Ejercicios 2

Tṕicos de Investigación CM-072

Scipy, Matplotlib, Probabilidad, scikit-learn

1. Explica en términos matemáticos que es lo produce el siguiente código fuente

```
import numpy as np
from scipy import optimize
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D


def f1(x):
    return (4 - 2.1*x[0]**2 + x[0]**4 / 3.) * x[0]**2 + x[0] * x[1] + (-4 + \
        4*x[1]**2) * x[1] **2

x = np.linspace(-2, 2)
y = np.linspace(-1, 1)
xg, yg = np.meshgrid(x, y)

plt.figure()
plt.imshow(f1([xg, yg]))
plt.colorbar()

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
surf = ax.plot_surface(xg, yg, f1([xg, yg]), rstride=1, cstride=1,
                       cmap=plt.cm.jet, linewidth=0, antialiased=False)

ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('f(x, y)')
ax.set_title('titulo')
```

2. Explica línea por línea el siguiente código

```
import numpy as np
from sklearn import linear_model
from sklearn.datasets.samples_generator import make_regression
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

X, y = make_regression(n_samples=100, n_features=2, n_informative=1,
                        random_state=0, noise=50)

X_train, X_test = X[:80], X[-20:]
y_train, y_test = y[:80], y[-20:]
```

```
regr = linear_model.LinearRegression()


regr.fit(X_train, y_train)


print(regr.coef_)


X1 = np.array([1.2, 4])
print(regr.predict(X1))


print(regr.score(X_test, y_test))

fig = plt.figure(figsize=(8, 5))
ax = fig.add_subplot(111, projection='3d')

ax.scatter(X_train[:, 0], X_train[:, 1], y_train, facecolor='#00CC00')
ax.scatter(X_test[:, 0], X_test[:, 1], y_test, facecolor='#FF7800')


coef = regr.coef_
line = lambda x1, x2: coef[0] * x1 + coef[1] * x2

grid_x1, grid_x2 = np.mgrid[-2:2:10j, -2:2:10j]
ax.plot_surface(grid_x1, grid_x2, line(grid_x1, grid_x2),
                alpha=0.1, color='k')
ax.xaxis.set_visible(False)
ax.yaxis.set_visible(False)
ax.zaxis.set_visible(False)
plt.show()
```

3. Explica lo que hace el siguiente código

```
import numpy as np
from sklearn.datasets import load_boston
from sklearn.linear_model import SGDRegressor
from sklearn.cross_validation import cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.cross_validation import train_test_split

data = load_boston()
X_train, X_test, y_train, y_test = train_test_split(data.data, data.target)


X_scaler = StandardScaler()
y_scaler = StandardScaler()
X_train = X_scaler.fit_transform(X_train)
y_train = y_scaler.fit_transform(y_train)
X_test = X_scaler.transform(X_test)
y_test = y_scaler.transform(y_test)

regresor = SGDRegressor(loss='squared_loss')
scores = cross_val_score(regresor, X_train, y_train, cv=5)
print ('Validacion cruzada r2 score:', scores)
print ('Promedio de la validacion cruzada  r2 score:', np.mean(scores))
```

```
regressor.fit_transform(X_train, y_train)
print ('Prueba r2 score', regressor.score(X_test, y_test))
```

4. Considera una secuencia de lanzamientos infinitos. La probabilidad de un éxito en el i-ésimo lanzamiento es algún número positivo p_i. Sea N el evento que no hay éxitos y sea I el evento donde hay un número infinito de éxitos.

    - Asumiendo que los lanzamientos son independientes y que $\sum_{i=1}^{\infty} p_i = \infty$. Muestra que $P(N) = 0$ y $P(I) = 1$.
    - Asumimos que $\sum_{i=1}^{\infty} p_i < \infty$. Muestra que $P(I) = 0$.

5. Supongamos que X e Y son variables aleatorias geométricas con parámetro p, independientes, identicamente distribuidas. Muestra que

$$P(X = i | X + Y = n) = \frac{1}{n-1}, \quad i = 1, \ldots, n-1.$$

6. Considera el siguiente `pdf`

$$f_X(x) = \begin{cases} p\lambda e^{-\lambda x} & \text{si} \;\; x \geq 0, \\ (1-p)\lambda e^{\lambda x} & \text{si} \;\; x < 0 \end{cases}$$

   donde λ y p son escalares con $\lambda > 0$ y $p \in [0, 1]$. Encuentra la media y varianza de X de dos maneras:

    - calculando la esperanza asociada.
    - usando la estrategia divide y vencerás y la media y varianza de la variable aleatoria exponencial.

7. Sean X e Y variablea aleatorias normales estándar indepedientes. El par (X, Y) puede ser descrita en coordenadas polares de la siguiente forma

$$X = R\cos\Theta, \quad Y = R\sin\Theta$$

   para $R \leq 0$ y $\Theta \in [0, 2\pi]$.

    - Muestra que Θ es uniformemente distribuida en $[0, 2\pi]$, que R tiene un `pdf`

$$f_R(r) = re^{-r^2/2} \quad r \geq 0$$

      y que R y Θ son independientes.
    - Muestra que R^2 tiene una distribución exponencial con parámetro $1/2$.