Machine Learning and Data Mining
Lab 3: Multi-Linear Regression

Instructions

(1) In this exercise you will use `jupyter-notebook` or `google collaboratory` and `python>=3.5`. Type your code, display the outputs, and write your answers to the questions asked in the notebook.
(2) Upload PDF of your notebook file via *Juno*. Typeset your name, roll number and section on the top of the file. Also, filename has to specified as "your_first_name_last_name_roll_number.PDF".)

Learning Outcomes:

- Identify *collinearity* and *multicollinearity* in data via scatterplots, variance inflation factor
- Identify statistical significance of variables using p-values
- Perform model diagnostics via residual plots to identify *heteroscedasticity*, *outliers, etc.*
- Improve model on a given training sample data

In this exercise you will use the `Advertising` sample dataset of a company. The data contains advertising budget that the company spends in thousands of dollars across different media such as 'TV', 'radio' and 'newspaper'. The 'sales' gives the value of products sold by the company in millions of dollars.

(a) Read the data to a `pandas` data frame object. Then drop the first column "Unnamed:0" as it contains only the instance number.

(b) Perform a multilinear regression of 'sales' onto 'TV', 'radio' and 'newspaper'. Let us call this **model-1**.

(i) What is the equation of the linear model? What are the 95% confidence intervals of the regression coefficients?

(ii) Comment on the goodness of fit.

(iii) How large is the effect of each media on sales? Explain your answer using p-values and confidence intervals.

(c) Plot scatter matrix graphs of the data. Is there any evidence of *collinearity* in the data? Compute values of bivariate correlation, *r*, using the corr()

function between every pair of features to support your explanation. Note: *r* is also sometimes known as Pearson's correlation coefficient.

(d) *Multicollinearity* refers to a situation when three or more features are correlated to each other. If multicollinearity in not removed from data, then erroneous models are produced. "Variance inflation factor" (VIF) of features are used to identify multicollinearity. Compute VIFs of every feature. What do they tell you about the features?

(e) Linear regression makes certain assumptions. These are: (i) mean of residuals or errors is zero, (ii) Equal variance of residuals or errors (also called *Homoscedasticity or same-spread*) (iii) Errors should be uncorrelated with each other.

Examine the validity of these assumptions by performing diagnostics of **model-1** by plotting the 'sales predicted from the model' and 'residuals'. Comment on the presence of *non-linearity*, *outliers*, and *heteroscedasticity*.

(f) Explore another model by removing the feature which has least effect on 'sales' and including an interaction between the pair of features. Let's call this **model-2**. Perform a fit of **model-2** to the data and comment on the goodness of fit as compared to **model-1**. Is **model-2** better than **model-1** if prediction accuracy is concerned**?**

Explain if the interaction term is relevant in explaining 'sales'. Also create diagnostic residual plots of the fit of **model-2**.

(g) Predict the value of 'sales' if TV = 350 and radio = 50 using **model-2**. Specify 95% confidence interval of the prediction.

SOLUTION (A)

In [ ]:

```
import pandas as pd

##SOLUTION (a)
#read file "Advertising.csv" to a dataframe object. Fill in the blank.
df = pd.read_csv(___________)
print(df.head())
print(df.info())
```

In [ ]:

```
##SOLUTION (a) continued
#Drop the first column "Unnamed:0" from the dataframe as it contains the index o
f the instances
dfdata = df.drop(columns='Unnamed: 0')
print(dfdata.head())
print(dfdata.info())
```

SOLUTION (B): Perform a multilinear regression of ‘sales’ onto ‘TV’, ‘radio’ and ‘newspaper’. Let's call the first model as 'model_1'.

In [ ]:

```
import statsmodels.formula.api as smf

#Fit a model: sales = beta_0 + (beta_1 * TV) + (beta_2 * radio) + (beta_3 * news
paper).
#Here, beta_0, beta_1 and beta_2 are the parameters to be determined.
model_1 = smf.ols('sales ~ TV + radio + newspaper', dfdata).fit()
print(model_1.summary())
print("----- p-values ------")
print(model_1.pvalues)
```

b(i) sales = 2.9389 + (0.0458 *TV)* + *(0.1885* radio) + (-0.001 * newspaper).

b(ii) R-squared shows that about 90% of the variability in sales values has been explained by the multilinear model. So, the model fits the data reasonably well.

b(iii) Increase in the budget of radio by 1 unit will produce an increase in sales by 0.1885 units. Increase in the budget of TV by 1 unit will produce a smaller increase in sales by 0.0458 units. Whereas, increase in the budget of newspaper by 1 unit will produce least increase in sales compared to the other media.

The P-value of newspaper is 0.86 or 86%, which is much greater that 5%. It provides evidence that there is no relationship between newspaper and sales. (Note: "no relationship" is also called null hypothesis in text books). Whereas, TV and radio has statistically significant effect on sales as evident from their P-values being less than 5%. The 95 % confidence interval of beta_3 occurs over a range [-0.013, 0.011] which includes the value of zero. So, there is significant chance that beta_3 can take a value of 0. Note that for beta_1 and beta_2, a value of zero does not lie in their respective 95% confidence interval.

Solution (C)

In [ ]:

```
from pandas.plotting import scatter_matrix
scatter_matrix(dfdata, alpha=0.9, figsize=(12,12), diagonal='hist')
```

Our features or predictors are TV, radio, and newspaper. Our response variable is 'Sales'. TV - radio bivariate data appears uncorrelated. There seems to be also weak correlation, if at all, between TV and newspaper, and also between radio and newspaper. So, next we quantify the bivariate correlation coefficients between the features to be more concrete.

In [ ]:

```
#calculate correlation coefficients between a pair of features.
pair_corr_coeff = dfdata.corr()
print(pair_corr_coeff)

#Another style of displaying the above result
pair_corr_coeff.abs().style.background_gradient()
```

From the matrix of correlation coefficients above, we infer the presence of only radio-newspaper correlations (r=0.354). Correlations between other features are even weaker.

In [ ]:

```
#Another style of displaying the correlation coefficients
#Create a color matrix plot of correlation coefficents of the pairs of features.
import matplotlib.pyplot as plt
import numpy as np

%matplotlib inline

plt.matshow(np.abs(pair_corr_coeff))
plt.colorbar()
plt.xticks(range(len(pair_corr_coeff.columns)), pair_corr_coeff.columns, rotatio
n='vertical');
plt.yticks(range(len(pair_corr_coeff.columns)), pair_corr_coeff.columns);
```

Solution (D)

Let us examine VIF factors of the features

In [ ]:

```
# compute VIF of each feature.
from statsmodels.stats.outliers_influence import variance_inflation_factor
from patsy import dmatrices
#gather features
formula1 = 'sales ~ TV + radio + newspaper'

# get y and X dataframes based on this regression:
yvar, Xvar = dmatrices(formula1, dfdata, return_type='dataframe')

#print(Xvar)
# For each Xvar, calculate VIF and save in dataframe
vif = pd.DataFrame()
vif["VIF"] = [variance_inflation_factor(Xvar.values, i) for i in range(Xvar.shap
e[1])]
vif["Predictors"] = Xvar.columns
vif.round(3)
```

What do you conclude from the VIF factors of the features?

State your answer here.

Solution (E)

Perform Diagnostics of the least squares regression fit

In [ ]:

```
#First compute the studentized residuals.
studentized_residuals_model1 = model_1.get_influence().resid_studentized_interna
l
#Second compute the predicted values of Y from the model.
predicted_Y_model1 = model_1.predict()

plt.plot(predicted_Y_model1, studentized_residuals_model1, 'o'), plt.xlabel("pre
dicted sales"), plt.ylabel("Studentized residuals")
```

For a linear model, the residual plot should not show any non-linear pattern.

In other words, one should expect a horizontal trendline in the above plot. Let's find out if this is the case.

Draw a smooth line through the scatter plot via "LOWESS Smoothing". LOWESS (Locally Weighted Scatterplot Smoothing) is a method used in regression analysis to draw a "trendline" through a scatter plot to help us see association or relationships between variables. This line is non-parameteric.

lowess(Y, X, is_sorted=True, frac=0.025, it=0);

Here 'frac' is between 0 and 1 and accounts for the fraction of datapoints used in computing a Y value.

Tune the value of 'frac' to get the smoothed line.

In [ ]:

```
from statsmodels.nonparametric.smoothers_lowess import lowess

multi_lowess_result = lowess(studentized_residuals_model1, predicted_Y_model1, i
s_sorted=False, frac=1/2, it=0)

plt.plot(predicted_Y_model1, studentized_residuals_model1, 'o'), plt.xlabel("pre
dicted sales"), plt.ylabel("Studentized residuals")

plt.plot(multi_lowess_result[:, 0], multi_lowess_result[:, 1], 'r-', linewidth=3
)

plt.xlabel('predicted sales'), plt.ylabel('studentized residual')
```

The trendline in orange color is not horizontal. So our assumption of the linear model is not satisfied. There is evidence of non-linear or parabolic pattern in the residual plot. How can we improve the model to remove the non-linear pattern?

Datapoints whose absolute values of studentized residuals are greater than 3 are considered outliers. Few points are outliers. Can you identify the data points that are outliers?

If we remove the outliers, then we see that the variance or spread of the residuals or errors are approximately constant across the predicted response. So we conclude absence of heteroscedasticity.

SOLUTION: F

We drop the feature "newspaper" because it has least amount of influence on 'sales' as evident from p-values.

Explore model-2: Sales = beta_0 + (beta_1 *TV)* + *(beta_2* radio) + (beta_3 *TV* radio)

We have introduced a new term which is a product of pair of features TV and radio. This product is called a "synergy or interaction" term. Note that this model so constructed is not linear anymore because of the new interaction term.

In [ ]:

```
#Let's create the second model
# Fill the below with the formula of the model-2
model_2 = smf.ols(_____________________, dfdata).fit()
print(model_2.summary())
print("----- p-values ------")
print(model_2.pvalues.sort_values())

print("--- Significant Features - features with p-values < 5% -------")
print(model_2.pvalues[model_2.pvalues<0.05])
```

Is the newly added interaction term in model-2 significant in explaining 'sales'?

Is the accuracy of model_2 better than model_1?

State your answers here.

Diagnostic plots of model_2.

In [ ]:

```
from statsmodels.nonparametric.smoothers_lowess import lowess

#Perform Diagnostics of the least squares regression fit on model_2
#First compute the studentized residuals.
studentized_residuals_model2 = model_2.get_influence().resid_studentized_interna
l
#Second compute the predicted values of Y from the model.
predicted_Y_model2 = model_2.predict()

plt.plot(predicted_Y_model2, studentized_residuals_model2, 'o'), plt.xlabel("pre
dicted sales"), plt.ylabel("Studentized residuals")

multi_lowess_result = lowess(studentized_residuals_model2, predicted_Y_model2, i
s_sorted=False, frac=1/2, it=0)

plt.plot(multi_lowess_result[:, 0], multi_lowess_result[:, 1], 'r-', linewidth=3
)

plt.xlabel('predicted sales'), plt.ylabel('studentized residual')
```

What does the diagnostic plot tell you?

SOLUTION (G)

Use model-2 to predict the value of 'sales' if TV = 350 and radio = 50. Specify 95% confidence interval of the prediction.

EXTRA PROBLEM (ONLY FOR HYPERCURIOUS STUDENTS)

Do you have other ideas to further improve the model?

Create model_3: sales = beta_0 + (beta_1 *TV)* + *(beta_2* TV^2) + (beta_3 *radio)* + *(beta_4* TV * radio).

The hope is that by adding the TV^2 term we will be able to remove the non-linear pattern.

Perform regression and construct the residual plots. Comment on what you see in the residual plots.

Is model_3 better better than model_1 and model_2?

In [ ]:

```
import statsmodels.formula.api as smf
import numpy as np

#Let's create the third model
model_3 = smf.ols('sales ~ TV + np.square(TV) + radio + (TV*radio)', data=dfdata
).fit()

#print summary of regression results for model_3. Fill the blanks below.
print(_____________________________)
print("----- p-values ------")
print(_____________________________)

print("--- Significant Features-------")
print(_____________________________)
```

In [ ]:

```
import matplotlib.pyplot as plt
from statsmodels.nonparametric.smoothers_lowess import lowess

#Perform Diagnostics of the least squares regression fit on model_3
#First compute the studentized residuals. Fill the blank below.
studentized_residuals_model3 = ___________________________________

#Second compute the predicted values of Y from the model. Fill the blank below.
predicted_Y_model3 = ____________________________

#Residual plot: Plot a graph of the 'predicted_Y_model3' and 'studentized_residu
als_model3' as points.
#Label the axes. Fill the balnks below
plt.plot(__________), plt.xlabel(_____________), plt.ylabel(_____________)

#Use the lowess function to draw a trendline in the residual plot.
multi_lowess_result = lowess(_____________________________________)

# plot the trendline
plt.plot(_____________________________________ )
```

In [ ]:

```

```