Day 1: (Machine Learning Engineering)

Sangeetha Venkatesan

(Models)

Structure: Linear Regression

Logistic Regression

Model Evaluation: Metrics (how well the models behave)

Model Deployment: (Flask, Docker, venv)

(Bentoml)

Tree Based Models: (Credit Risk)

Steps:

1) Find a dataset
2) Explaining how a model could be used
3) EDA, features
4) Training multiple models
5) Put Model in Webservice
6) put the service to the cloud

Neural networks: Image classification. (Pytorch)

Serverless Deeplearning. (Aws lambda)

Kubernetes

KServe (Make Model Predictions)

1.1 Intro to Machine Learning:

Car Price Prediction: (How can we come up

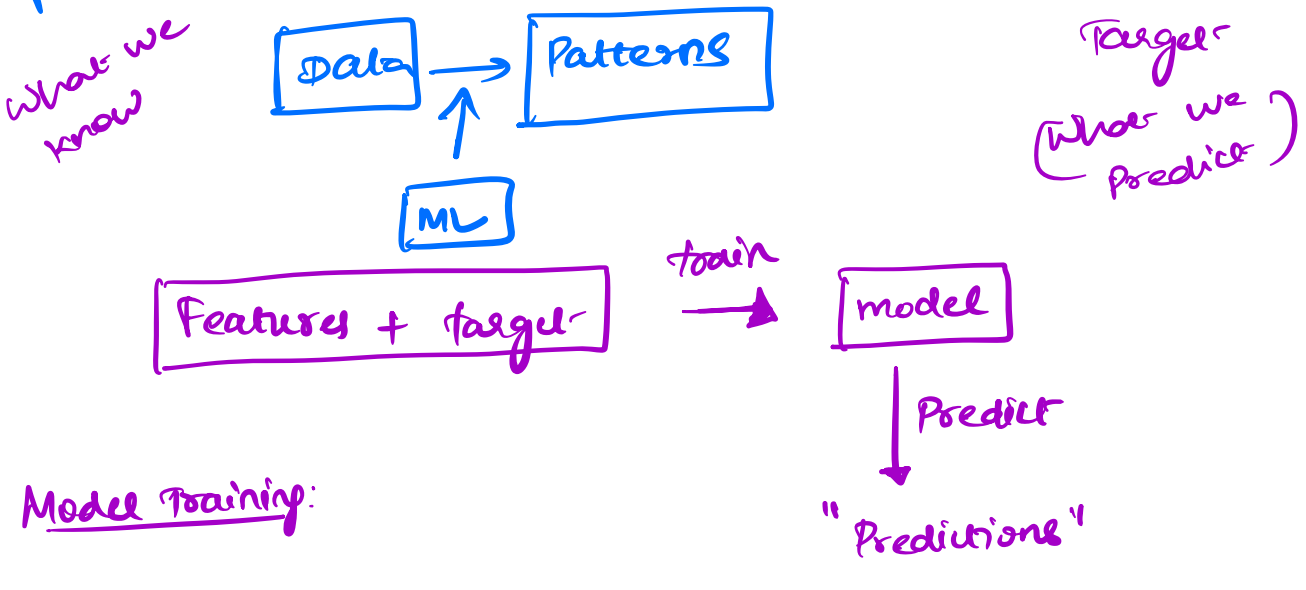
with best price?)

Think about what we know? (Features)

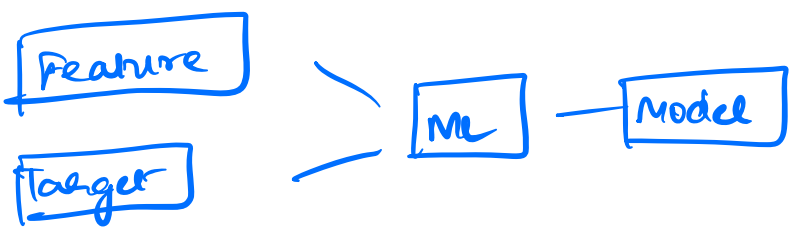
Features → Year, make, Mileage → Predict (Price)

What we know

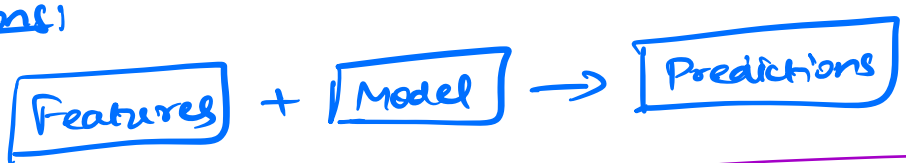
Target (What we predict)



Model Training:



Predictions:



1.2 Machine Learning vs Rule Based

Spam Detection: (Use case) (Classification)

Rule Based

Question: What makes the message SPAM?

1) Sequence of if, else

if sender = promotions then "spam" if title contains "tax
else "not spam" (we end up in cycle of updates of Rules)

Machine Learning:

- Get data
- Define & calculate features (title, body, sender, sender domain, description)
- Train and use the model

Start with Rules & use Rules as features for machine learning

Email → encode → feature array [1, 1, 0, 0, 1]

- 1 → length of title
- 1 → length of body
- 0 → sender name
- 1 → description
- sender domain

Features (data) | Target (desired output)

target: 1

Apply:

Model

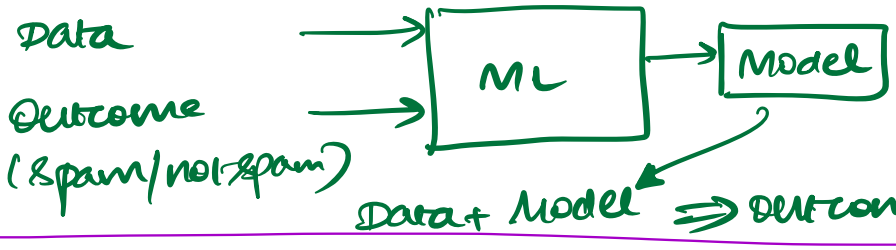
features (data)	Predictions (output)	Final outcome (decision)
[0, 0, 0, 1, 0, 1]	0.8	SPAM
[0, 0, 0, 1, 1, 0]	0.6	GOOD
	0.1	SPAM

Threshold ≥ 0.5

Rule:

CODE, DATA → Software → outcome

ML Basics:



1.3 Supervised Machine Learning:

(We show the model what the label is)

↑ use this to learn patterns extracted

↕

Predict.

Features + (Target) ← we tell & teach

(Desired output)

$$\begin{bmatrix} [1,1,0,0,0,1] \\ [0,0,0,1,1,1] \end{bmatrix} \quad \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

↗ feature matrix X

y (vector)

Rows: Observations

Columns: features that we want to extract.

training

features

$g(X) \simeq y \rightarrow$ target

↳ model that produces approx closes possible to target

the output can be probability

(Predictions (output))

Supervised Machine learning:

Regression: (outputs a number)

Car ⟹ $50 K

Classification: (category)

(teaching by showing examples [X & y])

→ Multiclass: (multiclass)

Binary

Ranking: (Recommender system)

0.7, 0.6, 0.5 ... (top 6)

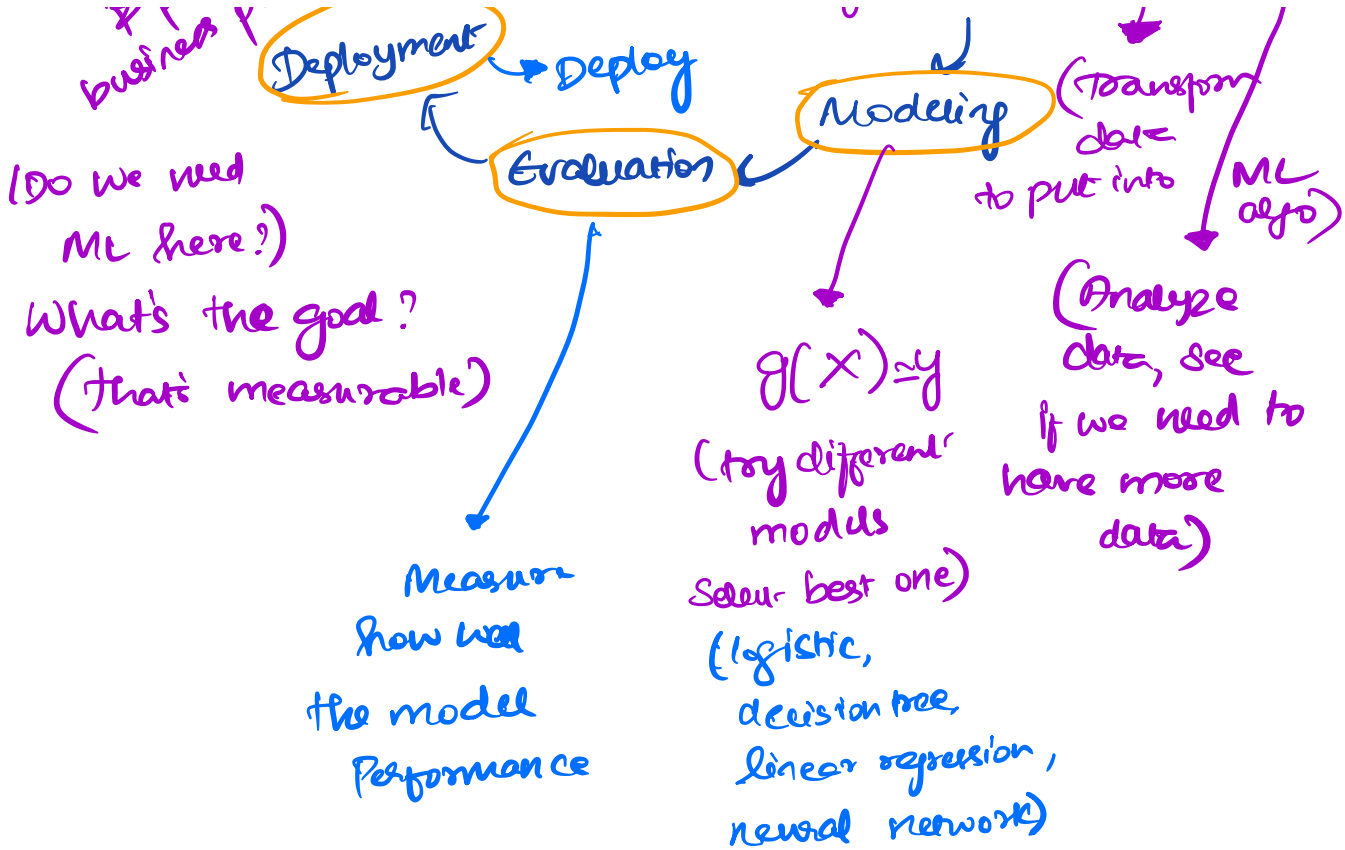
(search results)

1-4 CRISP-DM (Cross Industry Standard Process - Data Mining)

Methodology for organizing ML projects.

Problem understanding → Deployment

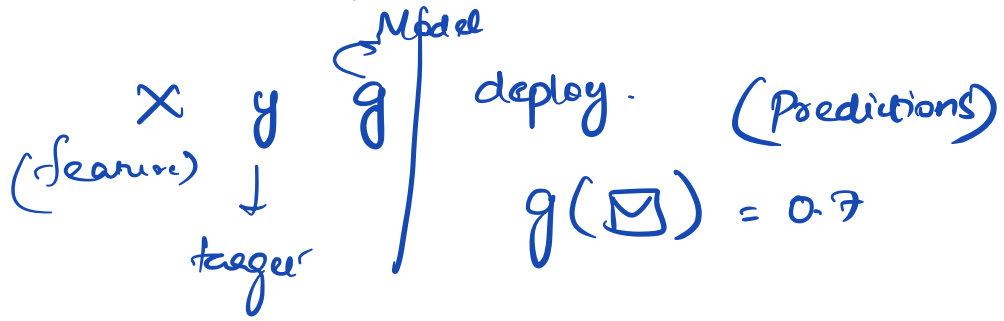
ML Projects:



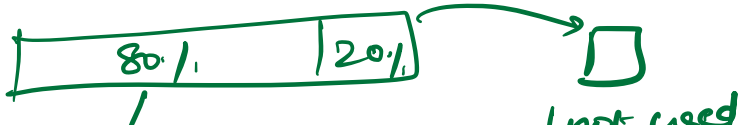
Start simple → learn from feedback → Improve

1-5 Model Selection Process

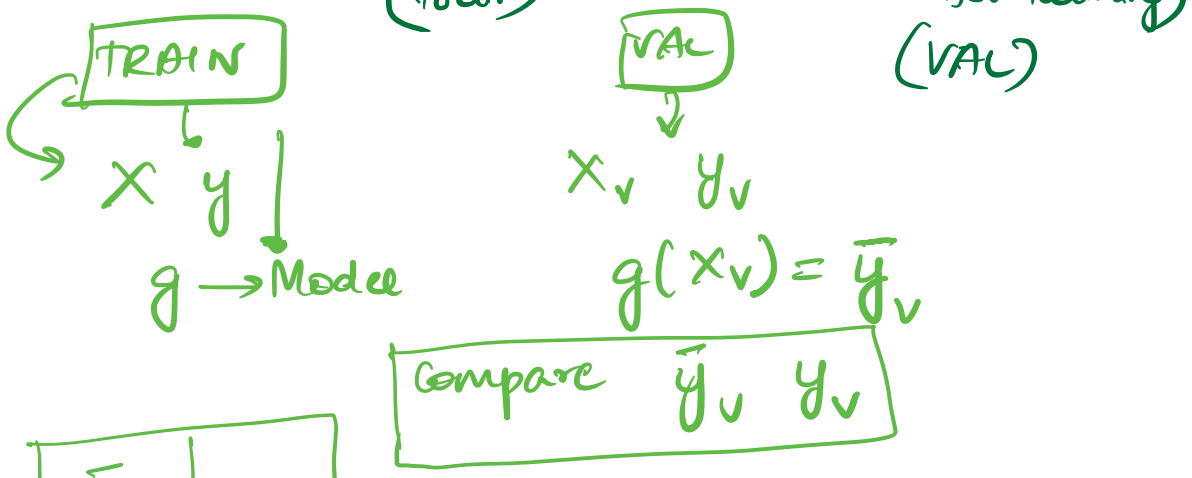
Modeling step: **Selecting the best Model:**



Holdout + train



(train) (for training)



(VAL)

	$\bar{y}_v$	y_v
1	0.8	1
0	0.7	0
0	0.6	1
1	0.8	0

(Pred) (target)

$\frac{4}{6}$ are correct = 66.1%

Select the model based on above evaluation

g_1	Logistic Regression	66.1%
g_2	Decision Tree	60.1%
g_3	Random Forest	67.1%
g_4	neural N/w	88.1%

Best Accuracy ←

Multiple comparisons: (try different Models and rate against same valid dataset)

Validation & Test (Hold out 2 dataset)

60

Hide the dataset.

1) Perform Model selection ← Valid Test

2) Select the best Model

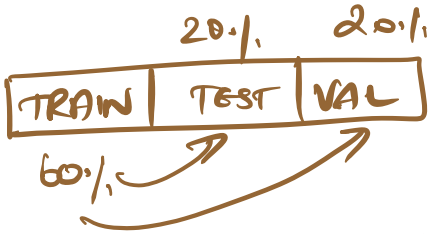
3) Apply the above Model on Test dataset (next round of validation)

Logistic Regression	66.%
Decision Tree	60.%
Random Forest	67.%
Neural Network	80.%

Take test dataset & we will apply on Neural network

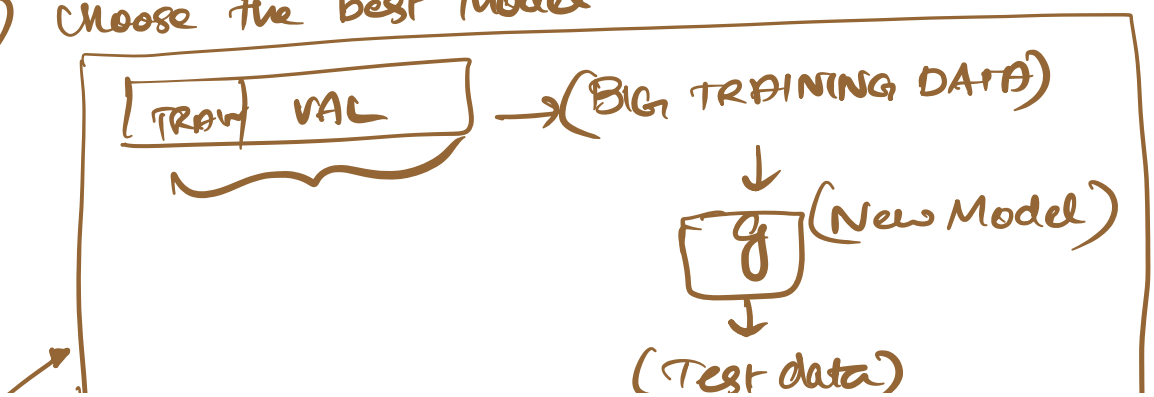
$\simeq$ 79.%

1) Split the dataset
2) Train the model
3) Validate the model
} Repeat for different models
4) Select the best one
5) Test data
6) Check



1) Perform Model Solution process

2) Choose the best Model



In this way we don't throw away the validation data set.