ALGORITHMS

MACHINE LEARNING

AGENDA

- representation of data models in a digital computer (mapping model components to data structure and functions)
- training algorithm (optimisation)
- experiment design and implementation, such as the strategy of configuring model hyper-parameters.

REGRESSION

The equation of regression line is represented as:

$$h(x_i) = \beta_0 + \beta_1 x_i$$

Now consider:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i = h(x_i) + \varepsilon_i \Rightarrow \varepsilon_i = y_i - h(x_i)$$

- Y = *wx* + *b*
- A *regression algorithm* is a machine learning algorithm that produces a *regression model*.
- Regression Model
- A *regression model* type of *model* that outputs continuous (typically, floating-point).

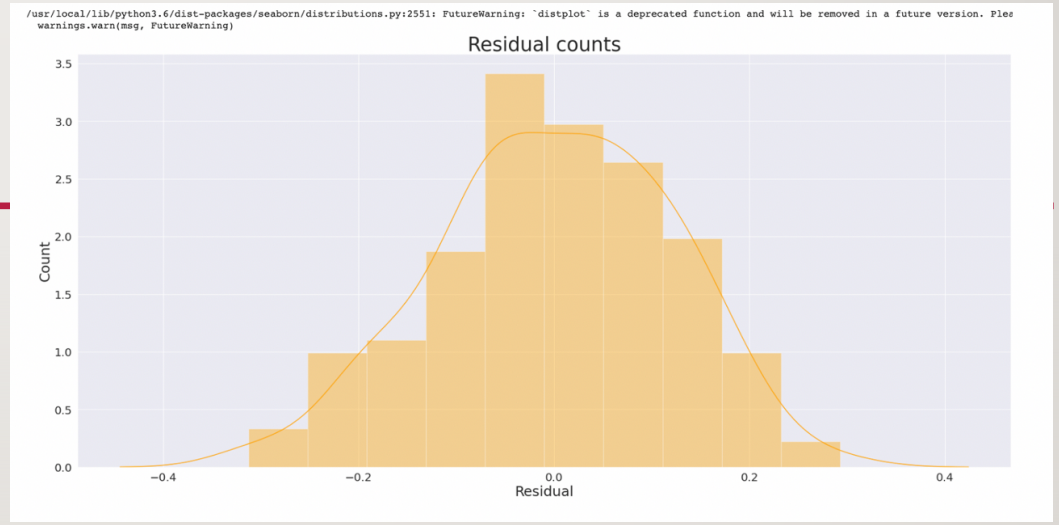
REGRESSION EVALUATION METRICS

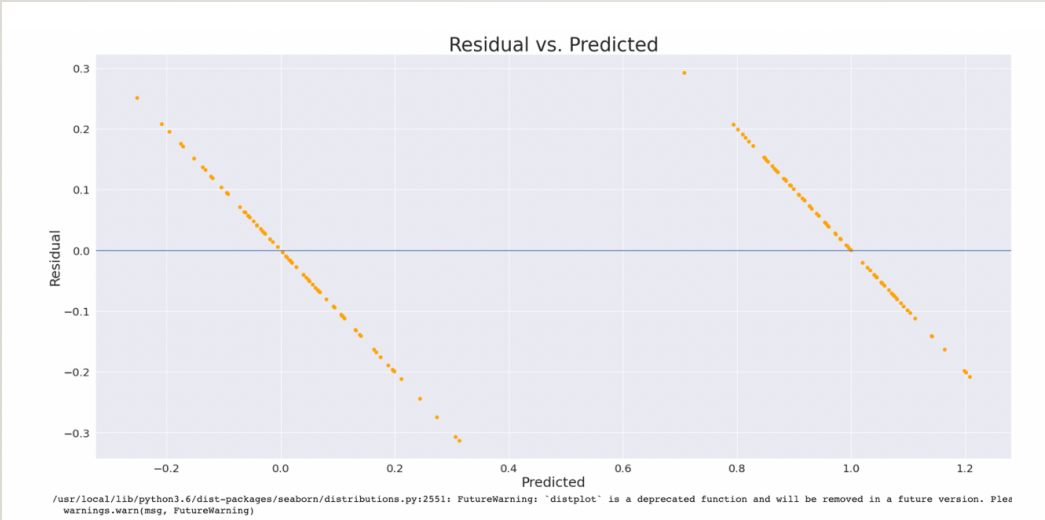
- Three common evaluation metrics for regresson problems:
- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
  All basic variations on the difference between what you predicted and the true values.
- Comparing these metrics:
- **MAE** is the easiest to understand, because it's the average error.
  MSE more popular than MAE, because MSE "punishes" larger errors, tends to be useful in the real world.
  RMSE is even more popular than MSE, because RMSE is interpretable in the "y" units (target units) .
- All of these are loss functions, because we want to minimize them.

Let's take which attributes we would like to focus on our feature:

Price ~ Rooms + Landsize + Bedroom + Bathroom + Car

```
Coefficients:
 [938.23786125]
Mean squared error: 0.01
Coefficient of determination: 0.94
```





REGRESSION METRICS

```
print("Linear Regression R^2 Score: ", lr.score(X_train, Y_train))
print("Linear Regression Test R^2 Score: ", lr.score(X_test, Y_test))
y_pred = lr.predict(X_test)
print("Mean Squared Error: ", mean_squared_error(y_pred, Y_test))
print("Mean Absolute Error: ", mean_absolute_error(y_pred, Y_test))
print("Cross Validation Score: ", cross_val_score(lr, X_test, Y_test, cv=5))
```

- Since hyperparameters are not optimized by the machine learning algorithm itself, it is often optimized using *cross-validation* and techniques like *grid search*, *random search*, *Bayesian optimization*, *evolutionary optimization*, and others.