- **Programmer:**
  - **Shaun Pritchard**
  - **Ismael A Lopez**

Assigment 2

Brief overview of assignment

You will perform a penalized (regularized) least squares fit of a linear model using elastic net, with the model parameters obtained by coordinate descent

---

Overview: Primary - Assignment 2

We analyzed the credit card data from N=400 training observations that you examined in Programming Assignment 1 using a penalized (regularized) least squares fit of a linear model using elastic net, with model parameters obtained by coordinate descent.

Initially, we each worked independently, then we collaborated afterwards to finalize the assignment deliverables. This resulted in the completion of 2 methods for achieving the same goal, namely implementing ElastNet with coordinate descent. This is the second take on assignment 2. The aim was to display different methods of achieving the same abstraction.

> ***Table of Contents***

✓ 0s completed at 3:23 PM

> ***Deliverables***

- **Deliverable 6.1**
- **Deliverable 6.2**
- **Deliverable 6.3**
- **Deliverable 6.4**
- **Deliverable 6. Reason for difference**

Import Packages

Import packages for manipulating data

```
 1 import numpy as np
 2 import pandas as pd
 3 import matplotlib.pyplot as plt
 4 import matplotlib as mpl
 5 import matplotlib.mlab as mlab
 6 import math
 7 import csv
 8 import random
 9 %matplotlib inline
10
```

Import packages for splitting data

```
 1 from sklearn.model_selection import train_test_split, cross_val_score
 2 from sklearn.model_selection import GridSearchCV
 3
```

Import packages for modeling data

```
10 #from sklearn.utils._testing import ignore_warnings
11 import warnings
12 warnings.filterwarnings('ignore', category=ConvergenceWarning) # To f
13 warnings.filterwarnings('ignore', category=UserWarning)
14 from itertools import product
15
```

Import packages for Scaling and Centering data

```
1 from sklearn.preprocessing import StandardScaler
```

Import packages for Measuring Model Perormance

```
1 from sklearn.metrics import mean_squared_error
2 from sklearn.metrics import r2_score
3 from sklearn.metrics import make_scorer
```

Data Processing

Import Data

[ ] ↳ *3 cells hidden*

Lets change the categorical values

[ ] ↳ *3 cells hidden*

Define our learning rates

[ ] ↳ *3 cells hidden*

Create the Regression Objects

[ ] ↳ *6 cells hidden*

Deliverable 6.1

> Illustrate the effect of the tuning parameter on the inferred elastic net regression coefficients by generating six plots (one for each α value) of nine lines (one for each of the p=9 features), with the y-axis as $\beta\hat{}j$, j=1,2,...,9, and the x-axis the corresponding log-scaled tuning parameter value log10(λ) that generated the particular $\beta\hat{}j$.

Manual:

```
1 Mβ_per_λ = []
2 tP = []
3 for tuning_param in tuning_params:
4     cord = coordinateDescent(Xst, y_Centered , tuning_param[1] , tuni
```

```
plt.subplot(4, 2, 1)
plt.plot(TunnedMβ_df[TunnedMβ_df.Alpha.eq(0)].iloc[:,1:2],TunnedMβ_df

plt.title('Effect of tunning on Coefficients alpha = 0')
plt.ylabel('Standardize Coefficients')

plt.xscale('log')
plt.legend(loc='best')
plt.legend(TunnedMβ_df.columns[2:])


plt.subplot(4, 2, 2)
plt.plot(TunnedMβ_df[TunnedMβ_df.Alpha.eq(0.2)].iloc[:,1:2],TunnedMβ_

plt.title('Effect of tunning on Coefficients alpha = 0.2')

plt.xscale('log')
plt.legend(loc='best')
plt.legend(TunnedMβ_df.columns[2:])


plt.subplot(4, 2, 3)

plt.plot(TunnedMβ_df[TunnedMβ_df.Alpha.eq(0.4)].iloc[:,1:2],TunnedMβ_
```

```
plt.xscale('log')
plt.legend(loc='best')
plt.legend(TunnedMβ_df.columns[2:])


plt.subplot(4, 2, 6)

plt.plot(TunnedMβ_df[TunnedMβ_df.Alpha.eq(1)].iloc[:,1:2],TunnedMβ_df

plt.title('Effect of tunning on Coefficients alpha = 1')
plt.xlabel('Learning Rates λ')

plt.xscale('log')
plt.legend(loc='best')
plt.legend(TunnedMβ_df.columns[2:])

```

```
plt.title('Effect of tunning on Coefficients alpha = 0')
plt.ylabel('Standardize Coefficients')

plt.xscale('log')
plt.legend(loc='best')
plt.legend(TunnedLβ_df.columns[2:])
```

```
plt.legend(tunedLp_df.columns[2:])
```