

Supervised Learning



Agenda

1. Data understanding (load, desc stats, info)
 - a. `pd.read_csv`, `df.info()`, `df.describe()`
2. Train test split
 - a. `train_test_split`
3. Data preprocessing (feature engineering)
 - a. Transformation (normal, standard)
 - b. `OneHotEncoder`
 - c. `OrdinalEncoder`
4. Model selection (5 models)
5. Model evaluation and cross validation (metrics, `cross_val_score`)
6. Feature importance & feature selection
7. Hyperparams tuning
8. Pipeline



What is Machine Learning?

- Field of study that gives computers the ability to learn (from data) without being explicitly programmed.
- Examples: Stock price prediction, spam/not spam email classification, customer segmentation (clustering), fraud detection

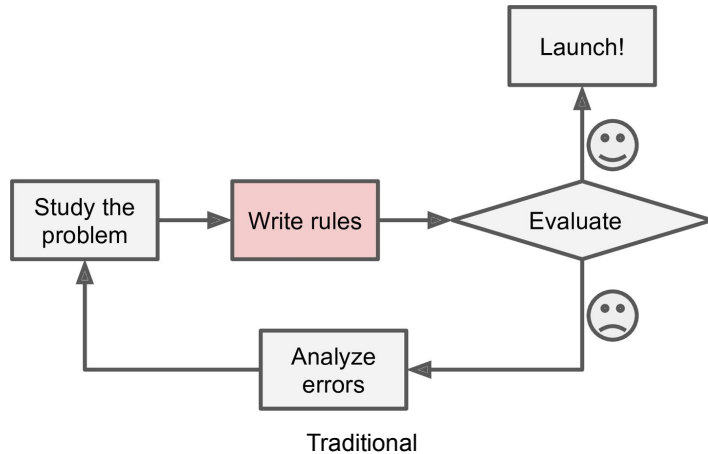


AI vs ML vs DS

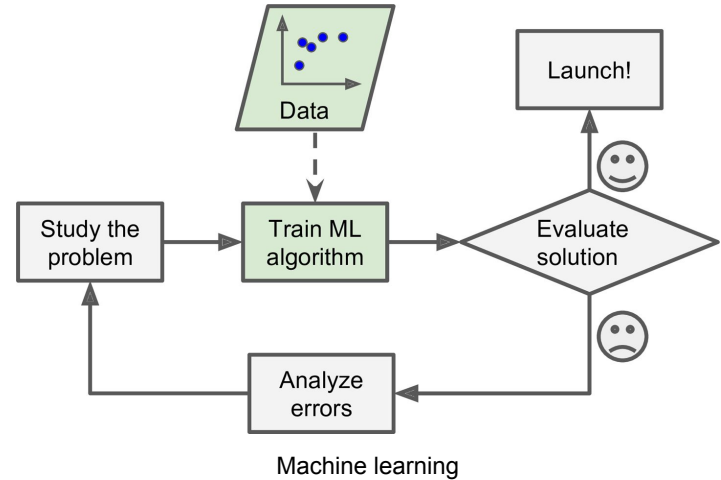
- AI is an area of computer science where the goal is to enable computers and machines to perform human-like task and **simulate human behavior**
- ML is a subset of AI that tries to solve a specific problem and make predictions using data
- DS is a field that attempts to find patterns and draw insights from data



Traditional vs Machine Learning : Spam Email Detection



1. Human manually identify characteristics of spam/not spam email eg: "mata minus", "berat badan"
2. Create rules from point 1
3. Rules/program become long and complex as spam email varies



1. Human created *training set* of labelled email (spam/not spam)
2. Machine learning will find pattern of spam email
3. Short and easy to maintain program

Types of machine learning

Supervised

In supervised learning, we have several data points or samples, described using predictor variables or features (X) and a target variable or **label** (Y).

Supervised Learning

x_1	x_2	x_3	x_p	Y

Target

Example

1. Spam/not spam
2. Stock price (actual/prediction)
3. Flower classification

Unsupervised

Uncovering hidden patterns and structures from **unlabeled data**.

Un-Supervised Learning

x_1	x_2	x_3	x_p	Y

No Target

Example

1. Customer segmentation
2. Dimension reduction
3. Feature selection

Labeled vs Unlabeled



Supervised Learning: Classification & Regression

Features/predictors

User ID	Gender	Age	Salary	Purchased
15624510	Male	19	19000	0
15810944	Male	35	20000	1
15668575	Female	26	43000	0
15603246	Female	27	57000	0
15804002	Male	19	76000	1
15728773	Male	27	58000	1
15598044	Female	27	84000	0
15694829	Female	32	150000	1
15600575	Male	25	33000	1
15727311	Female	35	65000	0
15570769	Female	26	80000	1
15606274	Female	26	52000	0
15746139	Male	20	86000	1
15704987	Male	32	18000	0
15628972	Male	18	82000	0
15697686	Male	29	80000	0
15733883	Male	47	25000	1

Figure A: CLASSIFICATION

Discrete or Non numerical target

Features/predictors

Temperature	Pressure	Relative Humidity	Wind Direction	Wind Speed
10.69261758	986.882019	54.19337313	195.7150879	3.278597116
13.59184184	987.8729248	48.0648859	189.2951202	2.909167767
17.70494885	988.1119385	39.11965597	192.9273834	2.973036289
20.95430404	987.8500366	30.66273218	202.0752869	2.965289593
22.9278274	987.2833862	26.06723423	210.6589203	2.798230886
24.04233986	986.2907104	23.46918024	221.1188507	2.627005816
24.41475295	985.2338867	22.25082295	233.7911987	2.448749781
23.93361956	984.8914795	22.35178837	244.3504333	2.454271793
22.68800023	984.8461304	23.7538641	253.0864716	2.418341875
20.56425726	984.8380737	27.07867944	264.5071106	2.318677425
17.76400389	985.4262085	33.54900114	280.7827454	2.343950987
11.25680746	988.9386597	53.74139903	68.15406036	1.650191426
14.37810685	989.6819458	40.70884681	72.62069702	1.553469896
18.45114201	990.2960205	30.85038484	71.70604706	1.005017161
22.54895853	989.9562988	22.81738811	44.66042709	0.264133632
24.23155922	988.796875	19.74790765	318.3214111	0.329656571

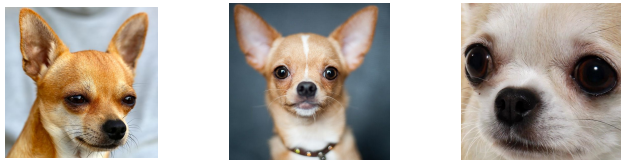
Figure B: REGRESSION

Continuous numerical target



Train a ML model

This is chihuahua



This is muffin



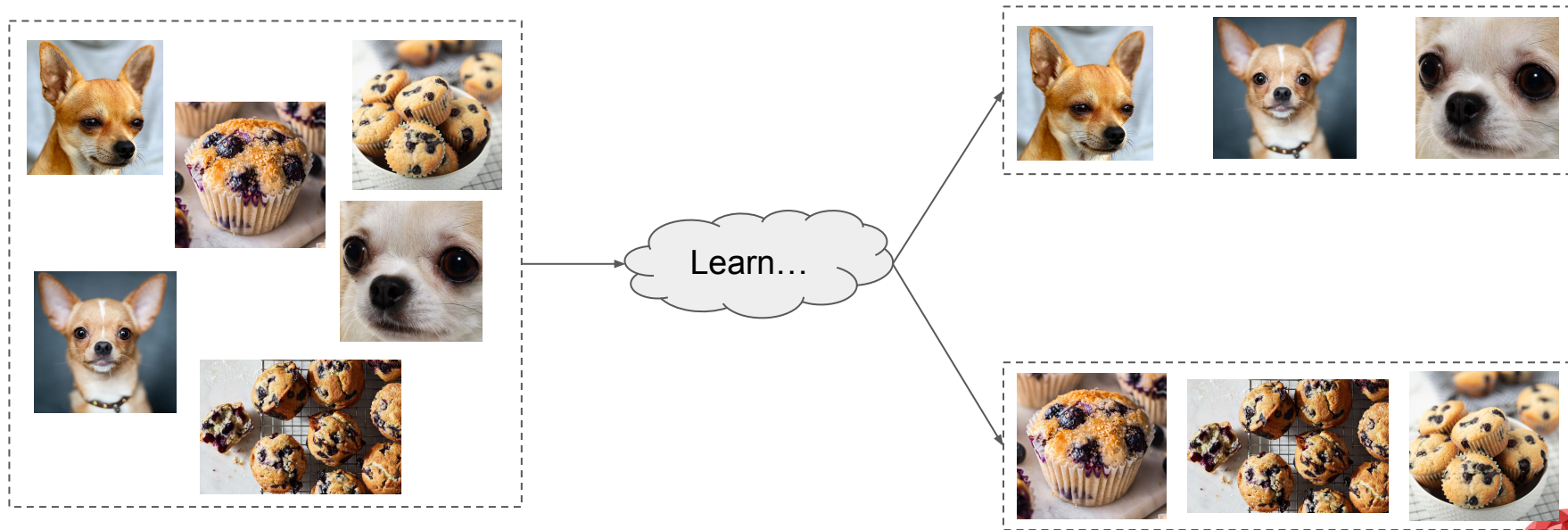
Learn...

Predict this



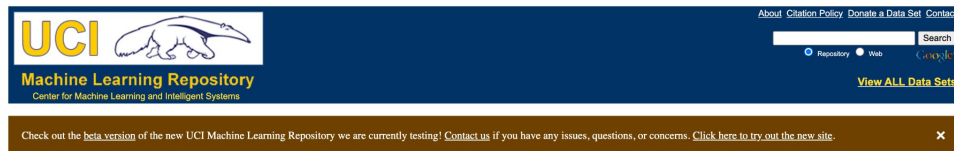
*classification

Train a ML model



*clustering















UCI Machine Learning Repository



Welcome to the UC Irvine Machine Learning Repository!

We currently maintain 622 data sets as a service to the machine learning community. You may [view all data sets](#) through our searchable interface. For a general overview of the Repository, please visit our [About](#) page. For information about citing data sets in publications, please read our [citation policy](#). If you wish to donate a data set, please consult our [donation policy](#). For any other questions, feel free to [contact the Repository librarians](#).

Supported By:  In Collaboration With: 

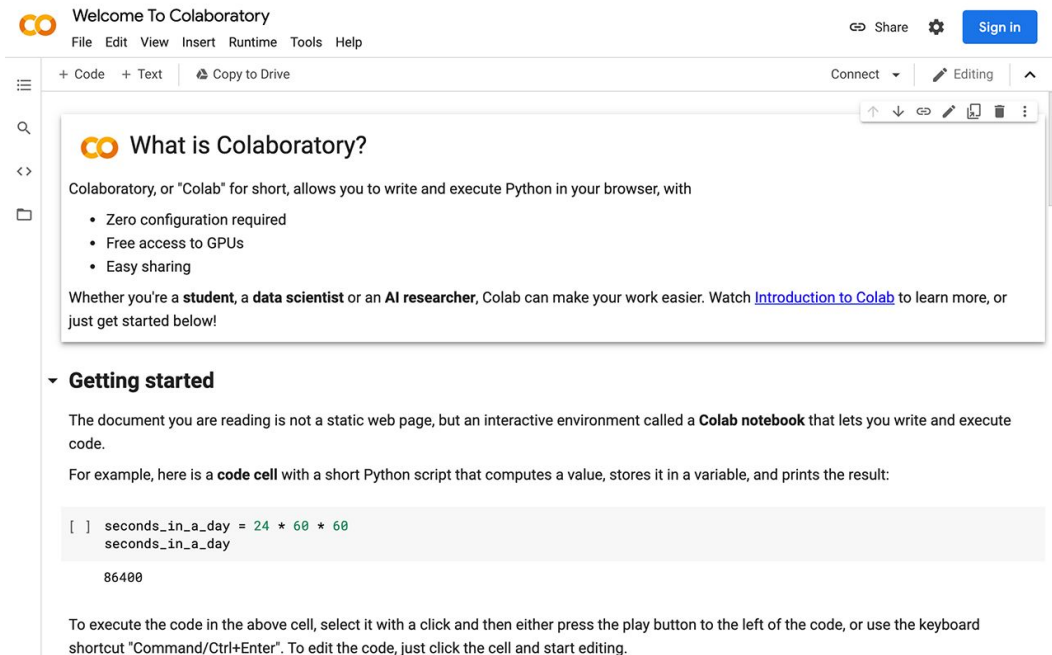
Latest News:	Newest Data Sets:	Most Popular Data Sets (hits since 2007):
<p>09-24-2018: Welcome to the new Repository admins Dheeru Dua and Elita Karna-Tamaskidze!</p> <p>04-04-2013: Welcome to the new Repository admins Kevin Bache and Moshe Lichman!</p> <p>03-01-2010: Note from donor regarding Netflix data</p> <p>10-16-2009: Two new data sets have been added.</p> <p>09-14-2009: Several data sets have been added.</p> <p>03-24-2008: New data sets have been added!</p> <p>06-25-2007: Two new data sets have been added: UJI Pen Characters, MAGIC Gamma Telescope</p>	<p>06-05-2021:  Average Localization Error (ALE) in sensor node localization process in WSNs</p> <p>05-25-2021:  Smers from culcode</p> <p>05-18-2021:  TamilSentMix</p> <p>05-02-2021:  Accelerometer</p> <p>04-21-2021:  Synchronous Machine Data Set</p> <p>04-21-2021:  Synchronous Machine Data Set</p> <p>04-20-2021:  Pedal Me Bicycle Deliveries</p>	<p>5207130:  Iris</p> <p>2715360:  Adult</p> <p>2205247:  Dry Bean Dataset</p> <p>2122743:  Wine</p> <p>2121420:  Heart Disease</p> <p>2106094:  Wine Quality</p> <p>2006448:  Rank Marketing</p>

<https://archive.ics.uci.edu/ml/index.php>

- The UCI Machine Learning Repository is a collection of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms.
- It is used by students, educators, and researchers all over the world as a primary source of machine learning data sets.
- Alternatives: kaggle, data.world



Google Colab



Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

Share Sign in

+ Code + Text Copy to Drive

Connect Editing

What is Colaboratory?

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch [Introduction to Colab](#) to learn more, or just get started below!

Getting started

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
[ ] seconds_in_a_day = 24 * 60 * 60
    seconds_in_a_day

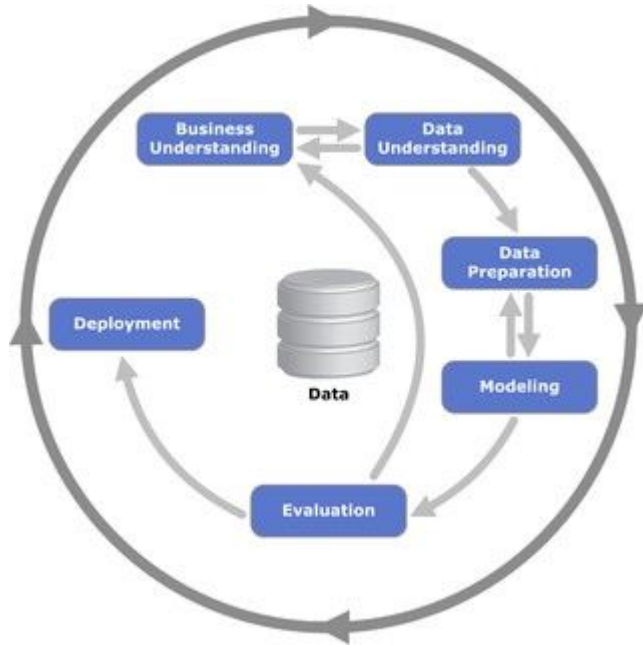
86400
```

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter". To edit the code, just click the cell and start editing.

- Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.
- Colab enables you to
 - Write and execute code in Python
 - Document your code that supports mathematical equations
 - Create/Upload/Share notebooks
 - Import/Save notebooks from/to Google Drive
 - Import/Publish notebooks from GitHub
 - Import external datasets e.g. from Kaggle
 - Integrate PyTorch, TensorFlow, Keras, OpenCV
 - Free Cloud service with free GPU



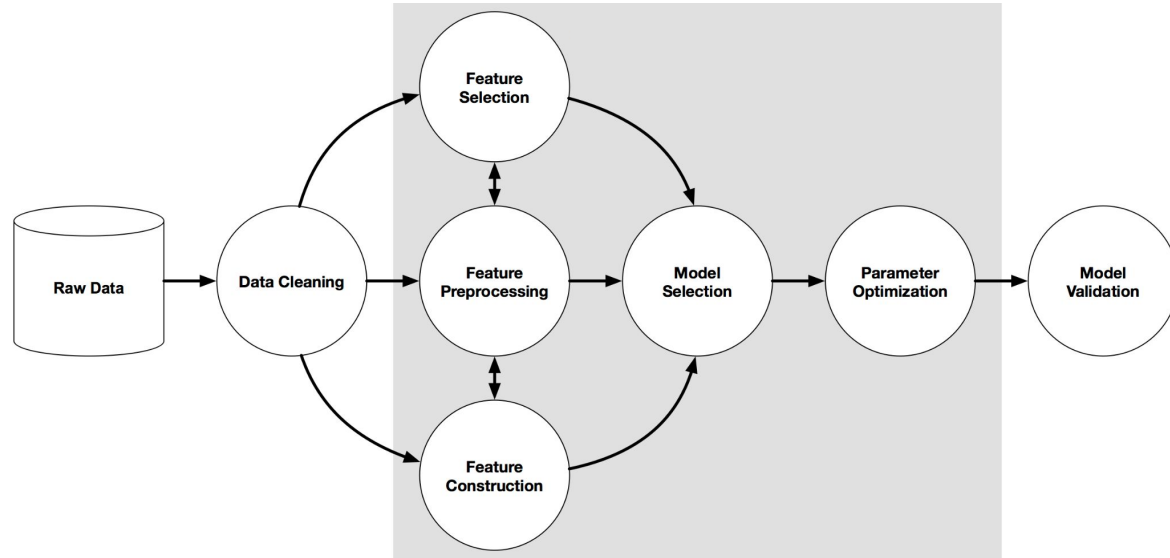
Problem solving framework



The **CR**oss Industry **S**tandard **P**rocess for **D**ata **M**ining (**CRISP-DM**) is a process model that serves as the base for a data science process. It has six sequential phases:

1. Business understanding – What does the business need?
2. Data understanding – What data do we have / need? Is it clean?
3. Data preparation – How do we organize the data for modeling?
4. Modeling – What modeling techniques should we apply?
5. Evaluation – Which model best meets the business objectives?
6. Deployment – How do stakeholders access the results?

Machine learning workflow



Data understanding

```
✓ [3] df.head()
```

	Class	age	menopause	tumor-size	inv-nodes	node-caps	deg-malig	breast	breast-quad	irradiat
0	no-recurrence-events	30-39	premeno	30-34	0-2	no	3	left	left_low	no
1	no-recurrence-events	40-49	premeno	20-24	0-2	no	2	right	right_up	no
2	no-recurrence-events	40-49	premeno	20-24	0-2	no	2	left	left_low	no
3	no-recurrence-events	60-69	ge40	15-19	0-2	no	2	right	left_up	no
4	no-recurrence-events	40-49	premeno	0-4	0-2	no	2	right	right_low	no

- Data understanding can be started by looking at the sample data
- How they are organized, existence of missing values, wrong data types, value range and distribution etc.

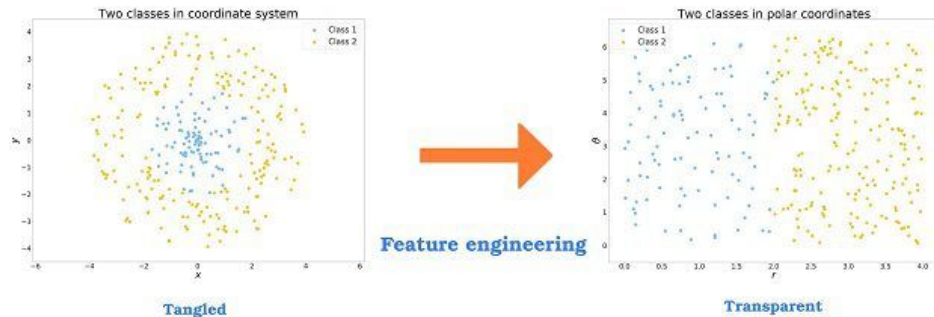


Feature selection

- Feature selection is the process of isolating the most consistent, non-redundant, and relevant features to use in model construction.
- Example of feature selection techniques:
 - Removes all low-variance features
 - Select by univariate analysis score
 - Feature ranking with recursive feature elimination



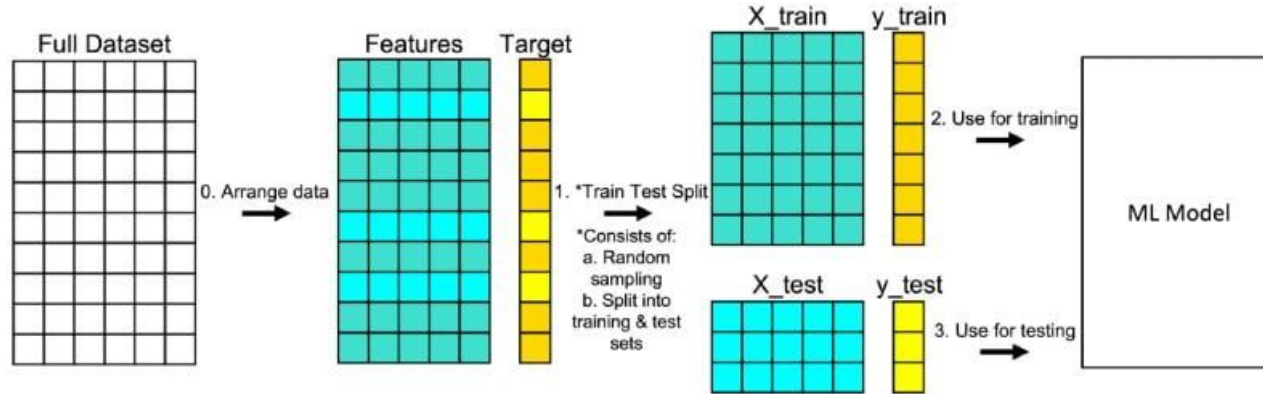
Feature engineering



*“Coming up with features is difficult, time-consuming, requires expert knowledge. ‘**Applied machine learning is basically feature engineering.**’ — Prof. Andrew Ng.*

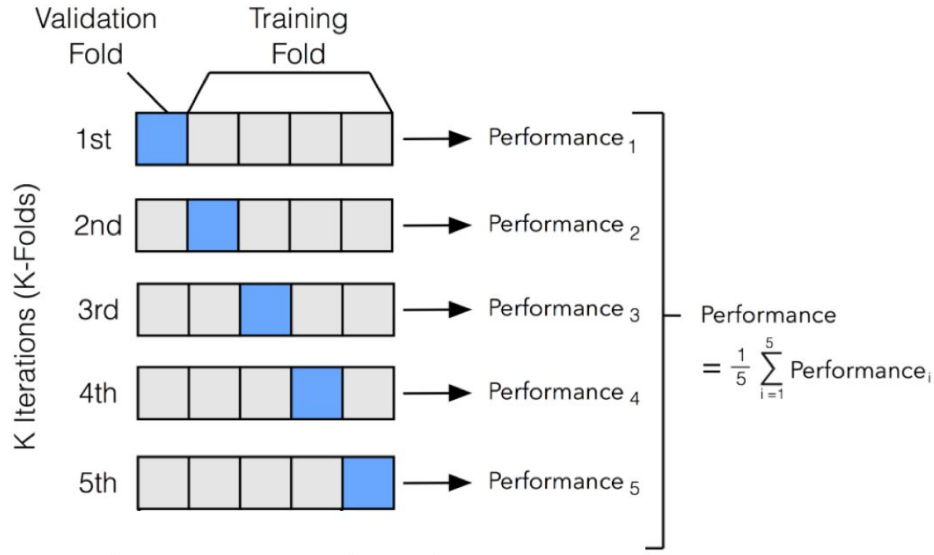
- Feature engineering is the ‘art’ of formulating useful features from existing data following the target to be learned and the machine learning model used.
- It involves transforming data to forms that better relate to the underlying target to be learned.
- Feature engineering include: imputing missing values, discretization, feature scaling, categorical encoding, creating new features, etc.

Train test split



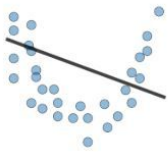


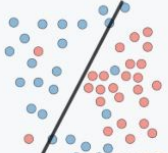
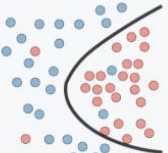
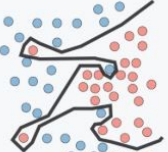

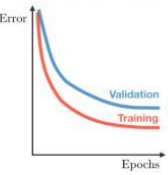
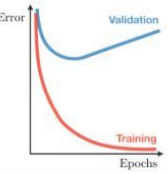
- Train test split is a model validation procedure that reveals how your model performs on new data.
- Every transformation or rules should be based on training dataset only not on whole dataset

Cross Validation



1. Divide the sample data into k parts
2. Use the k-1 parts for training, and 1 for testing
3. Repeat the procedure k times, rotating the test set
4. Determine an expected performance metric based on the results across the iterations

Underfitting and Overfitting

	Underfitting	Just right	Overfitting
Symptoms	<ul style="list-style-type: none"> • High training error • Training error close to test error • High bias 	<ul style="list-style-type: none"> • Training error slightly lower than test error 	<ul style="list-style-type: none"> • Very low training error • Training error much lower than test error • High variance
Regression illustration			
Classification illustration			
Deep learning illustration			
Possible remedies	<ul style="list-style-type: none"> • Complexify model • Add more features • Train longer 		<ul style="list-style-type: none"> • Perform regularization • Get more data

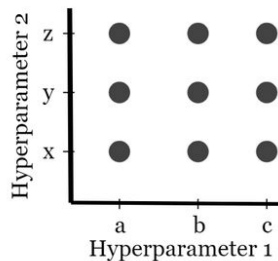
Hyperparams tuning

- Hyperparameter tuning is choosing a set of optimal hyperparameters for a learning algorithm.
- Example of tuning methods:
 - Grid search, process that searches exhaustively through a manually specified subset of the hyperparameter space of the targeted algorithm.

Grid Search

Pseudocode

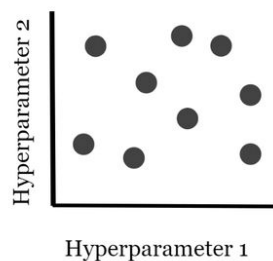
```
Hyperparameter_One = [a, b, c]  
Hyperparameter_Two = [x, y, z]
```



Random Search

Pseudocode

```
Hyperparameter_One = random.num(range)  
Hyperparameter_Two = random.num(range)
```



Classifications

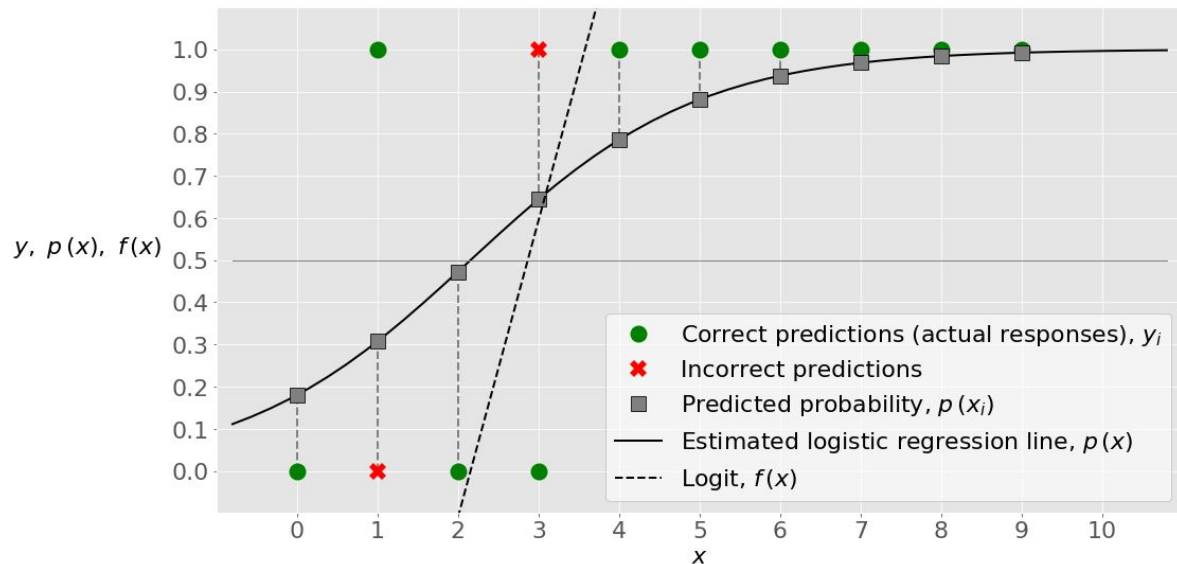


Classification models

1. Logistic Regression
2. Support Vector Machine
3. K Nearest Neighbors
4. Decision Tree
5. Ensemble: Random Forest

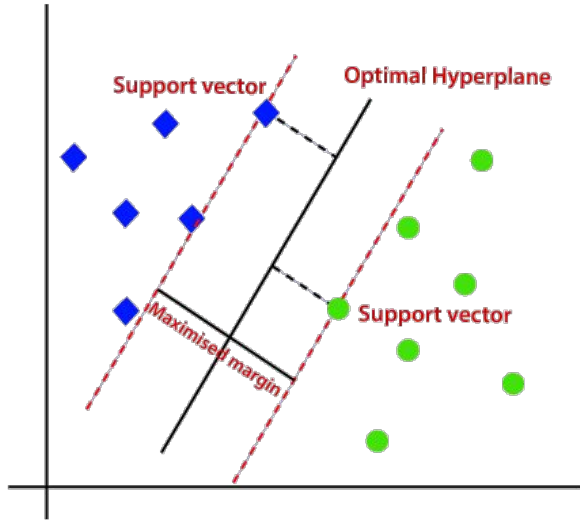


Logistic Regression

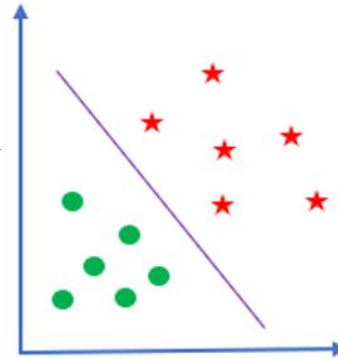


- Logistic regression, by default, is limited to two-class (binary) classification problems.
- Some extensions like one-vs-rest can allow logistic regression to be used for multi-class classification problems
- Although they require that the classification problem first be transformed into multiple binary classification problems.

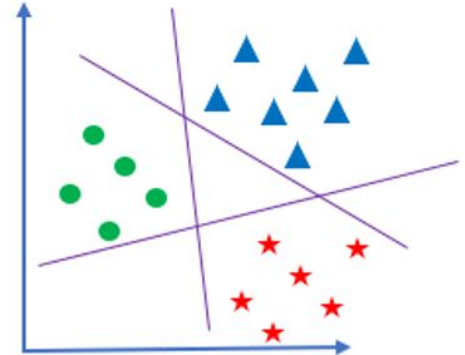
Support Vector



Binary classification

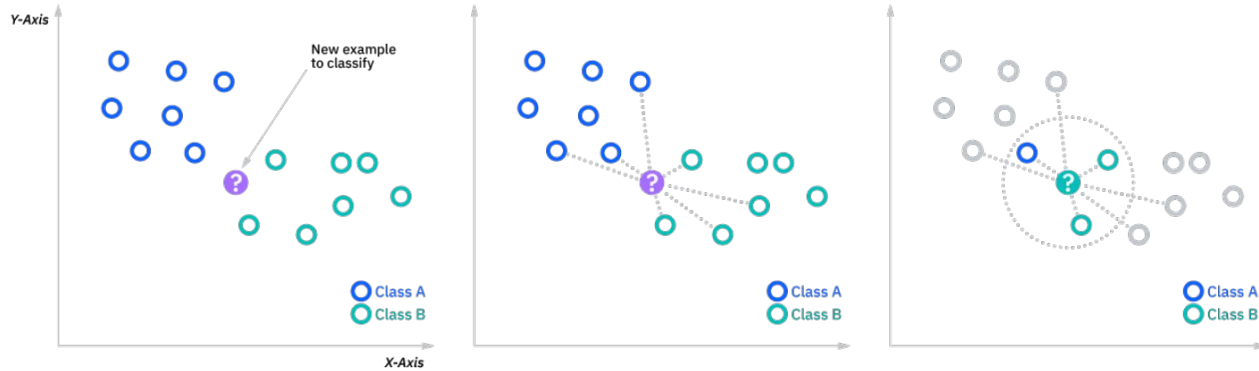


Multi-class classification



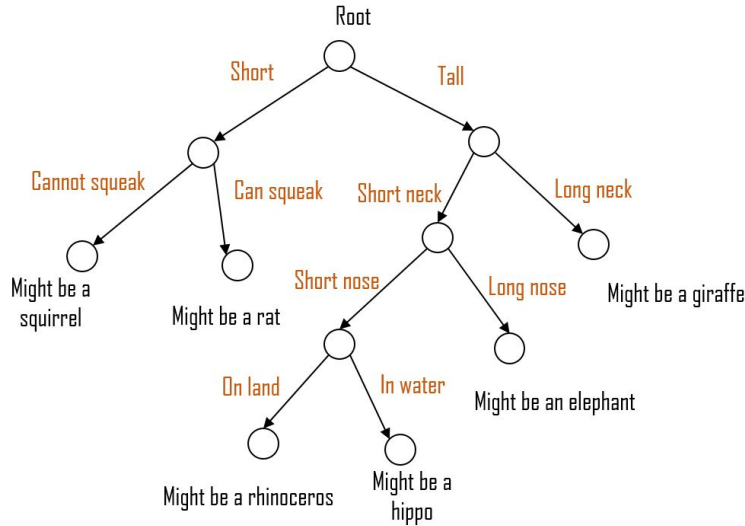
- SVM works by mapping data to a high-dimensional feature space so that data points can be categorized, even when the data are not otherwise linearly separable.
- A separator between the categories is found, then the data are transformed in such a way that the separator could be drawn as a hyperplane.

K-nearest neighbors



The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.

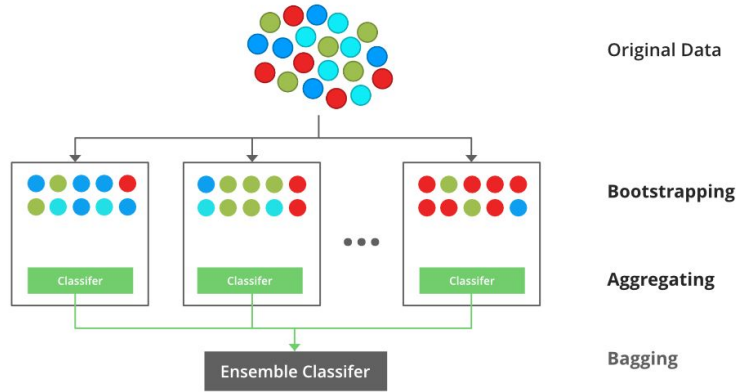
Decision Tree



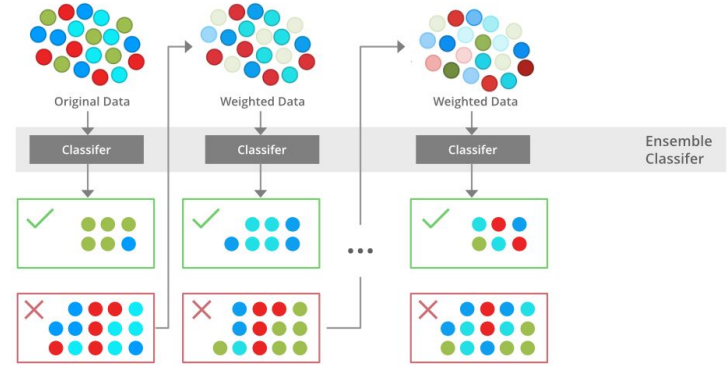
- Decision trees classify the examples by sorting them down the tree from the root to some leaf/terminal node, with the leaf/terminal node providing the classification of the example.
- Each node in the tree acts as a test case for some attribute, and each edge descending from the node corresponds to the possible answers to the test case.
- This process is recursive in nature and is repeated for every subtree rooted at the new node.

Ensemble

Bagging



Boosting



- Ensemble methods is a machine learning technique that combines several base models in order to produce one optimal predictive model
- Bagging decrease variance (solves overfitting). Boosting decrease bias
- Example of ensemble: Random forest, XGBoost, AdaBoost

Classification metrics

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

- Precision** — What percent of your predictions were correct?
 Precision is the ability of a classifier not to label an instance positive that is actually negative. For each class, it is defined as the ratio of true positives to the sum of a true positive and false positive. $\text{Precision} = TP / (TP + FP)$
- Recall** — What percent of the positive cases did you catch?
 Recall is the ability of a classifier to find all positive instances. For each class it is defined as the ratio of true positives to the sum of true positives and false negatives. $\text{Recall} = TP / (TP + FN)$
- F1 score** — What percent of positive predictions were correct?
 The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. F1 scores are lower than accuracy measures as they embed precision and recall into their computation. As a rule of thumb, the weighted average of F1 should be used to compare classifier models, not global accuracy.

$$\text{F1 Score} = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$$
- Support**
 Support is the number of actual occurrences of the class in the specified dataset. Imbalanced support in the training data may indicate structural weaknesses in the reported scores of the classifier and could indicate the need for stratified sampling or rebalancing. Support doesn't change between models but instead diagnoses the evaluation process.



Regression



Regression models

1. Linear Regression
 - a. Ordinary least square
 - b. Lasso
 - c. Ridge
 - d. Elastic-Net
2. Also works for regression
 - a. Support vector
 - b. KNN
 - c. Decision tree
 - d. Random forest



Ordinary least square

- Minimizing the sum of the squared residuals

$$\sum_{i=1}^M (y_i - \hat{y}_i)^2 = \sum_{i=1}^M \left(y_i - \sum_{j=0}^p w_j \times x_{ij} \right)^2$$



Lasso vs Ridge

Lasso

$$L_{lasso}(\hat{\beta}) = \sum_{i=1}^n (y_i - x_i' \hat{\beta})^2 + \lambda \sum_{j=1}^m |\hat{\beta}_j|.$$

Ridge

$$L_{hridge}(\hat{\beta}) = \sum_{i=1}^n (y_i - x_i' \hat{\beta})^2 + \lambda \sum_{j=1}^m w_j \hat{\beta}_j^2.$$

- The cost function for both ridge and lasso regression are similar. However, ridge regression takes the square of the coefficients and lasso takes the magnitude.
- Lasso regression can be used for automatic feature selection, as the geometry of its constrained region allows coefficient values to inert to zero.
- An alpha value of zero in either ridge or lasso model will have results similar to the regression model.
- The larger the alpha value, the more aggressive the penalization.



Elastic-Net

- Elastic-Net is a linear regression model trained with both L1 and L2-norm regularization of the coefficients. This combination allows for learning a sparse model where few of the weights are non-zero like Lasso, while still maintaining the regularization properties of Ridge.
- Elastic-net is useful when there are multiple features that are correlated with one another. Lasso is likely to pick one of these at random, while elastic-net is likely to pick both.
- Objective function to minimize below:

$$L_{enet}(\hat{\beta}) = \frac{\sum_{i=1}^n (y_i - x_i^T \hat{\beta})^2}{2n} + \lambda \left(\frac{1-\alpha}{2} \sum_{j=1}^m \hat{\beta}_j^2 + \alpha \sum_{j=1}^m |\hat{\beta}_j| \right)$$



Regression metrics

Mean squared error

$$\text{MSE} = \frac{1}{n} \sum_{t=1}^n e_t^2$$

Root mean squared error

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^n e_t^2}$$

Mean absolute error

$$\text{MAE} = \frac{1}{n} \sum_{t=1}^n |e_t|$$

Mean absolute percentage error

$$\text{MAPE} = \frac{100\%}{n} \sum_{t=1}^n \left| \frac{e_t}{y_t} \right|$$



Thank You

