STAT1005 - Assignment 3

**Instructions:**

* Please complete all questions put all your code and outputs into a single separate Python Notebook.
* Please rename your python notebook file as “{*UID*}-{*Name*}-assignment3.ipynb”, e.g., “3030016546-Chan\_Tai\_Man-assignment3.ipynb”.
* Please upload your ipynb file to the Moodle.
* Write down your explanation using Markdown cells.

**Submission:**

1. In Jupyter Notebook, save your file by clicking **File 🡪 Downloaded as 🡪 Notebook (.ipynb).**

2. Submit your **ipynb file** to Moodle.

**Enquiry:**

If you encounter any problems with this assignment, please feel free to post your questions on Moodle’s **Assignment Discussion Forum or email to TA,** MrCheung Wai Ki,Keith ([u3580369@connect.hku.hk](mailto:u3580369@connect.hku.hk)).

**Note:**

* If you are using CoLab, we recommend you use our initialized one: <https://bit.ly/3FvfxzQ>
* If you are using your local laptop, you can initialize it yourself or use the one on GitHub: <https://bit.ly/3xchoqJ>

**Questions:**

1. We want to use a linear regression model to predict the housing price (column 8) based on six predictors (columns 2 to 7) [5 marks]:

Data is available at “Real\_estate.csv”.

1. Write scripts to fit a linear regression to all data and print the R^2. Based on the results of the fitted model, which predictors do NOT have coefficient significantly far from zero (at a significance level of 0.05)?

Hint: use statsmodels to fit the data. Remember to add constant for intercept.

<https://www.statsmodels.org/dev/generated/statsmodels.regression.linear_model.OLS.html>

<https://www.statsmodels.org/stable/examples/notebooks/generated/ols.html>

1. Write scripts to use the fit model to predict the training data and obtain the predicted y. Calculate the Pearson’s correlation coefficient R between predicted and observed y. What is the difference between the Pearson’s R^2 and the SSE based R^2 calculated in Q1(a)?

Hint: use the “predict()” function in the above statsmodels object.

1. Now, we want to perform 5-fold cross-validation for linear regression and obtain the predicted y as a validation set instead of training set. Calculate the R^2 by Pearson’s correlation coefficient between predicted and observed y.

Hint: Use utility functions in scikit-learn. Please don’t add constant as scikit learn will add it by default.

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import cross\_val\_predict

1. Please visualize the observed y (x-axis) and predicted y (y-axis) from Q1(c) by using function seaborn.regplot().
2. If one predictor can be non-linearly transformed, e.g., by logarithm, which one you would like to transform to improve the prediction. What is the new R^2 between the observed and predicted y, based on 5-fold cross-validation as in Q1(d)?

Hint: Visualize the distribution of the predictor and outcome with

seaborn.pairplot(df\_reg, y\_vars="Y house price of unit area")

1. We want to use logistic regression to predict if a person has stroke or not (column 11) based on eight common diagnostic features (columns 2 to 10). [5 marks]:

Data is available at “stroke\_data.csv”.

1. Write scripts to fit a logistic regression to all instances. What is the log likelihood?

Hint: use statsmodels to fit the data. Remember to add constant for intercept.

https://www.statsmodels.org/dev/generated/statsmodels.discrete.discrete\_model.Logit.html

1. For the fitted model in Q2(a), what are the features with p>0.05? After removing them, what is the new log likelihood?

Hints: Use pandas.drop(feature\_list, axis=1)

1. Now, let’s perform 5-fold cross-validation, what is the accuracy with full features and with the filtered feature set defined in Q2(b).

Hint: Use utility functions in scikit-learn. Please don’t add constant as scikit learn will add it by default.

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import cross\_val\_predict

clf = LogisticRegression(max\_iter=300)

y\_pred\_proba\_cv = cross\_val\_predict(clf, X, y, cv=5, method='predict\_proba')

1. Plot the ROC curves for the filtered feature set used in Q2(c).

Hint: Use utility functions in scikit-learn:

from sklearn.metrics import roc\_curve

1. Based on the ROC curve with the filtered feature set used in Q2(d), if we want to have an FPR around 0.05 (as close as possible), what threshold we should use on the predicted probability? And what TPR we will have?

Hint: check the output from the roc\_curve.