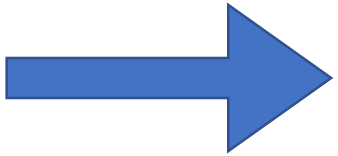
# End-to-end Project

# One Hot Encoding

| | COLOR | |
|---|---|---|
| | Red | |
| | Blue | |
| | Blue | |
| | Green | |
| | Blue | |
| | Red | |
| | Green | |



| Red | Green | Blue |
|---|---|---|
| 1 | 0 | 0 |
| 0 | 0 | 1 |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |

http://contrib.scikit-learn.org/categorical-encoding/index.html

# ML model fitting (Scikit-learn)

>_ Code

```
from sklearn.mlmodel import MachineLearningModelRegressor
mlmodel_reg = MachineLearningModelRegressor(random_state=42)
mlmodel_reg.fit(data_predictor_variables, data_labels)
```

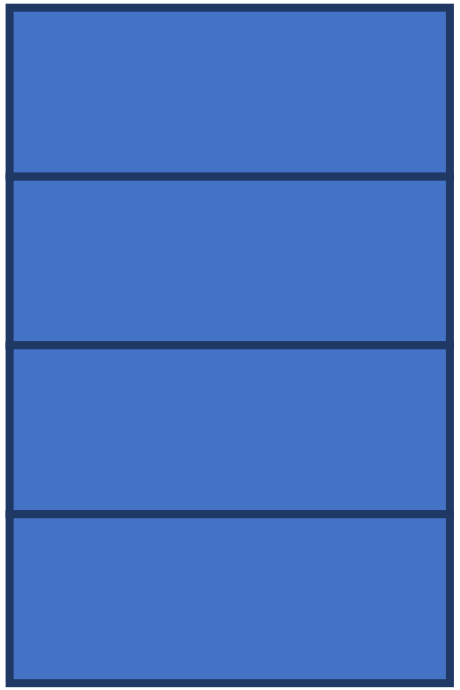
mlmodel = linear_model | tree | ensemble | svm | …
MachineLearningModelRegressor = LinearRegression | DecisionTreeRegressor | RandomForestRegressor | SVR |…

## Measuring Error

```
from sklearn.metrics import mean_squared_error
test_predictions = mlmodel_reg.predict(test_dataset)
mlmodel__mse = mean_squared_error(test_labels, test_predictions)
mlmodel_rmse = np.sqrt(mlmodel_mse)
```

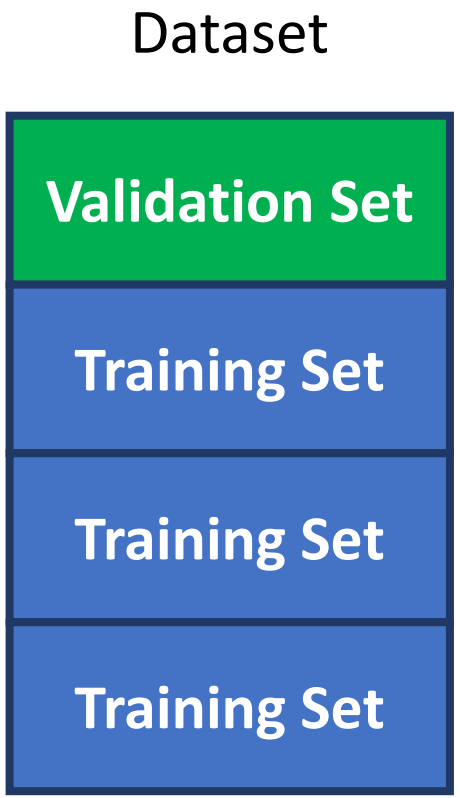
# $K$-fold cross-validation

Dataset



4-fold cross-validation

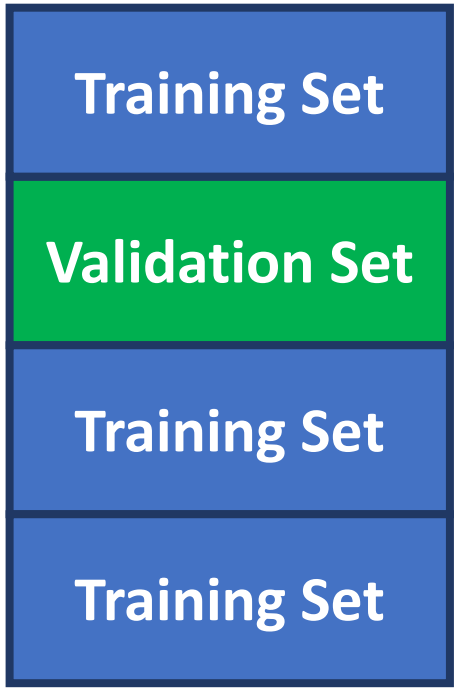
# $K$-fold cross-validation




4-fold cross-validation

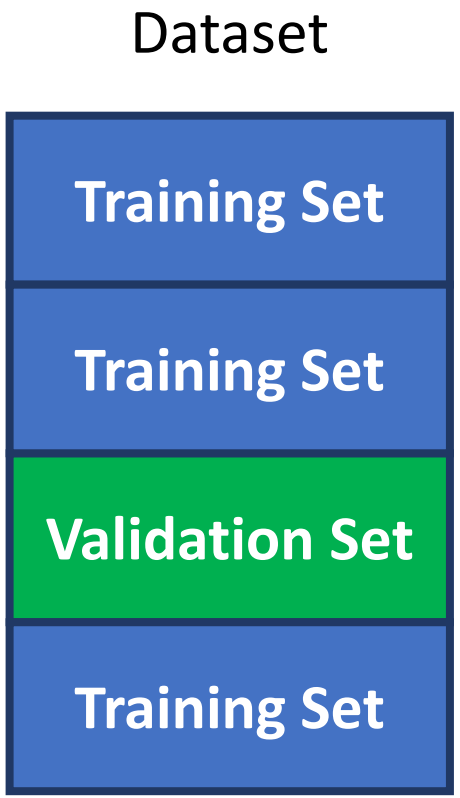
# $K$-fold cross-validation

Dataset




4-fold cross-validation

# $K$-fold cross-validation




4-fold cross-validation

# $K$-fold cross-validation

Dataset


4-fold cross-validation

# $K$-fold cross-validation

Dataset

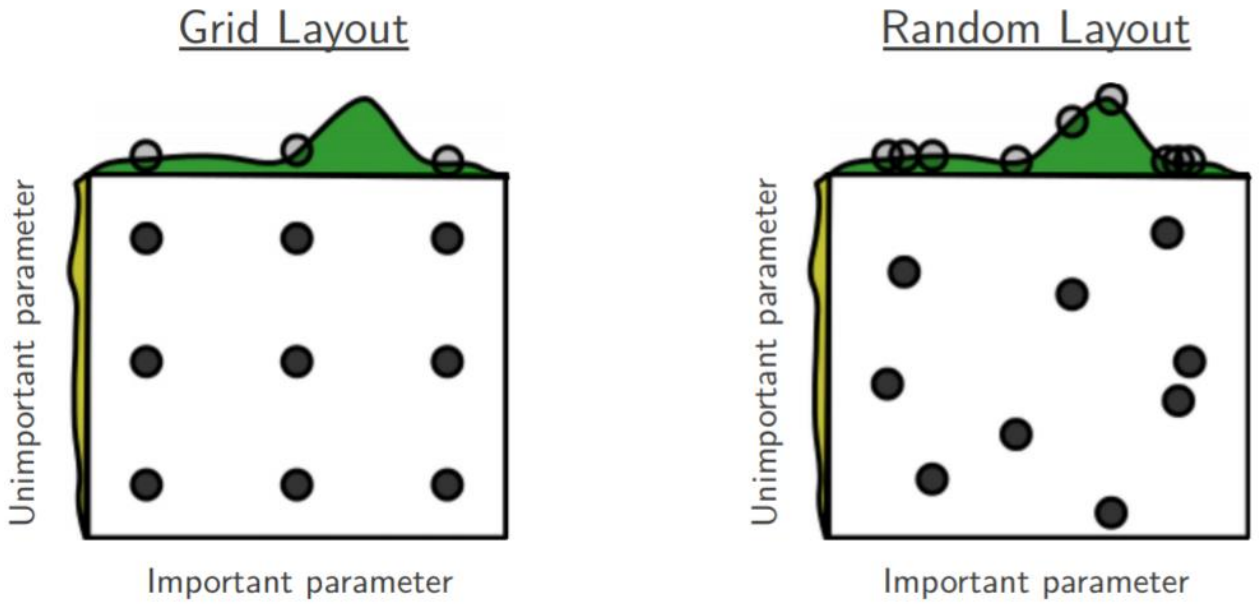

4-fold cross-validation

```
from sklearn.model_selection import cross_val_score
```

# Hyperparameter tuning

- Grid search
- Randomized search
- Bayesian optimization
- Informed search

```
def my_ML_model(data, a, b):
    c = find_model_best_param(data, a, b)
    return(c)
```




https://blog.usejournal.com/a-comparison-of-grid-search-and-randomized-search-using-scikit-learn-29823179bc85