▾ Grade: /100 points

Assignment 01: Supervised learning, Linear models, and Loss functions

In this assignment, you're going to write your own methods to fit a linear model using OLS and LAD cost functions.

Data set

For this assignment, we will examine some data representing possums in Australia and New Guinea. The data frame contains 46 observations on the following 6 variables:

- sex: Sex, either m (male) or f (female).
- age: Age in years.
- headL: Head length, in mm.
- skullW: Skull width, in mm.
- totalL: Total length, in cm.
- tailL: Tail length, in cm.

Follow These Steps Before Submitting

Once you are finished, ensure to complete the following steps.

1. Restart your kernel by clicking 'Kernel' > 'Restart & Run All'.

2. Fix any errors which result from this.

3. Repeat steps 1. and 2. until your notebook runs without errors.

4. Submit your completed notebook to OWL by the deadline.

Before you start recall that

L1 loss function (sum of magnitudes, used for LAD model):

$$L_1\left(\theta\right) = \sum_{i=1}^{n}|y_i - \hat{y_i}|$$

L2 loss function (RSS, residual sum of squares, used for OLS model):

$$L_2\left(\theta\right) = \sum_{i=1}^{n}(y_i - \hat{y_i})^2$$

▼ Global Toolbox

```
import  matplotlib.pyplot  as  plt
import  numpy  as  np
import  pandas  as  pd
import  scipy.stats  as  ss
import  scipy.optimize  as  so
%matplotlib  inline
```

▼ Part 1: OLS Regression

Question 1.1: /10 points

Read in the `possum.csv` file as a `pandas.DataFrame`. Investigate the relationship between the possum's age and its tail length by plotting a scatter plot of the `age` and `tailL` columns. Add an `alpha` (transparency of the plotted dots) in case some data are overlapping.

```
#  Read  in  the  data  with  pandas
possum_data  =  pd.read_csv("possum.csv")
#  Make  the  scatter  plot  (don't  forget  the  axis  labels)
possum_data.plot(kind  =  "scatter",  x  =  "age",  y  =  "tailL",  alpha  =  0.5)
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[1], line 2
      1 # Read in the data with pandas
----> 2 possum_data = pd.read_csv("possum.csv")
      3 possum_data.plot(kind = "scatter", x = "age", y = "tailL")
      4 plt.show()

NameError: name 'pd' is not defined
```

在 STACK OVERFLOW 中搜索

▼ Question 1.2: /5 point

Recall that the linear model, we obtain predictions by computing

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta}$$

Here, $\mathbf{X}$ is a design matrix which includes a column of ones, $\hat{\beta}$ are coefficients, and $\hat{\mathbf{y}}$ are outcomes. Write a function `linearModelPredict` to compute linear model predictions given data and a coefficient vector. The function should take as it's arguments a 1d-array of coefficients `b` and the design matrix `X` as a 2d-array and return linear model predictions `yp`.

Test the function by setting

```
X  =  np.array([[1,0],[1,-1],[1,2]])
b  =  np.array([0.1,0.3])
```

and call your function with these values!

Report $\hat{\mathbf{y}}$. What is the dimensionality of the numpy-array that you get back?

Hint: Read the documentation for `np.dot` or the `@` operator in `numpy`.

```
def  linearModelPredict(b,X):
      yp  =
      return  yp

#  Always  important:  Test  the  new  function  you  have  written!

#  By  the  way:  What  happens  when  b  is  a  2d-array?
```

▾ Question 1.3: /15 points

Write a function `linearModelLossRSS` which computes and returns the loss function for an OLS model parameterized by β, as well as the gradient of the loss. The function should take as its first argument a 1d-array `beta` of coefficients for the linear model, as its second argument the design matrix `X` as a 2d-array, and as its third argument a 1d-array `y` of observed outcomes.

Test the function with the values

```
X  =  np.array([[1,0],[1,-1],[1,2]])
b  =  np.array([0.1,0.3])
y  =  np.array([0,0.4,2])
```

Report the loss and the gradient.

Written answer: To minimize the cost do you need increase or decrease the value of the parameters?

```
def  linearModelLossRSS(b,X,y):

      return  (residual_sum_of_squares,  gradient)
```

▾ Question 1.4: /15 points.

Now that you've implemented a loss function in question 1.3, it is now time to minimize it!

Write a function `linearModelFit` to fit a linear model. The function should take as its first argument the design matrix `X` as a 2d-array, as its second argument a 1d-array `y` of outcomes, and as its third argument a function `lossfcn` which returns as a tuple the value of the loss, as well as the gradient of the loss. As a result, it should return the estimated betas and the R2.

Hint: Using `scipy.optimize.minimize` to minimize the customized loss function

Test the function with the values:

```
X  =  np.array([[1,0],[1,-1],[1,2]])
y  =  np.array([0,0.4,2])
```

Report best parameters and the fitted R2

```
def  linearModelFit(X,y,lossfcn  =  linearModelLossRSS):

      return  (estimated_betas,R2)
```

▾ Question 1.5: /15 points

Use the above functions to fit your model to the possum data. Then use your model and the fitted parameters to make predictions along a grid of equally spaced possum ages.

Hint : Don't forget to include a column of ones in your design matrix to allow bias

Plot the data and add a line for the predicted values. You can get these by generating a new X-matrix with equally space ages (using for example np.linspace). Also report the R2 value for the fit. You can do this by either printing out the R2 of the fit or putting it on your plot via the `annotate` function in matplotlib.

```
#  Make  the  design  matrix  using  np.c_
#  y  =  ...
#  X  =    np.c_[...]
#  Call  your  fitting  function

#  Create  the  scatter  plot  (see  question  1.1)

#  Create  a  new  X  matrix  with  equally  space  data

#  Add  the  line  to  the  graph

#  Report  R2
```

▾ Part 2: LAD Regression

Question 2.1: /13 points

In the previous section, we worked with the squared loss. Now, we'll implement a linear model with least absolute deviation loss.

Write a function `linearModelLossLAD` which computes the least absolute deviation loss function for a linear model parameterized by β, as well as the gradient of the loss. The function should take as its first argument a 1d-array `beta` of coefficients for the linear model, as its second argument the design matrix `X` as a 2d-array, and as its third argument a 1d-array `y` of observed outcomes.

Test the function with the values

```
X = np.array([[1, 0], [1, -1], [1, 2]])
b = np.array([0.1, 0.3])
y = np.array([0, 0.4, 2])
```

Report the loss and the gradient.

```
def linearModelLossLAD(b, X, y):

    return (sum_abs_dev, grad)
```

Question 2.2: /8 points

Use the above functions to fit your LAD model. Use your model to make predictions along a grid of equally spaced possum ages. Once fit, add the fitted line to the scatter plot as in question 1.5. Also report the R2-value.

Written answer: What is the difference in the fit obtained with an L1 as compared to the L2 cost function? Which one has a higher R2 value? Why?

Note: If you recieve an error from the optimizer, it may be because the loss function for the LAD model is not differentiable at its minimum. This will lead to some gradient based optimizers to fail to converge. If this happens to you then pass `method="Powell"` to `scipy.optimize.minimize`.

Written answer: The LAD fit does not give as much weight to the outlier (9,55) as the OLS fit. The R2 value is lower, however. This is because OLS minimized the RSS, and therefore maximizes R2.

Part 3: OLS Regression using Sklearn

Question 3.1: /7 points

Fit an OLS model to the possum data with the `linear_model` module from the `sklearn` package by using the `LinearRegression` class. In no more than two sentences, comment on the rsquared values from `sklearn` and the rsquared values from your models. Are they similar?

▾ Part 4: A new loss function

Question 4.1: /4 points

Assume that the following function has been proposed to you to use as a new loss function for regression:

$$L_3\left(\hat{\beta}\right) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^3$$

What do you think about this new loss function, would you accept it? Please, explain.

** Answer in this cell **

▾ Question 4.2: /8 points

Regardless of your answer to Question 4.1, taking $\hat{\mathbf{y}} = \mathbf{X}^2\hat{\beta}$ as your functional form for regression, use your knowledge of calculus to calculate the gradient of $L_3\left(\hat{\beta}\right)$. Type your answer (including the intermediate steps) in the cell below in Latex format. Note that $\hat{\beta} = [\beta_0, \beta_1]^T$, and $\mathbf{X}$ is an n-by-2 matrix.

$your_{answers}$