} MSE = 0$$

$$\frac{\partial}{\partial \theta_1} MSE = 0$$

Estimating Equations

To find the best values, we set derivatives equal to zero to **obtain the optimality conditions:**

$$0 = \frac{\partial}{\partial \theta_0} MSE = -\frac{2}{n}\sum_{i=1}^{n}\left(y_i - \hat{\theta}_0 - \hat{\theta}_1 x_i\right) \iff \frac{1}{n}\sum_i y_i - \hat{y}_i = 0$$

"Equivalent"

$$0 = \frac{\partial}{\partial \theta_1} MSE = -\frac{2}{n}\sum_{i=1}^{n}\left(y_i - \hat{\theta}_0 - \hat{\theta}_1 x_i\right)x_i \iff \frac{1}{n}\sum_i (y_i - \hat{y}_i)x_i = 0$$

Estimating equations

To find the best θ_0, θ_1, we need to solve the **estimating equations** on the right.

From Estimating Equations to Estimators

Goal: Choose θ_0, θ_1 to solve two estimating equations:

$$\frac{1}{n}\sum_i y_i - \hat{y}_i = 0 \quad \boxed{1} \qquad \text{and} \qquad \frac{1}{n}\sum_i (y_i - \hat{y}_i)x_i = 0 \quad \boxed{2}$$

$\boxed{1}$

$$\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{\theta}_0 - \hat{\theta}_1 x_i\right) = 0 \iff \left(\frac{1}{n}\sum_{i=1}^{n} y_i\right) - \hat{\theta}_0 - \hat{\theta}_1\left(\frac{1}{n}\sum_{i=1}^{n} x_i\right) = 0$$

Separating terms: $\frac{1}{n}\sum_{i=1}^{n} y_i = \bar{y}$, $\frac{1}{n}\sum_{i=1}^{n} x_i = \bar{x}$

$$\iff \bar{y} - \hat{\theta}_0 - \hat{\theta}_1\bar{x} = 0$$

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

From Estimating Equations to Estimators

Goal: Choose θ_0, θ_1 to solve two estimating equations:

$$\frac{1}{n}\sum_i y_i - \hat{y}_i = 0 \quad \boxed{1} \qquad \text{and} \qquad \frac{1}{n}\sum_i (y_i - \hat{y}_i)x_i = 0 \quad \boxed{2}$$

Now, let's try: $\boxed{2}$ - $\boxed{1}$ * $\bar{x}$

$$\frac{1}{n}\sum_i (y_i - \hat{y}_i)x_i - \frac{1}{n}\sum_i (y_i - \hat{y}_i)\bar{x} = 0 \iff \frac{1}{n}\sum_i (y_i - \hat{y}_i)(x_i - \bar{x}) = 0$$

$$\left(\text{using } \hat{y}_i = \hat{\theta}_0 + \hat{\theta}_1 x_i\right) \Rightarrow \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{\theta}_0 - \hat{\theta}_1 x_i\right)(x_i - \bar{x}) = 0$$

$$\left(\text{using } \hat{\theta}_0 = \bar{y} - \hat{\theta}_1 \bar{x}\right) \Rightarrow \frac{1}{n}\sum_i \left(y_i - \bar{y} + \hat{\theta}_1 \bar{x} - \hat{\theta}_1 x_i\right)(x_i - \bar{x}) = 0$$

$$\Rightarrow \frac{1}{n}\sum_i \Big((y_i - \bar{y}) - \hat{\theta}_1 (x_i - \bar{x})\Big)(x_i - \bar{x}) = 0$$

$$\Rightarrow \frac{1}{n}\sum_i \left[(y_i - \bar{y})(x_i - \bar{x}) - \hat{\theta}_1 (x_i - \bar{x})^2\right] = 0$$

$$\Rightarrow \frac{1}{n}\sum_i (y_i - \bar{y})(x_i - \bar{x}) = \hat{\theta}_1 \frac{1}{n}\sum_i (x_i - \bar{x})^2$$

Plug in definitions of correlation and SD:

$$r\sigma_y\sigma_x = \hat{\theta}_1\sigma_x^2$$

Solve for $\hat{\theta}_1$:

$$\hat{\theta}_1 = r\frac{\sigma_y}{\sigma_x}$$

Reminder

$$\sigma_x^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

$$r = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{\sigma_x}\right)\left(\frac{y_i - \bar{y}}{\sigma_y}\right)$$

Estimating Equations

Estimating equations are the equations that the model fit has to solve. They help us:

- Derive the estimates.
- Understand what our model is paying attention to.

$$\frac{1}{n}\sum_i y_i - \hat{y}_i = 0$$

$$\frac{1}{n}\sum_i (y_i - \hat{y}_i)x_i = 0$$

For SLR:

- The residuals should **average to zero** (otherwise we should fix the intercept!)

The Modeling Process

1. Choose a model ✅

How should we represent the world?

$$\hat{y} = \theta_0 + \theta_1 x$$

SLR model

2. Choose a loss function ✅

How do we quantify prediction error?

$$L(y, \hat{y}) = (y - \hat{y})^2$$

Squared loss

3. Fit the model ✅

How do we choose the best parameters of our model given our data?

$$\hat{R}(\theta) = \frac{1}{n} \sum_{i=1}^{n} (y_i - (\theta_0 + \theta_1 x))^2$$

MSE for SLR

4. Evaluate model performance

How do we evaluate whether this process gave rise to a good model?

$$\hat{y} = \hat{\theta}_0 + \hat{\theta}_1 x \quad \begin{cases} \hat{\theta}_0 = \bar{y} - \hat{\theta}_1 \bar{x} \\ \hat{\theta}_1 = r \dfrac{\sigma_y}{\sigma_x} \end{cases}$$

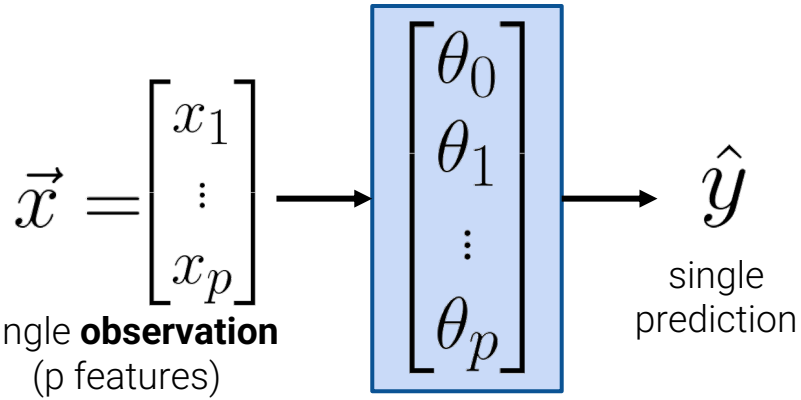
Multi Linear Regression

Multiple Linear Regression

Define the **multiple linear regression** model:

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p$$

Predicted value of y



NBA 2018-2019 Dataset

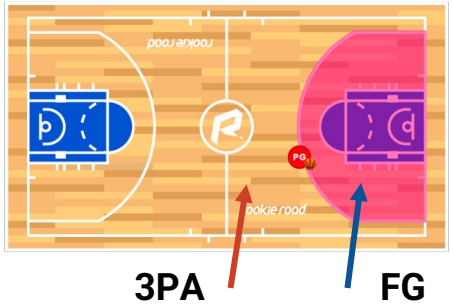
How many points does an athlete score per game?
PTS (average points/game)

To name a few factors:

- **FG**: average # 2 point field goals
- **AST**: average # of assists
- **3PA**: average # 3 point field goals attempted

	FG	AST	3PA	PTS
1	1.8	0.6	4.1	5.3
2	0.4	0.8	1.5	1.7
3	1.1	1.9	2.2	3.2
4	6.0	1.6	0.0	13.9
5	3.4	2.2	0.2	8.9
6	0.6	0.3	1.2	1.7

Rows correspond to individual players.



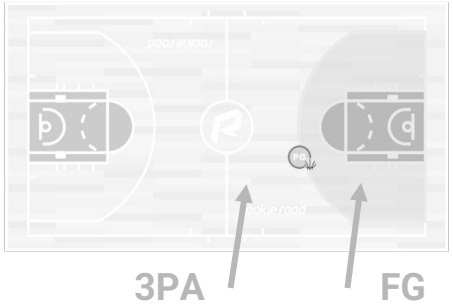
assist: a pass to a teammate that directly leads to a goal

Multiple Linear Regression Model

How many points does an athlete score per game?
PTS (average points/game)

To name a few factors:

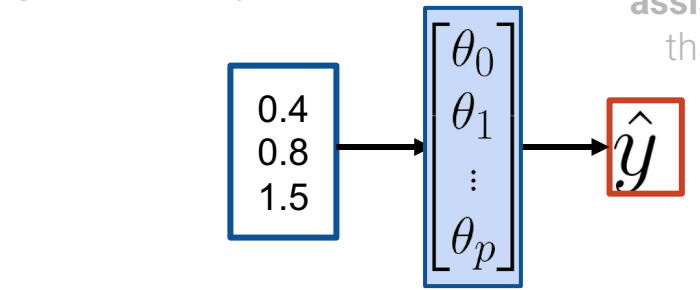
- **FG**: average # 2 point field goals
- **AST**: average # of assists
- **3PA**: average # 3 point field goals attempted



assist: a pass to a teammate that directly leads to a goal

	FG	AST	3PA	PTS
1	1.8	0.6	4.1	5.3
2	0.4	0.8	1.5	1.7
3	1.1	1.9	2.2	3.2
4	6.0	1.6	0.0	13.9
5	3.4	2.2	0.2	8.9
6	0.6	0.3	1.2	1.7

Rows correspond to individual players.



$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p$$
$$= \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$

FG AST 3PA

Today's Goal: Ordinary Least Squares

1. Choose a model	**Multiple Linear Regression**
2. Choose a loss function	L2 Loss **Mean Squared Error (MSE)**
3. Fit the model	Minimize average loss with ~~calculus~~ geometry
4. Evaluate model performance	Visualize, ~~Root MSE~~ Multiple R^2

In statistics, this model + loss is called **Ordinary Least Squares (OLS)**.

The solution to OLS are the minimizing loss for parameters $\hat{\theta}$, also called the **least squares estimate**.

Today's Goal: Ordinary Least Squares

1. Choose a model

Multiple Linear Regression

For each of our n data points:

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p$$

$$\hat{\mathbb{Y}} = \mathbb{X}\boldsymbol{\theta}$$

Linear Algebra!!

2. Choose a loss function

L2 Loss

Mean Squared Error (MSE)

3. Fit the model

Minimize average loss with ~~calculus~~ geometry

4. Evaluate model performance

Visualize,
~~Root MSE~~
Multiple R^2

From One Feature to Many Features

Dataset for SLR

x	y
x_1	y_1
x_2	y_2
$\vdots$	$\vdots$
x_n	y_n

	FG	PTS
1	1.8	5.3
2	0.4	1.7
3	1.1	3.2
4	6.0	13.9
5	3.4	8.9
...	...	...

Dataset for Multiple Linear Regression

$x_{:,1}$	$x_{:,2}$		$x_{:,p}$	y
x_{11}	x_{12}	$\cdots$	x_{1p}	y_1
x_{21}	x_{22}	$\cdots$	x_{2p}	y_2
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
x_{n1}	x_{n2}	$\cdots$	x_{np}	y_n

	FG	AST	3PA	PTS
1	1.8	0.6	4.1	5.3
2	0.4	0.8	1.5	1.7
3	1.1	1.9	2.2	3.2
4	6.0	1.6	0.0	13.9
5	3.4	2.2	0.2	8.9
...	...	...	...	...

From One Feature to Many Features

Dataset for Multiple Linear Regression

$x_{:1}$	$x_{:2}$		$x_{:p}$	y
x_{11}	x_{12}	$\cdots$	x_{1p}	y_1
x_{21}	x_{22}	$\cdots$	x_{2p}	y_2
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
x_{n1}	x_{n2}	$\cdots$	x_{np}	y_n

Feature 2

$$\{x_{12}, x_{22}, \ldots x_{n2}\}$$

Observation i

$$\{x_{i1}, x_{i2}, \ldots x_{ip}, y_i\}$$

	FG	AST	3PA	PTS
1	1.8	0.6	4.1	5.3
2	0.4	0.8	1.5	1.7
3	1.1	1.9	2.2	3.2
4	6.0	1.6	0.0	13.9
5	3.4	2.2	0.2	8.9
...	...	...	...	...

Model

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p$$

$$\begin{cases} \hat{y}_1 = \theta_0 + \theta_1 x_{11} + \theta_2 x_{12} + \cdots + \theta_p x_{1p} \\ \hat{y}_2 = \theta_0 + \theta_1 x_{21} + \theta_2 x_{22} + \cdots + \theta_p x_{2p} \\ \vdots \\ \hat{y}_n = \theta_0 + \theta_1 x_{n1} + \theta_2 x_{n2} + \cdots + \theta_p x_{np} \end{cases}$$

Vector Notation

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p$$

This part looks a little like a dot product…

We want to collect all the θ_i's into a single vector.

$$= \theta_0 + \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} = \theta_0 \cdot 1 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p$$

😤What about this one???

Vector Notation

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p$$

$$= \theta_0 \cdot 1 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p$$

$$= \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{bmatrix} \cdot \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} = x^\top \theta$$

bias term,
intercept term

We want to collect all the θ_i's into a single vector.

Matrix Notation

$$\begin{cases} \hat{y}_1 = \theta_0 + \theta_1 x_{11} + \theta_2 x_{12} + \cdots + \theta_p x_{1p} \\ \hat{y}_2 = \theta_0 + \theta_1 x_{21} + \theta_2 x_{22} + \cdots + \theta_p x_{2p} \\ \vdots \\ \hat{y}_n = \theta_0 + \theta_1 x_{n1} + \theta_2 x_{n2} + \cdots + \theta_p x_{np} \end{cases}$$

$$\begin{cases} \hat{y}_1 = x_1^\top \theta \quad \text{where } x_1^\top = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \end{bmatrix} \text{is datapoint/observation 1} \\ \hat{y}_2 = x_2^\top \theta \quad \text{where } x_2^\top = \begin{bmatrix} 1 & x_{21} & x_{22} & \cdots & x_{2p} \end{bmatrix} \text{is datapoint/observation 2} \\ \vdots \\ \hat{y}_n = x_n^\top \theta \quad \text{where } x_n^\top = \begin{bmatrix} 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} \text{is datapoint/observation n} \end{cases}$$

Matrix Notation

$$\begin{cases} \hat{y}_1 = x_1^\top \theta & \text{where } x_1^\top = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \end{bmatrix} \text{is datapoint/observation 1} \\ \hat{y}_2 = x_2^\top \theta & \text{where } x_2^\top = \begin{bmatrix} 1 & x_{21} & x_{22} & \dots & x_{2p} \end{bmatrix} \text{is datapoint/observation 2} \\ \vdots & \\ \hat{y}_n = x_n^\top \theta & \text{where } x_n^\top = \begin{bmatrix} 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} \text{is datapoint/observation n} \end{cases}$$

	FG	AST	3PA	PTS
1	1.8	0.6	4.1	5.3
2	0.4	0.8	1.5	1.7
3	1.1	1.9	2.2	3.2
4	6.0	1.6	0.0	13.9
5	3.4	2.2	0.2	8.9
...	...	...	...	...

For data point/observation 2, we have

$$x_2 = \begin{bmatrix} 1 \\ 0.4 \\ 0.8 \\ 1.5 \end{bmatrix} \quad y_2 = 1.7 \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

$$\begin{aligned} \hat{y}_2 &= x_2^\top \theta \\ &= \theta_0 + \theta_1 \cdot 0.4 + \theta_2 \cdot 0.8 + \theta_3 \cdot 1.5 \end{aligned}$$

Dimension check

$x_2 \in \mathbb{R}^4 \text{or } \mathbb{R}^{(p+1)}$

$\theta \in \mathbb{R}^4 \text{or } \mathbb{R}^{(p+1)}$

$y_2 \in \mathbb{R} \quad \hat{y}_2 \in \mathbb{R}$

also called scalars

Matrix Notation

$$\begin{aligned}
\hat{y}_1 &= \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \end{bmatrix} \theta &= x_1^T \theta \\
\hat{y}_2 &= \begin{bmatrix} 1 & x_{21} & x_{22} & \dots & x_{2p} \end{bmatrix} \theta &= x_2^T \theta \\
&\vdots & & \vdots & & \vdots \\
\hat{y}_n &= \begin{bmatrix} 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} \theta &= x_n^T \theta
\end{aligned}$$

n row vectors, each with dimension **(p+1)**

Expand out each datapoint's (transposed) input

Matrix Notation

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ & & \vdots & & \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} \theta$$

n row vectors, each with dimension **(p+1)**

Vectorize predictions and parameters to encapsulate all n equations into a single matrix equation.

Matrix Notation

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \mathbb{X} \theta$$

Design matrix with dimensions n x (p + 1)

The Design Matrix $\mathbb{X}$

We can use linear algebra to represent our predictions of all n data points at once.

One step in this process is to stack all of our input features together into a **design matrix**:

$$\mathbb{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ 1 & x_{31} & x_{32} & \dots & x_{3p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix}$$

What do the **rows** and **columns** of the design matrix represent in terms of the observed data?

	Field Goals	Assists	3-Point Attempts	
Bias	**FG**	**AST**	**3PA**	**PTS**
1	1.8	0.6	4.1	5.3
1	0.4	0.8	1.5	1.7
1	1.1	1.9	2.2	3.2
1	6.0	1.6	0.0	13.9
1	3.4	2.2	0.2	8.9
...	...	...	...	...
1	4.0	0.8	0.0	11.5
1	3.1	0.9	0.0	7.8
1	3.6	1.1	0.0	8.9
1	3.4	0.8	0.0	8.5
1	3.8	1.5	0.0	9.4

Example design matrix
708 rows x (3+1) cols

The Design Matrix $\mathbb{X}$

We can use linear algebra to represent our predictions of all n data points at once.

One step in this process is to stack all of our input features together into a **design matrix**:

$$\mathbb{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ 1 & x_{31} & x_{32} & \dots & x_{3p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix}$$

A **row** corresponds to one **observation**, e.g., all (p+1) features for datapoint 3

A **column** corresponds to a **feature**, e.g. feature 1 for all n data points

Special all-ones feature often called the **bias/intercept**

Bias	FG (Field Goals)	AST (Assists)	3PA (3-Point Attempts)	PTS
1	1.8	0.6	4.1	5.3
1	0.4	0.8	1.5	1.7
1	1.1	1.9	2.2	3.2
1	6.0	1.6	0.0	13.9
1	3.4	2.2	0.2	8.9
...	...	...	...	...
1	4.0	0.8	0.0	11.5
1	3.1	0.9	0.0	7.8
1	3.6	1.1	0.0	8.9
1	3.4	0.8	0.0	8.5
1	3.8	1.5	0.0	9.4

Example design matrix
708 rows x (3+1) cols

The Multiple Linear Regression Model Using Matrix Notation

We can express our linear model on our entire dataset as follows:

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ 1 & x_{31} & x_{32} & \dots & x_{3p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_p \end{bmatrix}$$

$$\hat{\mathbb{Y}} = \mathbb{X}\theta$$

Prediction vector $\mathbb{R}^n$

Design matrix $\mathbb{R}^{n \times (p+1)}$

Parameter vector $\mathbb{R}^{(p+1)}$

Note that our **true output** is also a vector:

$$\mathbb{Y} \in \mathbb{R}^n$$

Today's Goal: Ordinary Least Squares

1. Choose a model ✅	Multiple Linear Regression
2. Choose a loss function	L2 Loss Mean Squared Error (MSE)
3. Fit the model	Minimize average loss with ~~calculus~~ geometry
4. Evaluate model performance	Visualize, ~~Root MSE~~ Multiple R^2

$$\hat{\mathbb{Y}} = \mathbb{X}\theta$$

$$R(\theta) = \frac{1}{n}||\mathbb{Y} - \mathbb{X}\theta||_2^2$$

More Linear Algebra!!

[Linear Algebra] Vector Norms and the L2 Vector Norm

The **norm** of a vector is some measure of that vector's **size/length**.

- The two norms we need to know are the L_1 and L_2 norms (sound familiar?).
- Today, we focus on L_2 norm. We'll define the L_1 norm another day.

For the n-dimensional vector $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$ is , the **L2 vector norm**

$$||\vec{x}||_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} = \sqrt{\sum_{i=1}^{n} (x_i^2)}$$

Mean Squared Error with L2 Norms

We can rewrite mean squared error as a squared L2 norm:

$$R(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$= \frac{1}{n}||\mathbb{Y} - \hat{\mathbb{Y}}||_2^2$$

With our linear model $\hat{\mathbb{Y}} = \mathbb{X}\theta$:

$$R(\theta) = \frac{1}{n}||\mathbb{Y} - \mathbb{X}\theta||_2^2$$

Ordinary Least Squares

The **least squares estimate** $\hat{\theta}$ is the parameter that **minimizes** the objective function $R(\theta)$:

$$R(\theta) = \frac{1}{n}||\mathbb{Y} - \mathbb{X}\theta||_2^2$$

How should we interpret the OLS problem?

A. Minimize the mean squared error for the linear model $\hat{\mathbb{Y}} = \mathbb{X}\theta$

B. Minimize the **distance** between true and predicted values $\mathbb{Y}$ and $\hat{\mathbb{Y}}$

C. Minimize the **length** of the residual vector, $e = \mathbb{Y} - \hat{\mathbb{Y}} = \begin{bmatrix} y_1 - \hat{y}_1 \\ y_2 - \hat{y}_2 \\ \vdots \\ y_n - \hat{y}_n \end{bmatrix}$

D. All of the above

E. Something else

Ordinary Least Squares

The **least squares estimate** $\hat{\theta}$ is the parameter that **minimizes** the objective function $R(\theta)$:

$$R(\theta) = \frac{1}{n}||\mathbb{Y} - \mathbb{X}\theta||_2^2$$

How should we interpret the OLS problem?

A. Minimize the mean squared error for the linear model $\hat{\mathbb{Y}} = \mathbb{X}\theta$

B. Minimize the **distance** between true and predicted values $\mathbb{Y}$ and $\hat{\mathbb{Y}}$

C. Minimize the **length** of the residual vector, $e = \mathbb{Y} - \hat{\mathbb{Y}} = \begin{bmatrix} y_1 - \hat{y}_1 \\ y_2 - \hat{y}_2 \\ \vdots \\ y_n - \hat{y}_n \end{bmatrix}$ Important for today

D. All of the above

E. Something else

Today's Goal: Ordinary Least Squares

1. Choose a model ✅

Multiple Linear Regression

$$\hat{\mathbb{Y}} = \mathbb{X}\theta$$

2. Choose a loss function ✅

L2 Loss

Mean Squared Error (MSE)

$$R(\theta) = \frac{1}{n}||\mathbb{Y} - \mathbb{X}\theta||_2^2$$

3. Fit the model

Minimize average loss with ~~calculus~~ geometry

The calculus derivation requires matrix calculus (out of scope, but here's a link if you're interested).

Instead, we can derive $\hat{\theta}$ using a **geometric argument**.

4. Evaluate model performance

Visualize,
~~Root MSE~~
Multiple R^2

$$\hat{\theta} = \left(\mathbb{X}^{\mathsf{T}}\mathbb{X}\right)^{-1}\mathbb{X}^{\mathsf{T}}\mathbb{Y}$$

This result is so important that it deserves its own slide.

It is the **least squares estimate** and the solution to the normal equation $\mathbb{X}^T\mathbb{X}\hat{\theta} = \mathbb{X}^T\mathbb{Y}$

.

$$\hat{\theta} = \left(\mathbb{X}^\intercal \mathbb{X}\right)^{-1} \mathbb{X}^\intercal \mathbb{Y}$$

This result is so important that it deserves its own slide.

It is the **least squares estimate** and the solution to the normal equation $\mathbb{X}^T\mathbb{X}\hat{\theta} = \mathbb{X}^T\mathbb{Y}$

.