```
# %%

# ============================================================================
# MODEL SELECTION BY GRID_SEARCH_CV
# ============================================================================


# %%

# importing the ibraries

from sklearn.datasets import make_regression
import numpy as np
from  sklearn.model_selection import KFold
from sklearn.model_selection import GridSearchCV
from sklearn import linear_model
from sklearn.model_selection import cross_val_score


#%%

# create dataset

X, y = make_regression(1000, 5, noise = 5.0)

#%%

# number of times to run

NUM_TRIALS = 30

#%%

# tune parameters taken

tuned_parameters = [{'solver' : ['svd', 'lsqr'],'fit_intercept': ['True'],'normalize': ['False']},
                    {'solver' : ['sag', 'cholesky'],'fit_intercept': ['False'],'normalize': ['true']}]


#%%

# score to be taken

score = 'r2'
non_nested_scores = np.zeros(NUM_TRIALS)
nested_scores = np.zeros(NUM_TRIALS)

#%%
```

```
#training and testing the different models with different datasets

# Loop for each trial

for i in range(NUM_TRIALS):
    x_train = X
    y_train=y

    inner_cv = KFold(n_splits=4, shuffle=True, random_state=i)
    outer_cv = KFold(n_splits=4, shuffle=True, random_state=i)
    model= GridSearchCV(estimator = linear_model.Ridge(), param_grid = tuned_parameters, scoring = score)
    model.fit(x_train, y_train)
    non_nested_scores[i] = model.best_score_

# Nested CV with parameter optimization

    model = GridSearchCV(estimator= linear_model.Ridge(), param_grid = tuned_parameters, cv=inner_cv, scoring= score)
    nested_score = cross_val_score(model, X=x_train, y=y_train, cv=outer_cv)
    nested_scores[i] = nested_score.mean()



#%%

# find model score

score_difference = non_nested_scores - nested_scores

print("Average difference of {:6f} with std. dev. of {:6f}.".format(score_difference.mean(), score_difference.std()))


# %%

# ============================================================================
# THE END
# ============================================================================
```