Regularization

March 15, 2018

1 Regularization

In this exercise, we will introduce regularization terms in our regression models to prevent overfitting. We will compare the effects of L2 (Ridge) to that of L1 (LASSO) regularization on prediction. For this exercise, we will be using the cars dataset. It is provided as cars.csv in the same folder as this notebook.

```
In [2]: #Import the necessary libraries
        %matplotlib inline

        from matplotlib import pyplot as plt
        import utils
        import numpy as np
        import statsmodels.api as sm
        from scipy import stats
```

Read the data as a pandas dataframe. For refernece: https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.html

```
In [3]: # Read the data
        filename = 'cars.csv'
        df = utils.read_dataset_from_csv(filename)
```

(0a). Print the number of observations in the dataset.

```
In [4]: print(len(df))
```

```
32
```

(0b). Print all the columns in the dataset

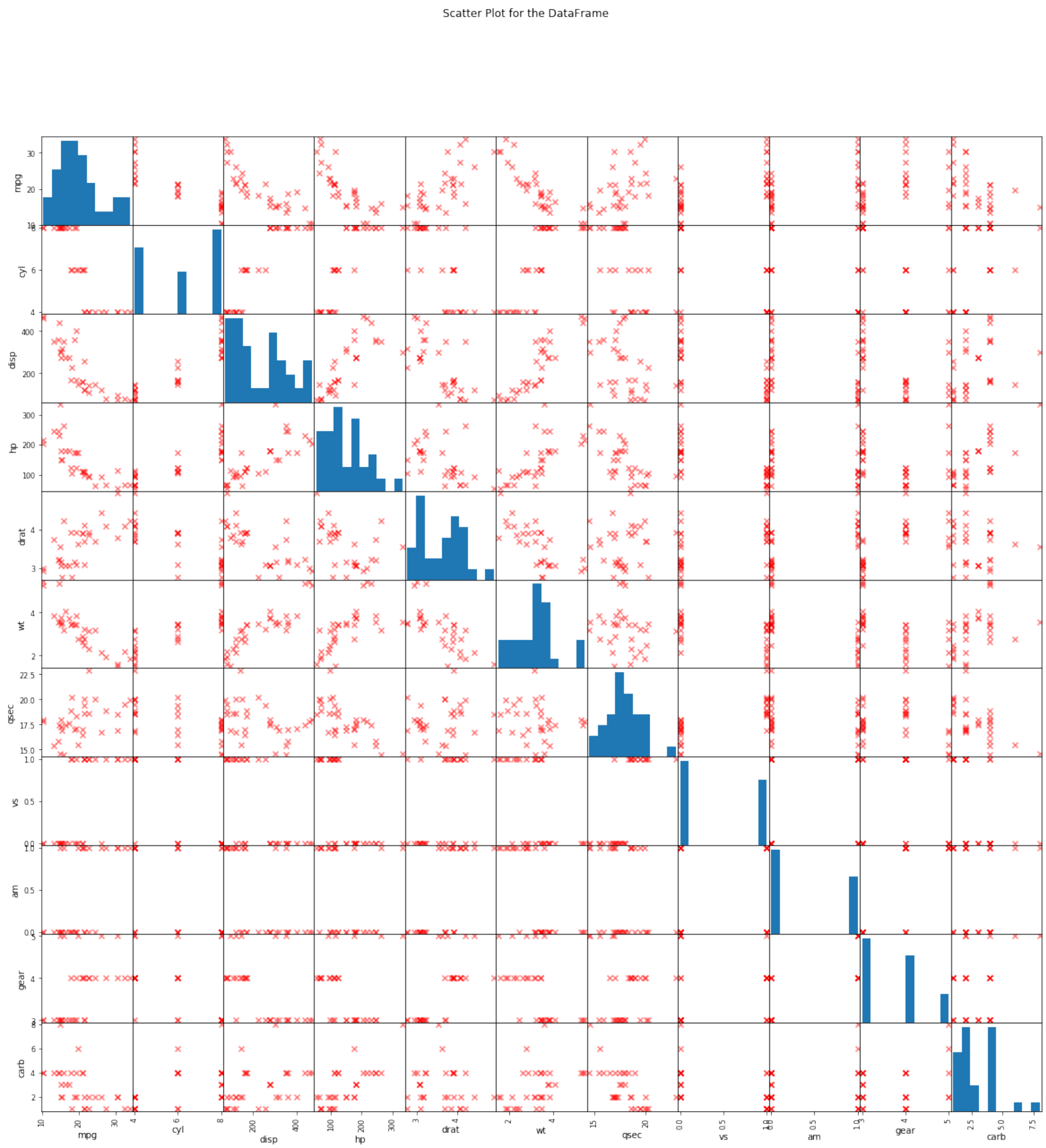
```
In [5]: col_names = df.columns.values
        col_names_str = ",".join(col_names)

        print(col_names_str)
```

```
name,mpg,cyl,disp,hp,drat,wt,qsec,vs,am,gear,carb
```

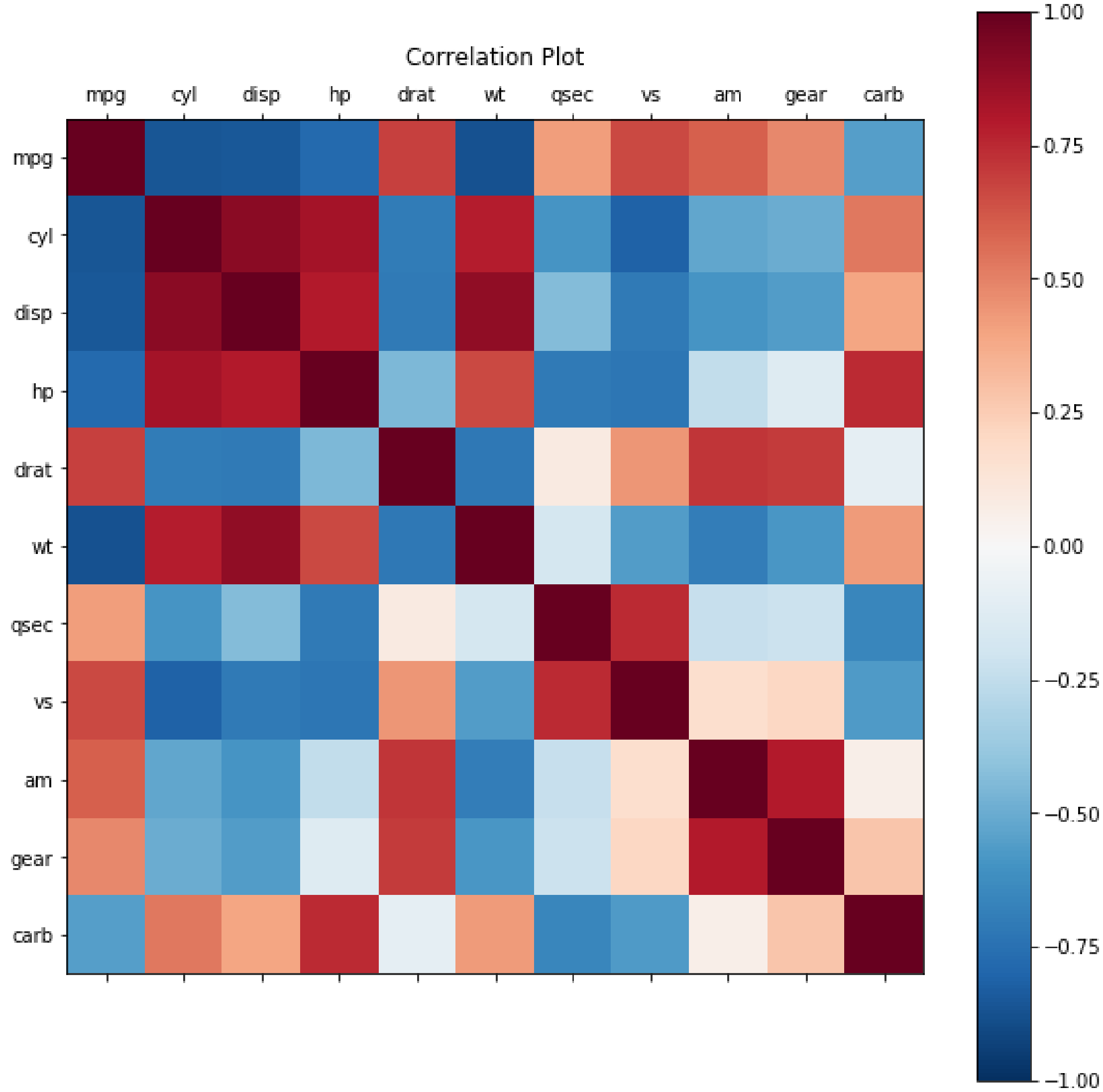
(0c).Produce a Scatter plot of all variables against each other. Feel free to use the scatter_plot_dataframe() function in utils.py. Note: this function call may take a while.

```
In [5]: utils.scatter_plot_dataframe(df)
```



(0d). Produce a plot of correaltions between all variables. Feel free to use the correlation_plot() function in utils.py

```
In [6]: utils.correlation_plot(df)
```



(0e). Using the plots above, which variables have a (roughly) linear relationship with 'mpg'?
cyl, disp, wt

(1). Ridge Regression. We will run Ridge regression by introducing an L2 penalty on the regression coefficients.

(1a). Build a simple OLS model using statsmodels.OLS Your dependent variable is 'mpg'. The independent variables are 'cyl', 'disp', 'hp', 'drat', 'wt', 'qsec', 'vs', 'am', 'gear', 'carb'. Do include the intercept using the add_constant() function in statsmodels. Hint: http://www.statsmodels.org/dev/generated/statsmodels.regression.linear_model.OLS.html
Store your multi-variate model in a variable called sv_model

```
In [13]: Y = df.mpg.values
         X = df[['cyl', 'disp', 'hp', 'drat', 'wt', 'qsec', 'vs', 'am', 'gear', 'carb']].as_matr
         X = sm.add_constant(X)
         sv_model = sm.OLS(Y,X)
```

(1b). Use statsmodels' fit_regularized() function to write a function get_sv_ridge() which fits the model with an L2 penalty of weight α and returns the output.

```
In [18]: def get_sv_ridge(alpha):
             return sv_model.fit_regularized(alpha=alpha, L1_wt=0)
```

(1c). Use get_sv_ridge() to plot the Ridge regression coefficients vs. α for each independent variable and for α in the range [0,2].

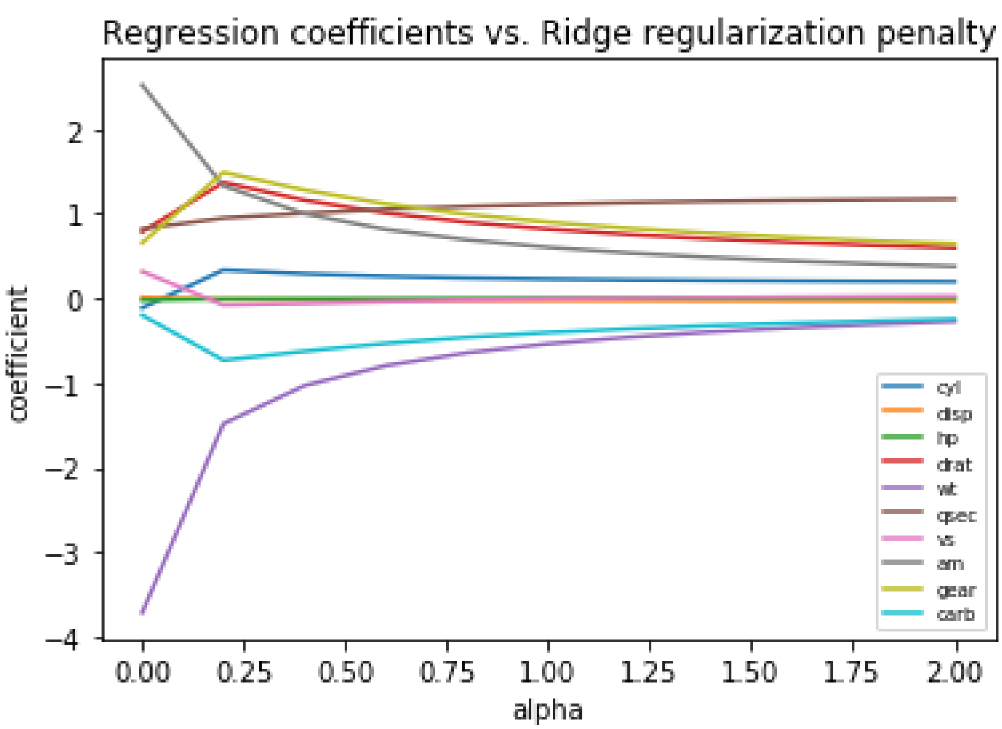
```
In [47]: coeff_array = [[] for i in range(10)]
         alpha_vals = []

         for i in range(11):
             alpha = 2 * i / 10
             alpha_vals.append(alpha)

             fit = get_sv_ridge(alpha)
             for j in range(10):
                 coeff_array[j].append(fit.params[j+1])

         ind_cols = col_names[2:]

         for j in range(10):
             plt.plot(alpha_vals, coeff_array[j], label=ind_cols[j]);
         plt.xlabel("alpha")
         plt.ylabel("coefficient")
         plt.title("Regression coefficients vs. Ridge regularization penalty")
         plt.legend(loc='lower right', fontsize='x-small');
```



(2). LASSO (Least Absolute Shrinkage and Selection Operator). We will now run LASSO by introducing an L1 penalty on the regression coefficients.

(2a). Use statsmodels' fit_regularized() function to write a function get_sv_lasso() which fits the model with an L1 penalty of weight α and returns the output.

```
In [42]: def get_sv_lasso(alpha):
             return sv_model.fit_regularized(alpha=alpha, L1_wt=1)
```

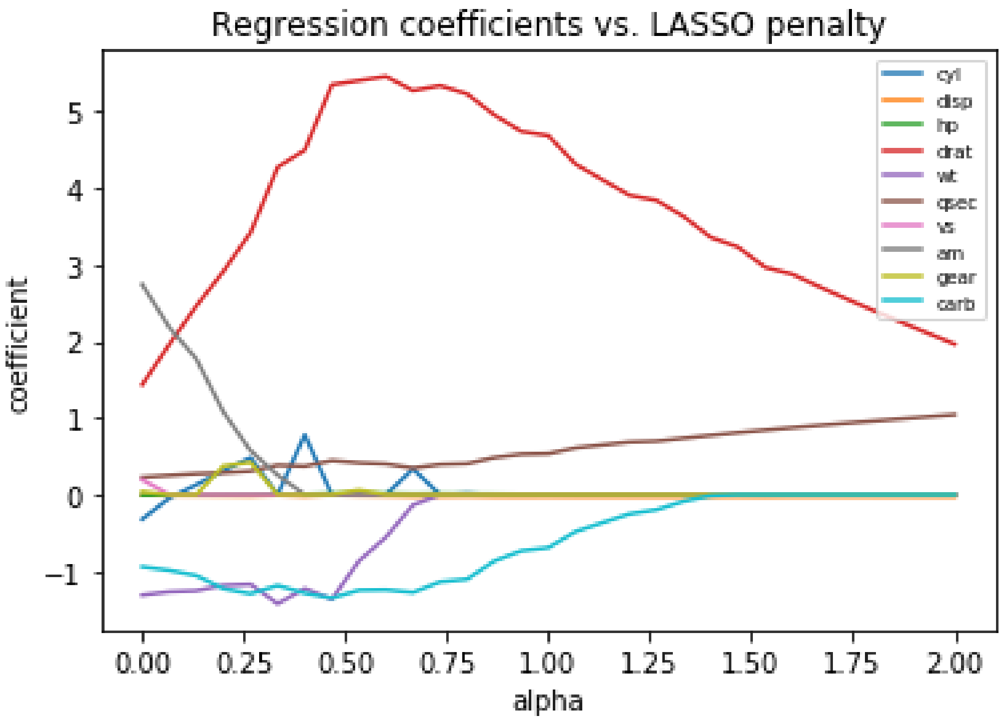
(2b). Use get_sv_lasso() to plot the LASSO coefficients vs. α for each independent variable and for α in the range [0,2].

```
In [50]: coeff_array = [[] for i in range(10)]
         alpha_vals = []

         for i in range(31):
             alpha = 2 * i / 30
             alpha_vals.append(alpha)

             fit = get_sv_lasso(alpha)
             for j in range(10):
                 coeff_array[j].append(fit.params[j+1])

         for j in range(10):
             plt.plot(alpha_vals, coeff_array[j], label=ind_cols[j]);
         plt.xlabel("alpha")
         plt.ylabel("coefficient")
         plt.title("Regression coefficients vs. LASSO penalty")
         plt.legend(loc='upper right', fontsize='x-small');
```



Regression coefficients vs. LASSO penalty
coefficient
alpha
cyl
disp
hp
drat
wt
qsec
vs
am
gear
carb