Tarea3_REGRESSION

December 14, 2017

1 Tarea 3. Regresión.

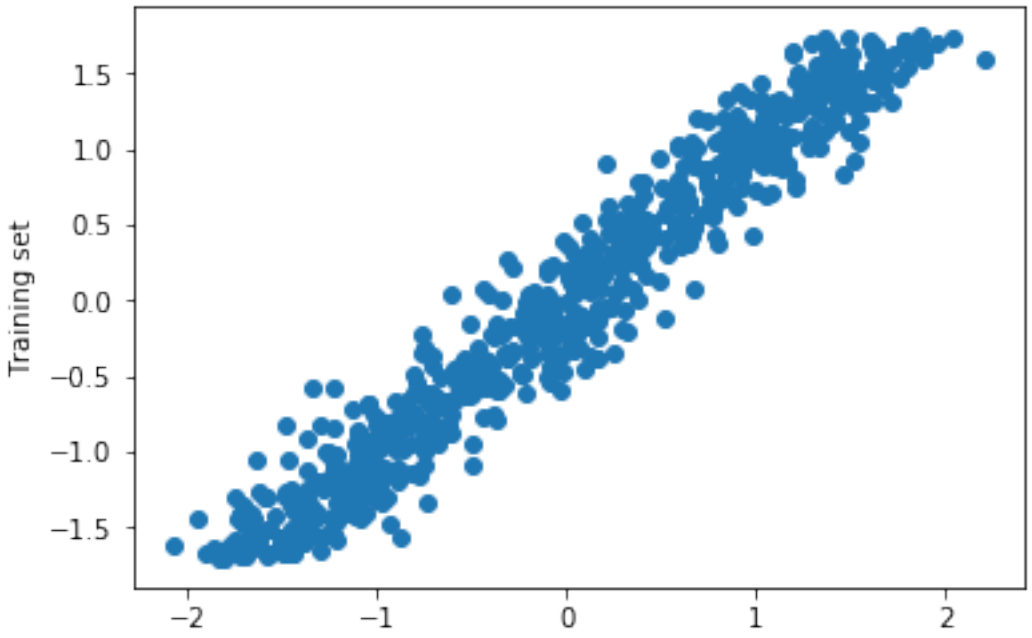
L.E. Rojón
138442

In [22]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import sklearn

from math import ceil
from sklearn.metrics import mean_squared_error
import numpy as np
from sklearn import preprocessing, linear_model
```

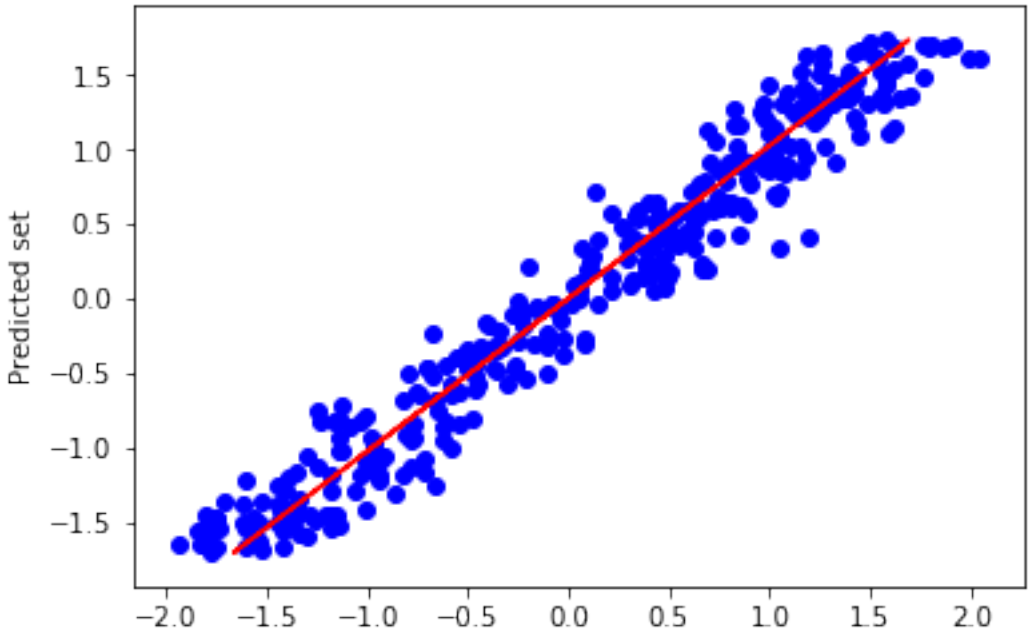
In [23]:
```
d = pd.read_csv("/home/luxorville/Documentos/Oto2017/Machine_Learning/regLin.csv")
```

In [24]:
```
X_train, X_test, y_train, y_test = train_test_split(pd.DataFrame(d['X']), pd.DataFrame(


normalizer = preprocessing.StandardScaler()
normalizer.fit(X_train)
X_train = pd.DataFrame(normalizer.transform(X_train))
X_test = pd.DataFrame(normalizer.transform(X_test))

normalizer.fit(y_train)
y_train = pd.DataFrame(normalizer.transform(y_train))
y_test = pd.DataFrame(normalizer.transform(y_test))
```

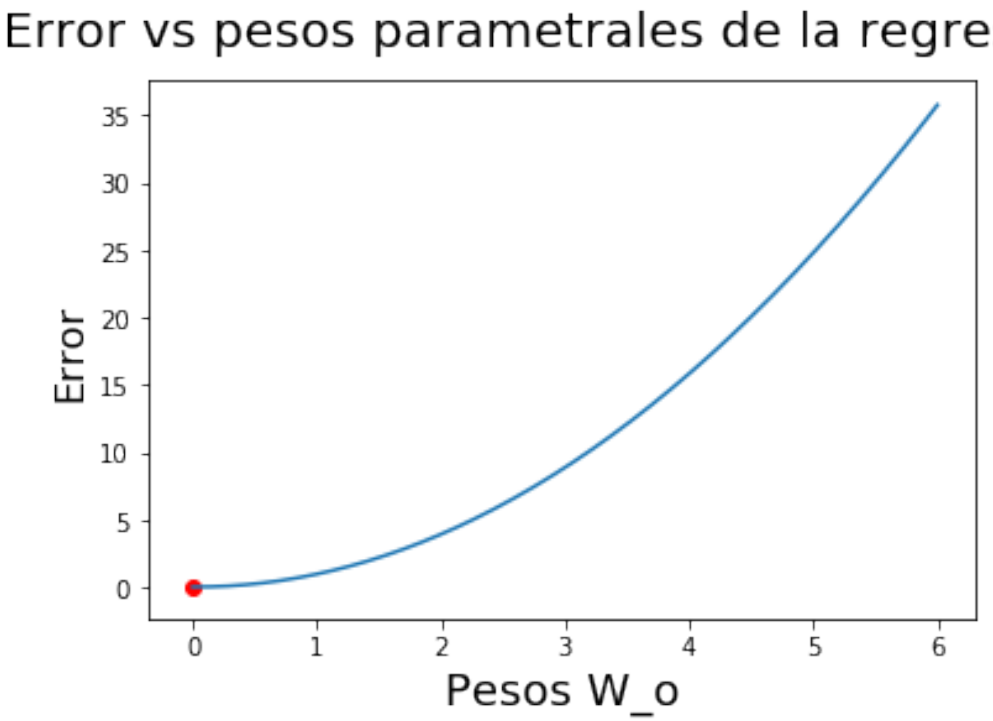
In [25]:
```
plt.scatter( y_train, X_train)
plt.ylabel('Training set')
plt.show()
```



In [26]:
```
reg=linear_model.LinearRegression()
reg.fit(X_train, y_train)
y_predict=reg.predict(X_test)
plt.plot( y_predict, X_test, color='red')
plt.scatter( y_test, X_test, color='blue')
plt.ylabel('Predicted set')
plt.show()
```



```
In [27]: a=reg.intercept_[0]
         b=reg.coef_[0][0]
         print a
         print b

-1.0666446531e-16
0.97287024552


In [28]: def predictionreg(X, a, b):
             X2=a+b*X
             return pd.DataFrame(X2).rename(index=str, columns={"X": "y"})

         def errori(Y1, Y2):
             return np.sum((Y1.values-Y2.values)**2)

In [30]: def calcError(X, y, w, w0):
                 return np.mean((w0 + w*X.values- y.values)**2)
         w0 = np.linspace(0,6,len(X_test))
         w1 = np.linspace(-2,3,len(X_test))

         inter, c = a, b
         errpred= calcError(X_test, y_test, c, inter)

         err0 = [calcError(X_test, y_test, c, w0[i]) for i in range(0,len(w0))]
         err1 = [calcError(X_test, y_test, w1[i], inter) for i in range(0,len(w1))]

In [31]: fig=plt.figure()
         plt.plot(w0,err0)
         fig.suptitle('Error vs pesos parametrales de la regresion', fontsize=20)
         plt.xlabel('Pesos W_o', fontsize=18)
         plt.ylabel('Error', fontsize=16)

         plt.scatter(inter, errpred, color='red')
         plt.show()
```



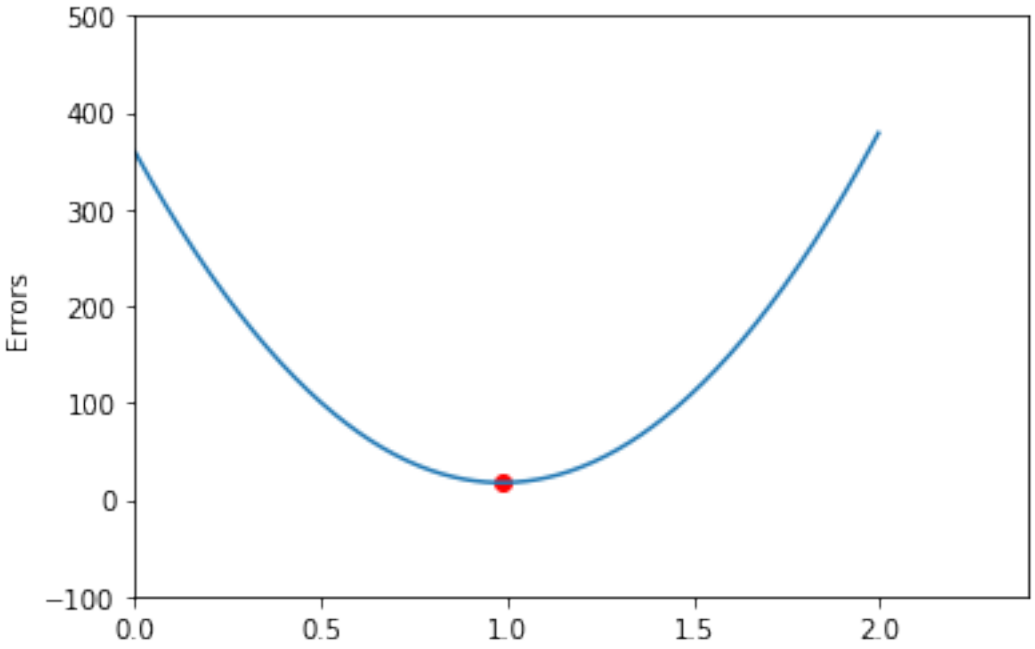

```
In [100]: epsilon=[]
          weight=[]
          miny=errori(y_test, predictionreg(X_test, 100, 100))
          minx=100
```

```
In [101]: for bt in range( int(ceil(-b)*1000), int(ceil(2*b)*1000), int(5*b)):
              br=bt/1000.0
              epsilon.append(errori(y_test, predictionreg(X_test, a, br)))
              weight.append(br)
              if errori(y_test, predictionreg(X_test, a, br))<miny:
                  miny=errori(y_test, predictionreg(X_test, a, br))
                  minx=br
```

```
In [102]: print "minimum is "+str(minx)
          print "real minimum is "+str(b)
```

```
minimum is 0.984
real minimum is 0.97287024552
```

```
In [104]: plt.plot(weight, epsilon)
          plt.scatter(  minx, miny, color='red')

          plt.axis([0, 2.4, -100, 500])
          plt.ylabel('Errors')
          plt.show()
```


500
400
300
200
100
0
−100
Errors
0.0
0.5
1.0
1.5
2.0