Google Developer Student Clubs

Head Start Machine Learning

Session 3:

Machine Learning with scikit learn

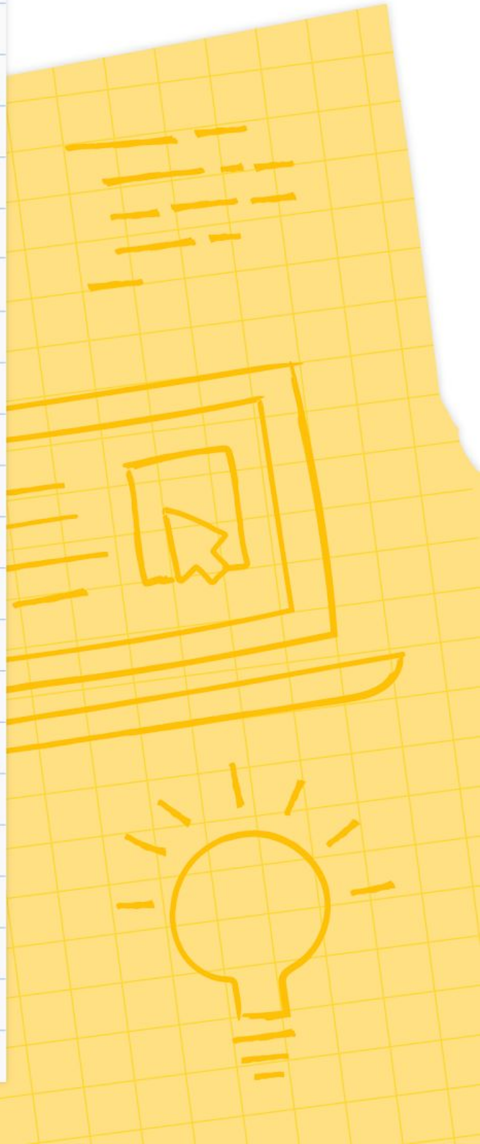
Amos Tan Li Sheng
Year 2 CS(AI)

Hong Jia Herng
Year 2 CS(IS)

Google Developer Student Clubs

Attendance

> Select Tab “Head Start Machine Learning”
> Mark your attendance by inserting “1”
> Do it within this 2 hours



What are we going to learn?

- Solve a ML problem using Scikit-Learn
- Data exploration
- Data preparation
- Hyperparameter search using k-fold cross-validation

Drop your question in the chat box if you don’t understand anything!

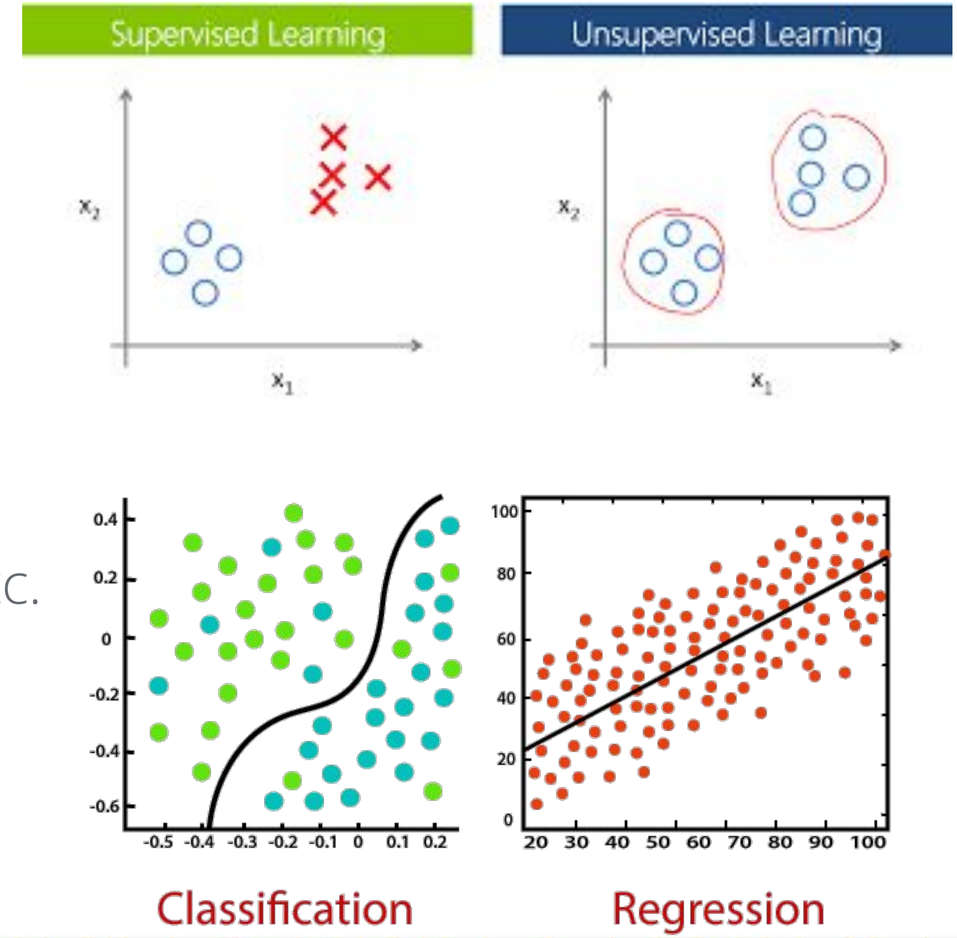
6 common steps when doing a ML project

1. Frame the problem
2. Get the data
3. Explore the data
4. Prepare the data for ML algorithms
5. Select a model and train it
6. Improve your model

1) Frame the problem

Some questions to ask yourself ...

- Is it a **supervised** or **unsupervised** learning task?
- Is it a **classification** or **regression** task?
- What **performance measure** to use for this problem?
  - Regression : MSE, RMSE, MAE, etc.
  - Classification : Accuracy, Precision, Recall, F1 score, etc.



2) Get the data

`After getting the data, you should do this!`

- Split dataset into **training set** and **test set**
  - Put **test set** aside and never look at it (typically below 40% of the whole dataset)
  - We want to work solely with our **training set** to prevent us from *overfitting* onto the entire dataset
  - **Test set** enables you to obtain an estimate for the generalization error that your model will have when deploying in production

3) Explore the data

Understand your data

- Study each attribute and its characteristics

```
train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
PassengerId    891 non-null int64
Survived       891 non-null int64
Pclass         891 non-null int64
Name           891 non-null object
Sex            891 non-null object
Age            714 non-null float64
SibSp          891 non-null int64
Parch          891 non-null int64
Ticket         891 non-null object
Fare           891 non-null float64
Cabin          204 non-null object
Embarked       889 non-null object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.6+ KB
```

Which features contain null values?

- Age, Cabin, Embarked

Which features are categorical?

- Survived (*Target*), Sex, Ticket, Cabin, Embarked, Pclass (*Ordinal*)

Which features are numerical?

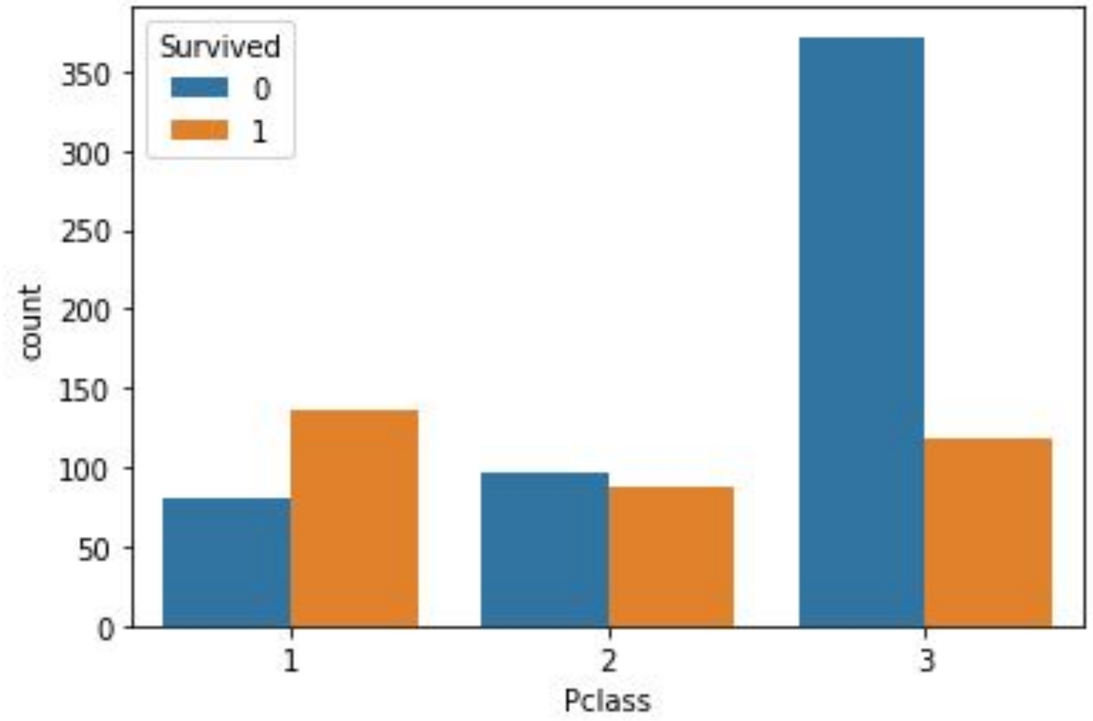
- Age, Fare, SibSp (*Discrete*), Parch (*Discrete*)

And more…

3) Explore the data

Understand your data

- Visualize the data



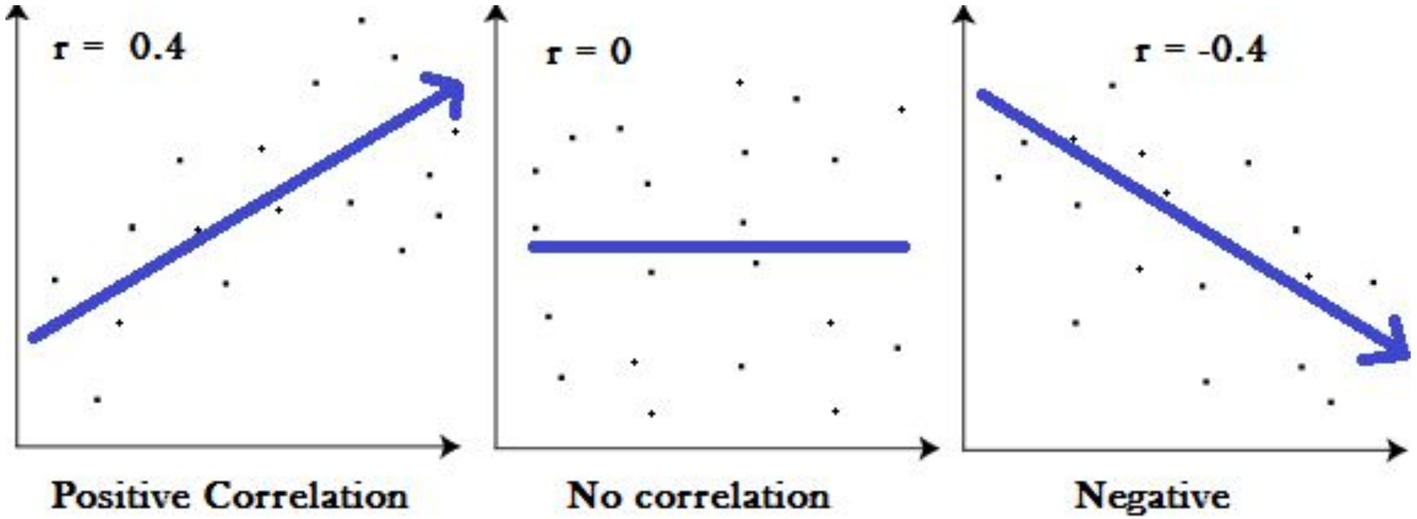
- Pclass: socio-economic status
  1 → upper
  2 → middle
  3 → lower

- From this visualization, we can infer that upper class people has higher surviving rate

3) Explore the data

Understand your data

- Determine correlation of each numerical feature to target



Pearson's r

- ranges from -1 to 1
- near to 1, strong positive correlation
- near to -1, strong negative correlation
- near to 0, no linear correlation

cons:

- only measure "linear" correlation (if $x\uparrow$, then $y\uparrow$ or $\downarrow$), may miss nonlinear correlation (when $x$close to 0, then $y\uparrow$ or $\downarrow$)

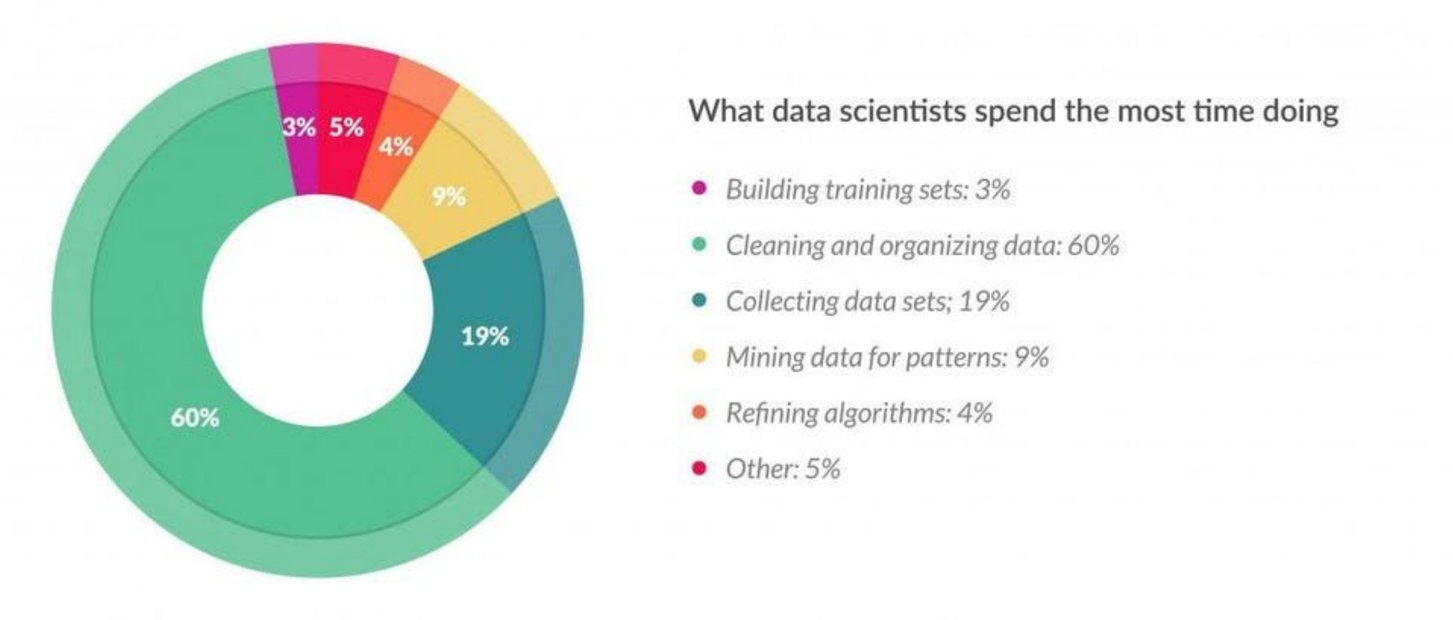
3) Explore the data

Understand your data

```
train_data.corr()["Survived"].sort_values(ascending=False)
```

```
Survived      1.000000
Fare          0.257307
Parch         0.081629
PassengerId  -0.005007
SibSp        -0.035322
Age          -0.077221
Pclass       -0.338481
Name: Survived, dtype: float64
```

Data scientists spend 60% of their time on cleaning and organizing data.



https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/?sh=5a9a94ed6f63

4) Data preparation

`Before feeding into ML algorithm`

Data cleaning

- Might have incomplete data / anomalies that results from human error
- Fix or remove outliers
- Fill in missing values (zero, mean, median, or drop the entire row/column)

	A
1	**university_name**
2	uni_malaya
3	um
4	UM
5	University of Malaya
6	Universiti Malaya
7	Unversiti of Malaya

4) Data preparation

Before feeding into ML algorithm

Feature selection

- Drop the features that provide no useful information for the task (could be based on correlation coefficient)
- Remove repeated feature, if any

(repeated columns are said to be linearly dependent with each other)

	Size in feet^2	Size in meter^2	Price
0	3743.122298	347.736061	112293.668944
1	3359.033153	312.054180	100770.994586
2	3101.394988	288.119594	93041.849651
3	3761.806730	349.471845	112854.201905
4	2310.515177	214.646860	69315.455309
5	2028.492572	188.446960	60854.777145
6	2939.060670	273.038736	88171.820096
7	3086.782949	286.762136	92603.488470
8	2687.292169	249.649442	80618.765066
9	2122.426592	197.173430	63672.797774

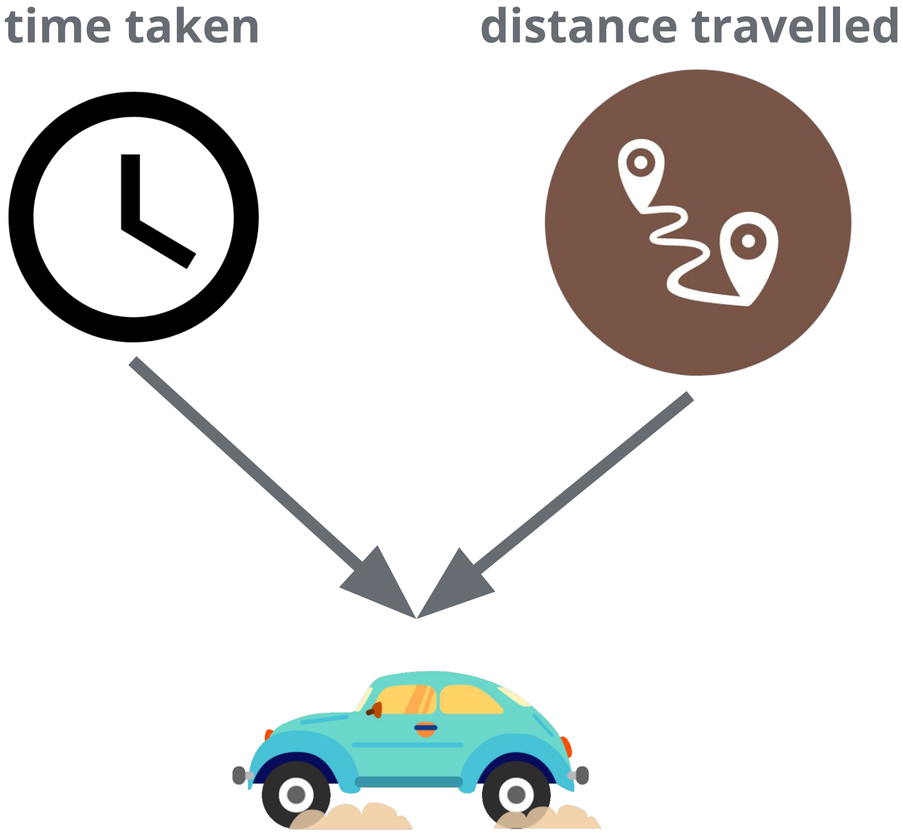
square meter = 0.09290304 * square feet

4) Data preparation

`Before feeding into ML algorithm`

Feature engineering

- Some attributes might not make much sense on their own
- Combining attributes could give more information to the model



4) Data preparation

Before feeding into ML algorithm

Handling categorical attributes

- Most ML algorithms prefer to work with numbers
- Need to convert categorical into numerical
- 2 common methods:
  - Ordinal encoding
  - One-hot encoding

Ordinal encoding

Breakfast
Every day
Never
Rarely
Most days
Never

Breakfast
3
0
1
2
0

One-hot encoding

Color
Red
Red
Yellow
Green
Yellow

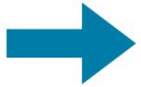
Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1
0	1	0

4) Data preparation

Before feeding into ML algorithm

Ordinal encoding

Breakfast
Every day
Never
Rarely
Most days
Never



Breakfast
3
0
1
2
0

- ML algorithms assume that 2 nearby values are more similar than 2 distant values
- Not suitable to use for the categories that have equal existence (red, blue, green)

4) Data preparation

Before feeding into ML algorithm

One-hot encoding

Color
Red
Red
Yellow
Green
Yellow

Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1
0	1	0

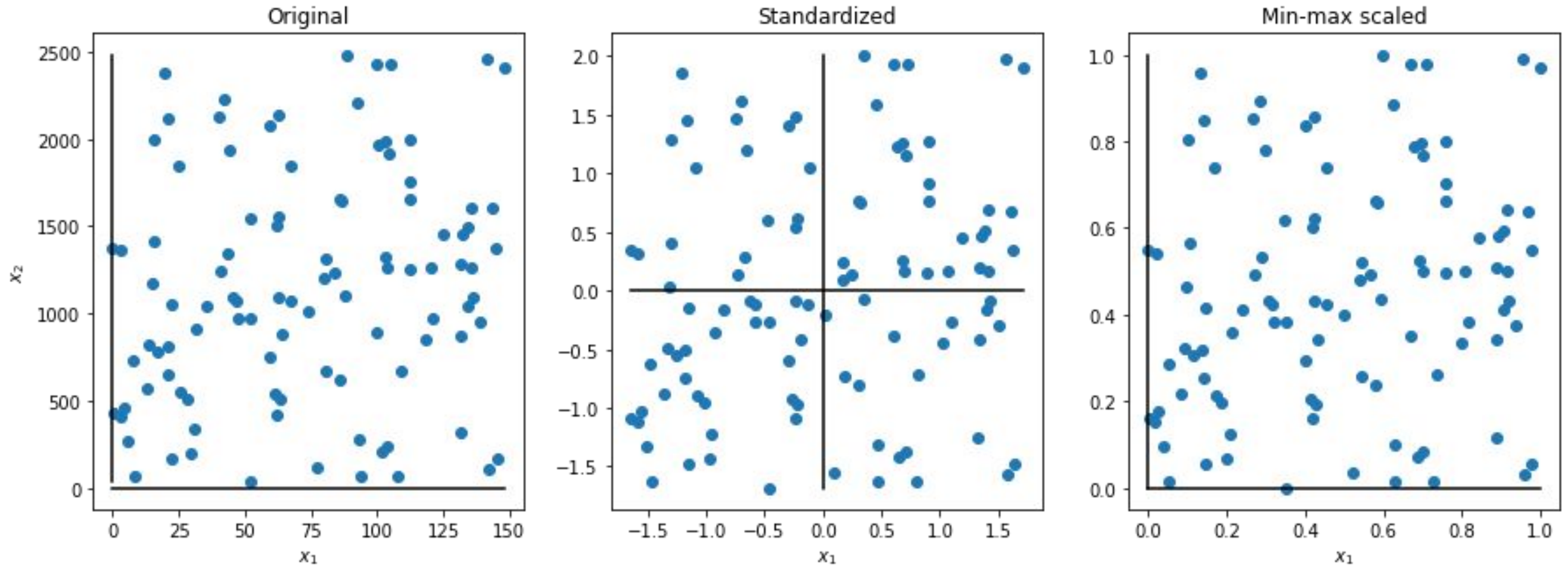
- Fix the issue that ordinal encoding produces
- But if there's a large number of categories, 1-hot encoding results in a large number of input features
  - may slow down training

4) Data preparation

Before feeding into ML algorithm

Feature scaling

- speed up training time



Google Developer Student Clubs

4) Data preparation

Before feeding into ML algorithm

Min-max scaling / Normalization

- Scale input features into the range of [0, 1]

$$x_{norm} = \frac{x - min(x)}{max(x) - min(x)}$$

4) Data preparation

Before feeding into ML algorithm

Standardization

- Scale input features into standard normally distributed data
- No bounding range

$$x_{norm} = \frac{x - mean(x)}{std(x)}$$

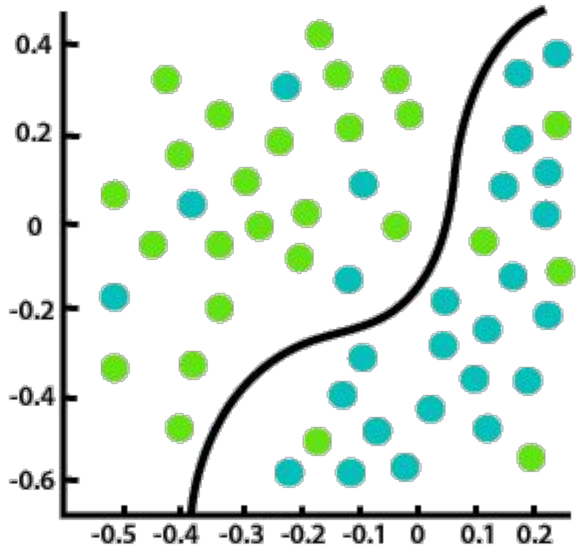
5) Select a model and train it

`The only part that uses ML algorithms`

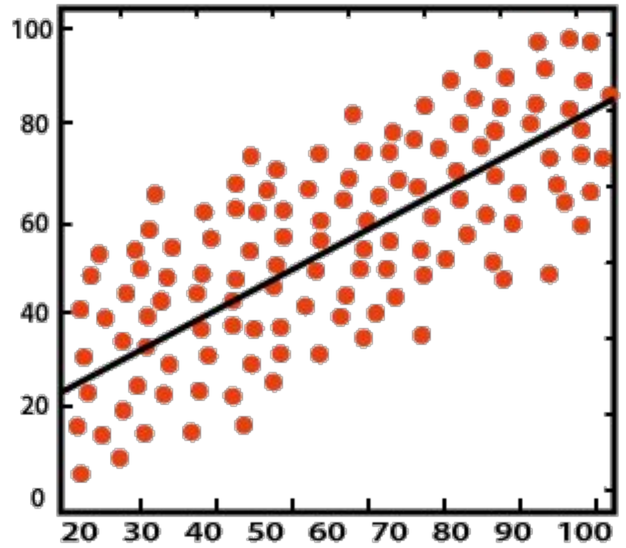
- After the data is well prepared, the rest you need to do is just select a suitable model and train it on the training set

- For a supervised learning problem, it is usually either a **classification** or **regression** problem

- https://scikit-learn.org/stable/



Classification

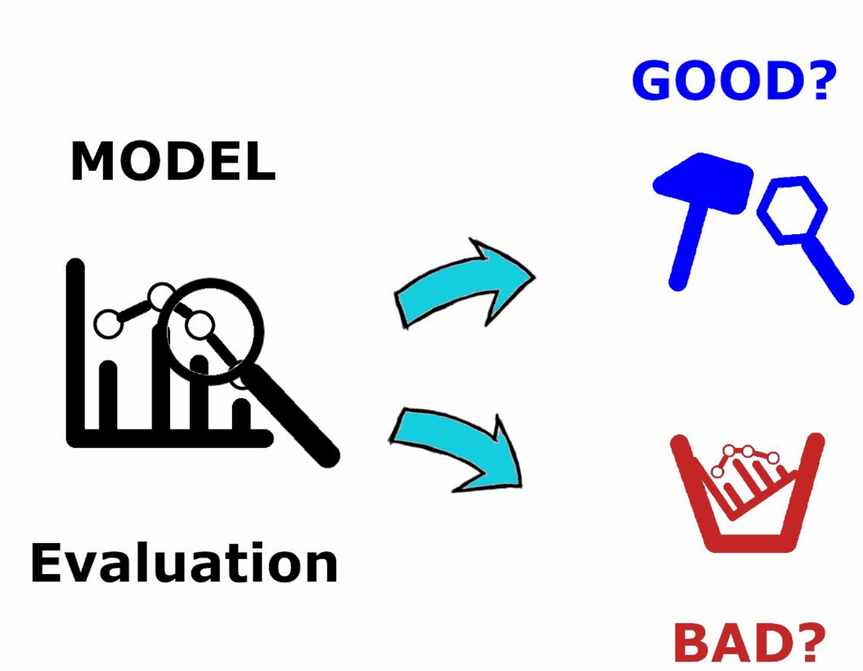


Regression

5) Select a model and train it

`Evaluate your model`

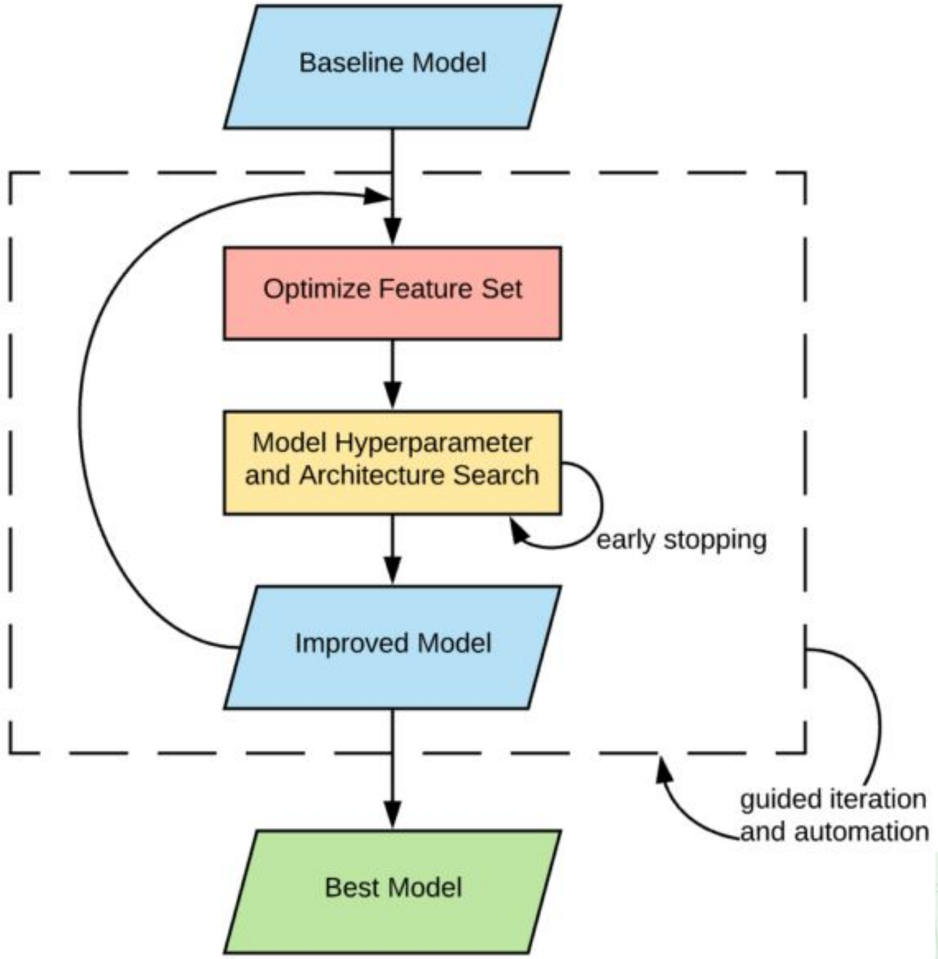
- Use the performance measure that you've chosen at **step 1)** to evaluate it
- It's a metric to tell how well your model is performing on predicting the target
- You can also use it to compare the performance of different models



6) Improving your model

What to do to improve the performance?

- Try different models
- Revisit previous data preparation steps
- Hyperparameter tuning

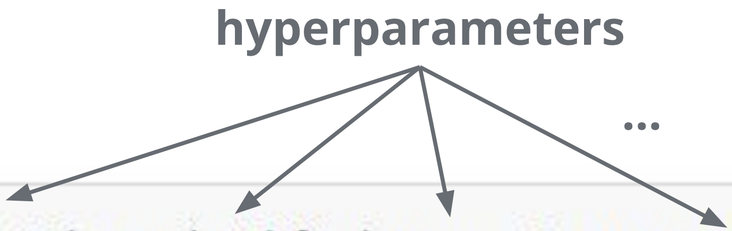


6) Improving your model

`What to do to improve the performance?`

Hyperparameter tuning

- Search for the best set of hyperparameters for the model
- 2 common ways: **Grid search** or **Randomized search**




```
class sklearn.svm.SVC(*, C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0,
shrinking=True, probability=False, tol=0.001, cache_size=200, class_weight=None,
verbose=False, max_iter=- 1, decision_function_shape='ovr', break_ties=False,
random_state=None)
```

6) Improving your model

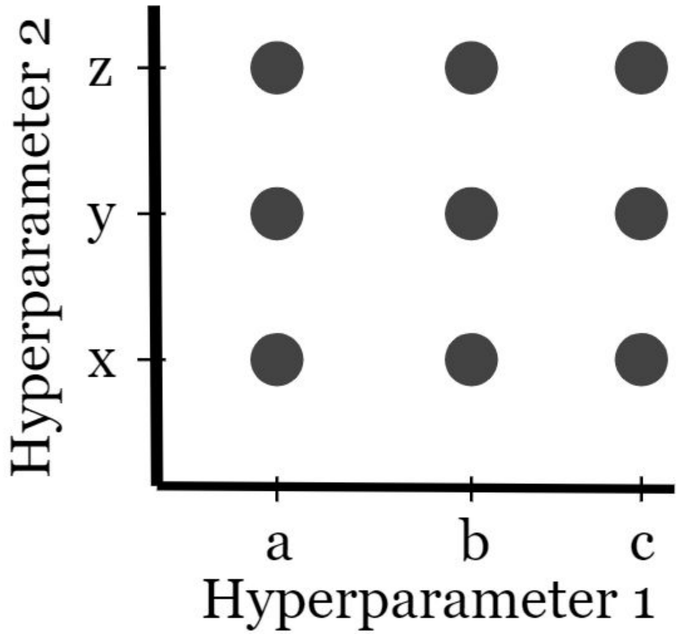
What to do to improve the performance?

Hyperparameter tuning

Grid Search

Pseudocode

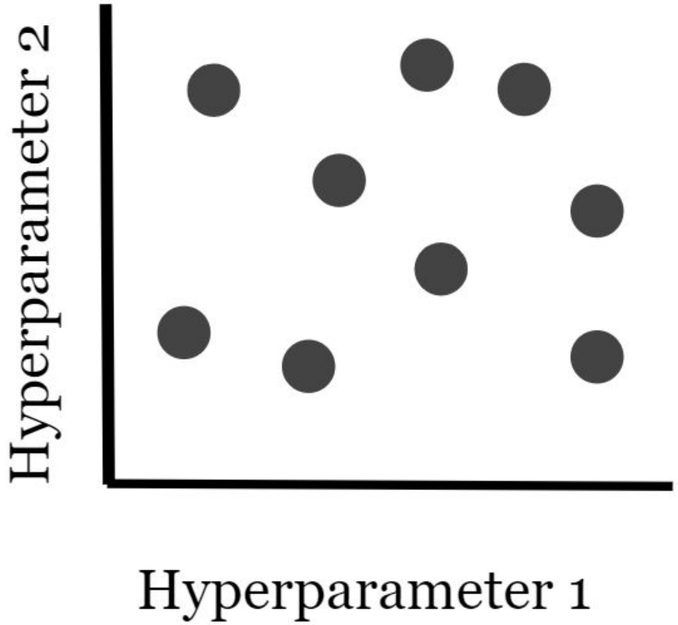
```
Hyperparameter_One = [a, b, c]
Hyperparameter_Two = [x, y, z]
```



Random Search

Pseudocode

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

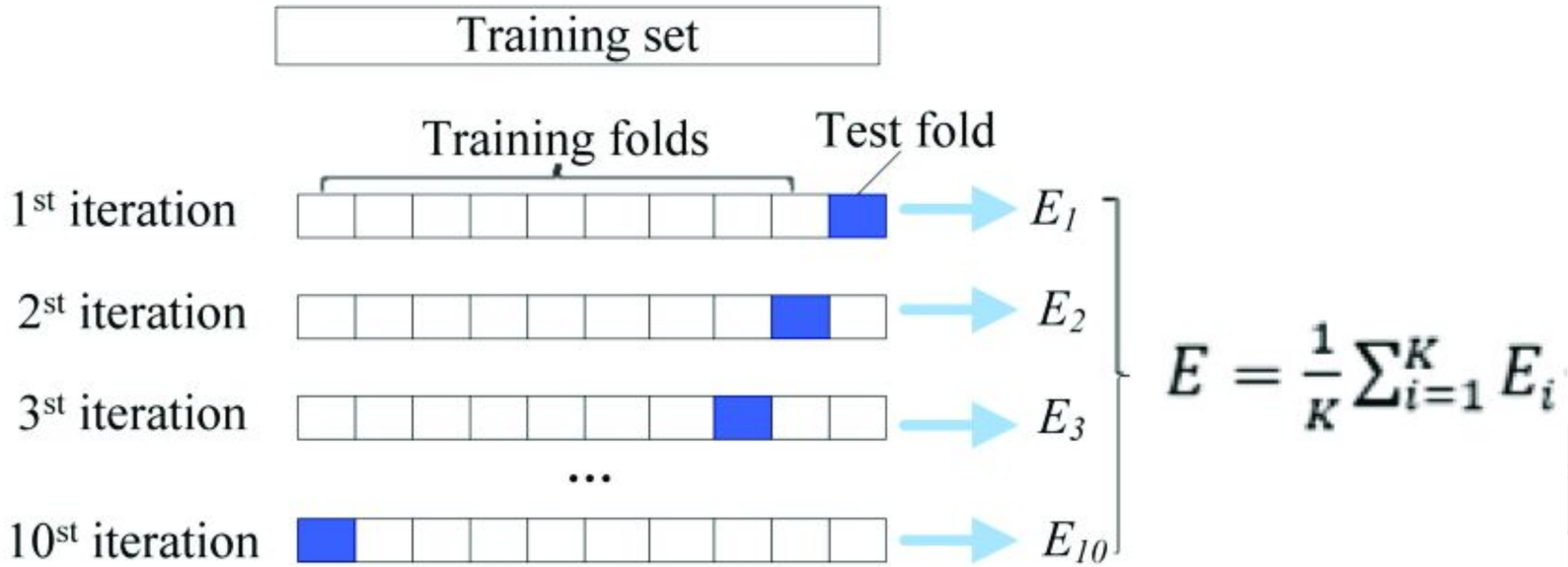


6) Improving your model

What to do to improve the performance?

Search using k-fold cross-validation

- Provide the number of folds, k
- Randomly split the training set into k folds
- Model trains on the k-1 folds and evaluate on the remaining 1 fold for k iterations
- Performance measure are averaged over k values



Q&A

Google Developer Student Clubs

Google Developer Student Clubs

Let’s do the hands-on session now!

Google Developer Student Clubs

Leave us your feedback

https://forms.gle/FGfQTmJxHUtNGzfw6

Group photo!

Google Developer Student Clubs