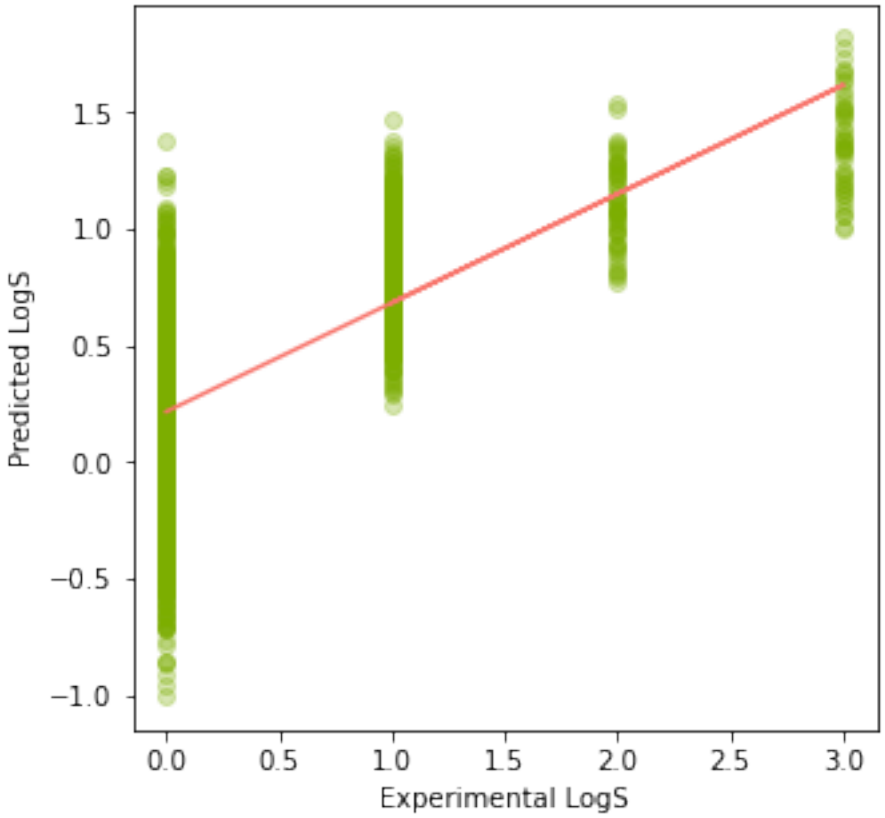
Task5

November 21, 2021

[3]:
```
from sklearn import datasets
import pandas as pd

# Generating Column Name manually.
column_names = [ 'buying','maint','doors','persons','lug_boot','safety','class']
data = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/
 ↪car/car.data', names = column_names)

# Convert Columns to INT Data Type
data['intBuying'] = 0
data['intMaint'] = 0
data['intLug_boot'] = 0
data['intSafety'] = 0
data['intClass'] = 0

data.loc[data['buying'] == 'vhigh', 'intBuying'] = 3
data.loc[data['buying'] == 'high', 'intBuying'] = 2
data.loc[data['buying'] == 'med', 'intBuying'] = 1
data.loc[data['buying'] == 'low', 'intBuying'] = 0

data.loc[data['maint'] == 'vhigh', 'intMaint'] = 3
data.loc[data['maint'] == 'high', 'intMaint'] = 2
data.loc[data['maint'] == 'med', 'intMaint'] = 1
data.loc[data['maint'] == 'low', 'intMaint'] = 0

data.loc[data['doors'] == '5more', 'doors'] = 5

data.loc[data['persons'] == 'more', 'persons'] = 6

data.loc[data['lug_boot'] == 'big', 'intLug_boot'] = 2
data.loc[data['lug_boot'] == 'med', 'intLug_boot'] = 1
data.loc[data['lug_boot'] == 'small', 'intLug_boot'] = 0

data.loc[data['safety'] == 'high', 'intSafety'] = 2
data.loc[data['safety'] == 'med', 'intSafety'] = 1
data.loc[data['safety'] == 'low', 'intSafety'] = 0
```

```
data.loc[data['class'] == 'vgood', 'intClass'] = 3
data.loc[data['class'] == 'good', 'intClass'] = 2
data.loc[data['class'] == 'acc', 'intClass'] = 1
data.loc[data['class'] == 'unacc', 'intClass'] = 0

# New DataFrame
selected_columns =␣
 ↪data[['intBuying','intMaint','doors','persons','intLug_boot','intSafety','intClass']]
dataNew = selected_columns.copy()

# Drop last class column and assign remaining columns to variable X
X = dataNew.drop(['intClass'], axis=1)
# Assign class column to variable y
y = dataNew.iloc[:,-1]
y = dataNew['intClass']

# Data Spliting
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42)

from sklearn.linear_model import LinearRegression
# Assign LinearRegression function to variable
linearR = LinearRegression()
# Model Training on Data
linearR.fit(X_train, y_train)

y_linearR_train_pred = linearR.predict(X_train)
y_linearR_test_pred = linearR.predict(X_test)

# Calculate Model Performance Metrics
from sklearn.metrics import mean_squared_error, r2_score
lr_train_mse = mean_squared_error(y_train, y_linearR_train_pred)
lr_train_r2 = r2_score(y_train, y_linearR_train_pred)
lr_test_mse = mean_squared_error(y_test, y_linearR_test_pred)
lr_test_r2 = r2_score(y_test, y_linearR_test_pred)

lr_results = pd.DataFrame(['Linear regression',lr_train_mse, lr_train_r2,␣
 ↪lr_test_mse, lr_test_r2]).transpose()
lr_results.columns = ['Method','Training MSE','Training R2','Test MSE','Test␣
 ↪R2']
lr_results

import matplotlib.pyplot as plt
import numpy as np
plt.figure(figsize=(5,5))
plt.scatter(x=y_train, y=y_linearR_train_pred, c="#7CAE00", alpha=0.3)
```

```
z = np.polyfit(y_train, y_linearR_train_pred, 1)
p = np.poly1d(z)
plt.plot(y_train,p(y_train),"#F8766D")
plt.ylabel('Predicted LogS')
plt.xlabel('Experimental LogS')

def export_to_pdf(Task5PDF):
    fn = export_to_html(Task5PDF)
    return convert_to_pdf(fn)
```



[ ]: