

**INSTRUCTIONS:** Please fill out this form with a PDF form editor to retain Acroform metadata and save the file with a name containing the substring lab3 in all lower case. Submit a copy to the correct Dropbox link by the due date listed on the course calendar. This worksheet is meant for students of Brooklyn College CISC 3440 to complete on their own. Contents are created and copyrighted.

Name:

Emplid:

Term:

1. (5 points) **Training models** Practice training models by setting parameters and hyperparameters for regression algorithms available in [1], then fill in the table below. An example is provided on the next page.

Parameters & Hyperparameters	Running time	Intercept	Coefficient (first 3)	Training Score
<code># use defaults</code> <code>SGDRegressor().fit(x_train, y_train)</code>				

## References

[1] Scikit Learn API Reference <https://scikit-learn.org/stable/modules/classes.html>

2. (5 points) Which of the combination of parameters and hyperparameters from Question 1 is the best model? Explain briefly in your own words.

Listing 1: Sample

```

1 import time                                # start Python file with imports
2 import numpy as np
3 from numpy import genfromtxt
4 from sklearn.impute import SimpleImputer
5 from sklearn.model_selection import train_test_split
6 from sklearn.linear_model import *
7
8 def print_row(item, t, i, s):
9     ii = (i.astype('float')[0] if type(i) == np.ndarray else i)
10    print("{}\t {:.4f}\t\t{>10.4f}\t\t{:.4f}".format(item, t, ii, s))
11
12 # Get Data
13 data = genfromtxt('data.csv', delimiter=',', skip_header=1)
14
15 # Clean Data
16 imp = SimpleImputer(missing_values=np.nan, strategy='mean')
17 data_imputed = imp.fit_transform(data)    # fill NaN values
18 y = data_imputed[:,8]                    # median household price, 9th column
19 x = np.delete(data_imputed, 8, 1)        # drop column before modeling
20
21 # Split into train and test samples
22 x_train, x_test, y_train, y_test = train_test_split(x, y)
23
24 print("Item \t\tTime \t\t Intercept \t\tScore")
25 print("-----")
26 s1 = time.perf_counter()
27 model1 = SGDRegressor(loss='huber').fit(x_train, y_train)
28 print_row("SGD Regression", (time.perf_counter()-s1), model1.intercept_, model1.score(x_train,y_train))
29
30 s2 = time.perf_counter()
31 model2 = LinearRegression().fit(x_train, y_train)
32 print_row("Linear Reg.", time.perf_counter()-s2, model2.intercept_, model2.score(x_train,y_train))

```

Listing 2: Sample Output

Item	Time	Intercept	Score
-----	-----	-----	-----
SGD Regression	0.2251	10.2277	0.0856
Linear Reg.	0.0037	-3568476.5740	0.6354