# Parameter Tuning and Pipeline

**Tozammel Hossain**

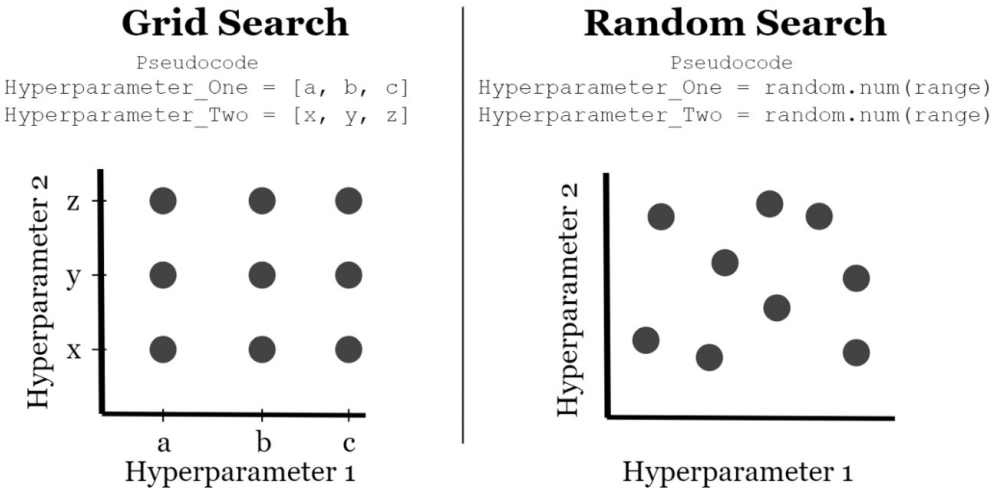
Data Science & Analytics
University of Missouri

# Hyper parameter

- **Most of the machine learning model has some hyper params which need to be select manually.**
  - Like for SVC params are
    - C - regularization parameter
    - gamma - Kernel coefficient
- **Choosing the best hyper parameter is always been an important part of a machine learning model**
- **Default params will not always provide the best output for a model.**

# Searching hyper Parameter

- **Sklearn provides two functions to search over hyper parameter space**
  - GridSearchCV
  - RandomSearchCV




https://medium.com/@jackstalfort/hyperparameter-tuning-using-grid-search-and-random-search-f8750a464b35

# GridSearchCV

- **Takes a dict of parameters**
  - Keys are the parameters
  - Each key has a list of values
- **Also takes a model which needs to be optimized**

```
searchable_param = {
    'C': [1e3, 1e5, ….],
'gamma': [0.0001, 0.1]}
```

- **GridSearchCV takes all combinations of C and gamma and choose the best combination based on accuracy score**

# GridSearchCV

```
model = GridSearchCV(
        mode, search_able_params, cv=10)
```

- Allow us to choose the size of cross validation fold while passing the cv value in GridSearchCV.

# RandomSearchCV

- **For random search we can provide parameters' distributions**
  - randomly picks some sample and choose the best combination

```
searchable_param = {
    'C': uniform(100, 10000),
    'gamma': uniform(0, 0.1)}
```

- **We can choose num of trials using n_iter**

# RandomSearchCV

- **RandomSearchCV will not always provide the best score.**
  - It will provide an idea about the value range that may give the best score

# Pipeline

- **Single machine learning task has multiple steps like**
  - Data preprocessing
  - Feature optimization
  - training model
  - ……
- **The idea of pipeline is to combine those step sequentially**
- **Passing the data to pipeline will apply the steps sequentially to the data (like assembly line)**

# Pipeline

- **E.g., We want to train a model which requires scaling the data and feature selection using PCA. So, the steps are:**
  - Scaling -> MinMaxScalar()
  - PCA -> PCA()
  - Model -> SVC()
- **Pipeline definition:**

```
pipe = Pipeline([(`scale`, MinMaxScalar()),
                 (`pca`, PCA()),
                 (`mode`, SVC())]
```

- **Pipeline takes a list of steps as a tuple as (tauple_name, function)**
- **fit funtion can be called on the pipeline**

# Parameter tuning in Pipeline

- **We can run GridSearchCV and RandomSearchCV both in a pipeline.**
- **To do this the params name in the parameter dict will be**
  - {step_name_of_pipeline}__{param_name}

**pipe = Pipeline([(`pca`, PCA()), (`model` SVC())])**

**If we want to do grid search on the n_features variable of PCA. Then params list will be.**

**params = {`pca__n_feature': [2,34, ……. ]}**

**Then we can pass pipeline to GridSearchCV along with the params.**

**GridSearchCV(pipe, params)**