

Machine learning

In simple terms, machine learning is a modern algorithm that can improve its performance by itself using more data.

Machine learning Model:

Machine learning Model is like a function that accept data as input and produce an output

Selecting a Model depends on goals:

Model can be simple and complex both depend upon use case a AI robot Model is more complex than A simple spam detection or classification model

Parameter of model:

As like function have parameter and model have also parameter , initially parameter are taken random and

Auto corrected during training to improve model performance where the model learns from the data by minimizing the error between its predictions and the actual outcomes.

Model Training and testing

Statsmodels : Think of **statsmodels** as your toolkit for doing **statistical tests**, **linear models**, **time series analysis**, and getting a deep look into **model diagnostics**.

Where **Scikit-learn** is more about prediction, **statsmodels** is more about **explanation** and **statistical inference**.

Formula API:

It features for setting own formula in model instead of building a model

Basic Syntax : `model = smf.ols(formula='dependent_variable ~ independent_variables', data=df)`

Formula Example : `'y ~ x1 + x2 + x3'`

OLS:

LS stands for **Ordinary Least Squares**.

It's the most basic and commonly used method to fit a **linear regression model**.

In simple terms:

OLS tries to draw the best-fitting straight line through the data points by minimizing the **sum of squared errors** between predicted and actual values.

Jupyter Notebook of training and Testing : [Osl](#)

Objective function or Cost Function:

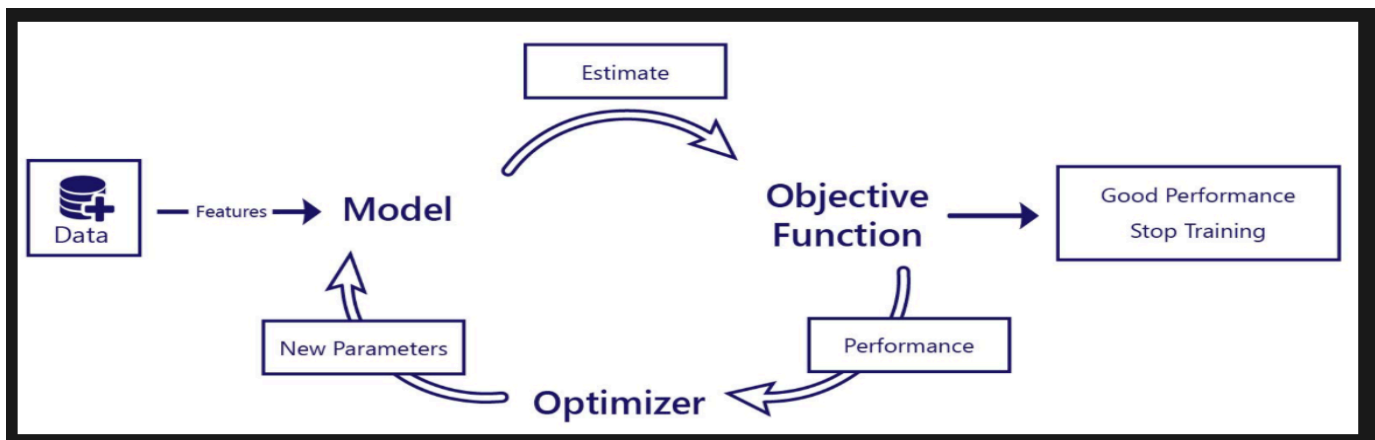
Objective function judge whether model is doing good job or bad job

The Data:

The provided during training

The Optimizer:

During the model training the model make a prediction and the objective function calculate how well model is performing and **optimizer is a code the change the model inside parameter so the model can perform better next time**



Saving the Model

Let look how to save the trained model after training for deployment

There are various ways to save the model

Build in - model.save() and model.load()

Method 1:

```
import statsmodels.formula.api as smf
model = smf.ols('salary ~ experience', data=df).fit()
```

```
# Save the model
```

```
model.save('ols_model.pkl')
```

```
# Load it later
```

```
import statsmodels.api as sm
loaded_model = sm.load('ols_model.pkl')
```

```
# Use the loaded model
```

```
print(loaded_model.summary())
```

Method 2:(using joblib)

```
import statsmodels.formula.api as smf
import joblib
```

```
# Train the model
```

```
model = smf.ols('salary ~ experience', data=df).fit()
```

```
# Save with joblib
```

```
joblib.dump(model, 'ols_model_joblib.pkl')
```

joblib

`Joblib` is a high-performance library used to:

- **Serialize** Python objects (save them to disk),
- **Deserialize** them (load them back from disk),
- And optionally **compress** them.

`JobLib.dump()`:

`joblib.dump()` function is used to serialize and save a Python object to a file. In the syntax `joblib.dump(model, model_filename)`, the function takes two arguments:

1. `model`: This is the object you want to save, which can be any Python object, typically a trained machine learning model.
2. `model_filename`: This is a string that specifies the filename (including the path, if necessary) where the model will be saved. The file will typically have a `.pkl` extension to indicate that it is a pickle file.

But builtYou can think of `joblib` as an **optimized version of `pickle`**

`.pkl`

`pkl` stands for `Pickle` – it's just the standard file extension for a Python object that has been serialized (i.e., turned into a byte stream and saved to a file).

