Introduction to AI: assignment 2 - EUR DS4H

Linear regression

Part I: Theoretical questions

A. Linear regression or not?

Indicate which of the prediction models below correspond to linear regression. If the model corresponds to linear regression indicate how to construct the matrix $\mathbf{X}$ for regression (suppose K observations are available). If the model does not correspond to linear regression, explain why.

1. $\hat{y} = \beta_1 x$
2. $\hat{y} = \beta_0 + \beta_1 x^2$
3. $\hat{y} = \beta_0 - 1.2\, x^{\beta_1}$
4. $\hat{y} = \beta_0 - 1.5\,\beta_1\, x^{(1)} + 2\,\beta_2\, x^{(2)}$
5. $\hat{y} = \beta_0 - \sum_{i=1}^{p} \beta_i \sin(ix)$
6. $\hat{y} = \beta_0 - \sum_{i=1}^{p} \sin(\beta_{i,1} x)$
7. $\hat{y} = \dfrac{1}{1 + \exp\left\{-\beta_0 - \sum_{i=1}^{p} \beta_i\, x^{(i)}\right\}}$

B. Transformation of variables

Can the following prediction models be transformed into linear regression models? If yes, explain how.

1. $\hat{y} = \exp\{\beta_0 + \beta_1 x\}$
2. $\hat{y} = \beta_0\, 2^{\beta_1 x}$
3. $\hat{y} = \beta_2\, 2^{\{\beta_0 + \beta_1 x\}} + \beta_3$
4. $\hat{y} = \beta_0\, x^{\beta_1}$

1) Yes. $X = \begin{bmatrix} x_1 \\ \vdots \\ x_K \end{bmatrix}$

2) Yes. $X = \begin{bmatrix} 1 & x_1^2 \\ \vdots & \vdots \\ 1 & x_K^2 \end{bmatrix}$

3) No. It cannot be put in the form $Y = X\beta$

4) Yes. $X = \begin{bmatrix} 1 & -1.5\, x_1^{(1)} & 2\, x_1^{(2)} \\ \vdots & \vdots & \vdots \\ 1 & -1.5\, x_K^{(1)} & 2\, x_K^{(2)} \end{bmatrix}$

5) Yes. $X = \begin{bmatrix} 1 & -\sin(x_1) & -\sin(2x_1) \dots & -\sin(px_1) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & -\sin(x_K) & -\sin(2x_K) \dots & -\sin(px_K) \end{bmatrix}$

6) and 7) No. They cannot be put in the form $X\beta$.

Yes, if $\hat{y} > 0$, do $\ln(\hat{y}) = \hat{y}'$, then $\hat{y}' = \beta_0 + \beta_1 x$

Yes, if $\hat{y} > 0$, do $\log_2(\hat{y}) = \hat{y}'$, then $\hat{y}' = \underbrace{\log_2(\beta_0)}_{\beta_0'} + \beta_1 x$

No.

Yes, if $\hat{y} > 0$ and $x > 0$, $\ln(\hat{y}) = \hat{y}'$, then

$\hat{y}' = \underbrace{\ln(\beta_0)}_{\beta_0'} + \beta_1 \cdot \underbrace{\ln(x)}_{x'} = \beta_0' + \beta_1 x'$

Part II: Sales prediction based on advertising budgets

In this part you are going to use the data set contained in the file *Advertising.csv*[1]. In this data set, the sales of a product, in thousands of units, for 200 different markets are given. For each market, it is also given the budget, in thousands of dollars, spent in advertising in 3 different media: TV, radio and newspaper. The objective is to find an adequate prediction model linking the explanatory variables, *i.e.* advertising spending on each medium, with the output variable, the sales. In order to achieve this objective, we are going to test and compare different linear regression approaches.

[1] Source: `https://www.statlearning.com/resources-second-edition`

A. Reading and visualizing the advertising data set

1. Launch *Jupyter* and create a new notebook named *labwork_2*.

2. To read the csv file *Advertising.csv*, you can use a library called *pandas*. Import the library with

   ```
   import pandas as pd
   ```

   pandas documentation can be found at `https://pandas.pydata.org/pandas-docs/stable/`.

3. Read the csv file and store its content in a data frame with the command *read_csv* from *pandas*:

   ```
   df = pd.read_csv("path_to_the_file")
   ```

   In *Windows*, it may be necessary to use *r"path_to_the_file"* in the argument of the function due to encoding issues.

4. Print the content of the data frame.

5. If the data frame contain an extra unnamed column, remove it with the command drop from *pandas*:

   ```
   df.drop(df.columns[0], axis=1)
   ```

   Print the content of the modified data frame.

6. Import *numpy* and, with the command *to_numpy* from *pandas*, generate a feature matrix $\mathbf{X}$ with the columns corresponding to advertising budgets for different media: $\mathbf{x}^{(1)}$-TV, $\mathbf{x}^{(2)}$-radio, $\mathbf{x}^{(3)}$-newspaper. With the same command generate the output vector $\mathbf{y}$ with the sales. Print the first row of these arrays to check if they are correct.

7. Import *pyplot* from *matplotlib* and generate separate scatter plots of $\mathbf{y}$ against each feature.

8. Generate one figure with the 3 scatter plots using the command *subplots* from *pyplot*. For example, if you have named the feature matrix X and the output vector y simply do

```
fig, axs = plt.subplots(1, 3, sharey=True)
axs[0].scatter(X[:,0],y)
axs[1].scatter(X[:,1],y)
axs[2].scatter(X[:,2],y)
```

Based on what you see, can simple linear regression based on one of these features be an adequate model to predict sales?

B. Simple linear regression

1. Import linear regression related functions from Scikit-learn with the command:

   ```
   from sklearn import linear_model as lm
   ```

2. To apply simple linear regression using the first data feature $\mathbf{x}^{(1)}$ use the following commands:

   ```
   linreg_1 = lm.LinearRegression()
   x_1 = X[:,[0]]
   linreg_1.fit(x_1, y)
   ```

   Apply simple linear regression separately on each feature of the dataset and retrieve the following quantities:

   (a) The regression coefficients β_0 and β_1.

   (b) The MSE of the prediction model using the data set used to fit it. To do so you can use the following commands:

   ```
   from sklearn import metrics as met
   MSE_1 = met.mean_squared_error(y, linreg_1.predict(x_1))
   ```

   (c) The RMSE.

   (d) The R^2 score. This quantity can be retrieved with the function *r2_score* from the module *metrics* of Scikit-learn (already imported in the commands above).

   (e) Draw the corresponding scatter plot and add the predicted line using the command *plot* from *pyplot*.

   (f) Based on the obtained results, can the sales be predicted adequately with simple linear regression?

C. Multiple linear regression with 2 or 3 explanatory variables

1. Fit a multiple linear regression model to each possible couple of explanatory variables and evaluate model adequacy as previously done for simple linear regression (RMSE and R^2). To do so, you

just need to modify the input features used in the functions *fit* and *predict* above.

Which couple of features is the best to predict sales using multiple linear regression?

2. For the best couple of features, draw the prediction plane and the data points in the same graphic. To help you, an example is given below on how to generate a 3D scatter plot and add to it the prediction plane:

```
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
ax = fig.gca(projection='3d')
# First feature test points
x_test_1 = np.linspace(np.min(X[:,0]),np.max(X[:,0]),20)
# Second feature test points
x_test_2 = np.linspace(np.min(X[:,1]),np.max(X[:,1]),20)
x_grid_1,x_grid_2= np.meshgrid(x_test_1,x_test_2)
X_test   = np.column_stack((x_grid_1.reshape(-1,1)
                            ,x_grid_2.reshape(-1,1)))
# Prediction
y_hat  = linreg.predict(X_test)
y_hat_grid = np.reshape(y_hat,[20,20])

ax.plot_surface(x_grid_1,x_grid_2,y_hat_grid,alpha = 0.2)
ax.scatter(X[:,0], X[:,1], y, c='r');
```

3. Apply multiple linear regression using the 3 features simultaneously. Based on RMSE and R^2 values, do you see a significant improvement in model adequacy?

D. Multiple linear regression with interaction

1. Refine the best model found with multiple linear regression with 2 features by adding an interaction term. Does model adequacy significantly improve?

2. Do a residual/leverage analysis to check if there are outliers and outliers with high leverage. To obtain the leverage scores you will need to evaluate a matrix inverse and to retrieve the diagonal elements of a matrix, these operations can be done with the *numpy* commands *linalg.inv* and *diag* respectively.