QBUS2820 Predictive Analytics
Semester 2, 2020

Individual Assignment 1

Key information

1. Required submissions:
   a. **ONE** written report (word or pdf format, through Canvas- Assignments- Report Submission (Individual Assignment 1)).
   b. **ONE** Jupyter Notebook .ipynb file (through Canvas- Assignments- Upload Your Program Code Files (Individual Assignment 1)).
2. Due date/time: **Friday 16-October-2020, 2pm** (report and Jupyter notebook submission). The late penalty for the assignment is 5% of the assigned mark per day, starting after 2pm on the due date. The closing date/time **Monday 26-October-2020, 2pm** is the last date/time on which an assessment will be accepted for marking.
3. Weight: **30%** of the final mark.
4. Length: The main text of your report (including Task A and Task B) should have a **maximum of 15 pages**. Especially for Task A, you should write a complete report including sections such as business context, problem formulation, data processing, EDA, and feature engineering, methodology, analysis, conclusions and limitations, etc.
5. If you wish to include additional material, you can do so by creating an appendix. There is no page limit for the appendix. Keep in mind that making good use of your audience's time is an essential business skill. Every sentence, table and figure have to count. Extraneous and/or wrong material will reduce your mark no matter the quality of the assignment.
6. **Anonymous marking**: As the anonymous marking policy of the University, please only include your student ID in the submitted report, and do **NOT** include your name. The file name of your report should follow the following format. Replace "SID" with your Student ID. Example: **SIDQbus2820Assignment1S22020**.
7. Presentation of the assignment is part of the assignment. **Markers will assign 5 marks for clarity of writing and presentation**. Numbers with decimals should be reported to the **four-decimal point.**

Key rules:

- Carefully read the requirements for each part of the assignment.
- Please follow any further instructions announced on Canvas.
- You must use Python for the assignment. Use "*random_state= 1*" when needed, e.g. when using "*train_test_split*" function of Python. For all other parameters that are not specified in the questions, use the default values of corresponding Python functions.
- Reproducibility is fundamental in data analysis, so that you will be required to submit a Jupyter Notebook that generates your results. Not submitting your code will lead to **a loss of 50%** of the assignment marks.

- Failure to read information and follow instructions may lead to a loss of marks. Furthermore, note that it is your responsibility to be informed of the University of Sydney and Business School rules and guidelines, and follow them.

- Referencing: Harvard Referencing System. (You may find the details at: http://libguides.library.usyd.edu.au/c.php?g=508212&p=3476130)

Task A. Moneyball (35 Marks)

You will work on the **NBA salary dataset**.

Note: This task does not require prior knowledge of basketball. You should not add any personal subjective assumptions about the data based on your existing knowledge. This can lead to inaccurate results. You should use the techniques that we learnt and you discovered to complete the prediction task.

1. Problem description

Select **three models** to predict NBA player salary from performance statistics.

The selected three models need to follow the below rules:

- one model is a linear regression model;
- one model is a kNN regression model;
- one model is a model that we have not covered in the QBUS2820 unit so far. You need to self-explore and self-study this model, since the ability of self-study is critical in the field of machine learning which is evolving quickly.

As a consultant working for a sports analytics company, the NBA league approached you to develop predictive models to predict NBA salaries based on machine learning techniques. To enable this task, you were provided with a dataset containing highly detailed performance of the NBA players.

As part of the contract, you need to **write a report** according to the details below.

2. Understanding the data

You can download the "*NBA_Train.csv*" and "*NBA_test.csv*" data for the Canvas. The response is the **SALARY($Millions**) column in the dataset.

NBA glossary link below and the glossary Table at the end of the Task A can help you understand the meaning of the variables better:
https://stats.nba.com/help/glossary

You should use the given test set to evaluate the performance of your work. The performance/scoring metric is: **Root Mean Squared Error (RMSE), for the test set.**

Your target the of the test set RMSE is: **less than 4.1 ($Millions).**

Please note **at least two of your three presented models** must achieve the above target.

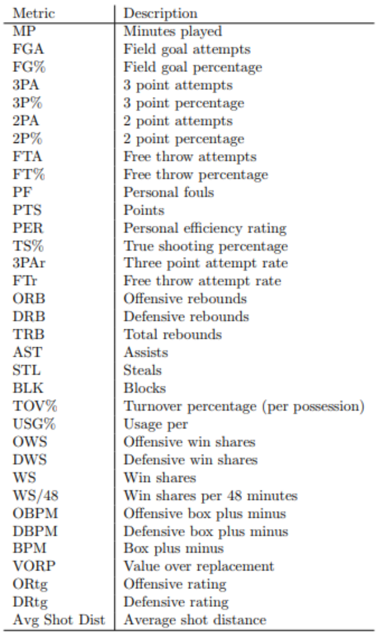
3. Written report

The purpose of the report is to describe, explain, and justify your solution to the client with polished presentation. Be concise and objective. Find ways to say more with less. When it doubts, put it in the appendix. You can refer to the file "**TaskA_instructions**" on more detailed instructions on how to work on Task A including writing the report.

Suggested outline:

1. **(2 marks)** Introduction: write a few paragraphs stating the business problem and summarising your final solution and results. Use plain English and avoid technical language as much as possible in this section (it should be for a wide audience).
2. **(2 marks)** Data processing and exploratory data analysis: provide key information about the data, discuss potential issues, and highlight interesting facts that are useful for the rest of your analysis.
3. **(3 marks)** Feature engineering.
4. **(14 marks)** Methodology of the linear regression and kNN regression models: present your selected models, your rationale, how you fit them, model selection process, some interpretations, etc.
5. **(7 marks)** Methodology of the model that is not covered in our unit: why you choose this model, briefly present how the model works, explain how the model is different to linear regression and kNN regression, how you fit the model, model selection process, some interpretations, etc.
6. **(5 marks)** Report and interpret the test set performance that meets the required criteria.
7. **(2 marks)** Final analysis, conclusion, limitations and future work suggestions.

The NBA glossary Table.



Metric	Description
MP	Minutes played
FGA	Field goal attempts
FG%	Field goal percentage
3PA	3 point attempts
3P%	3 point percentage
2PA	2 point attempts
2P%	2 point percentage
FTA	Free throw attempts
FT%	Free throw percentage
PF	Personal fouls
PTS	Points
PER	Personal efficiency rating
TS%	True shooting percentage
3PAr	Three point attempt rate
FTr	Free throw attempt rate
ORB	Offensive rebounds
DRB	Defensive rebounds
TRB	Total rebounds
AST	Assists
STL	Steals
BLK	Blocks
TOV%	Turnover percentage (per possession)
USG%	Usage per
OWS	Offensive win shares
DWS	Defensive win shares
WS	Win shares
WS/48	Win shares per 48 minutes
OBPM	Offensive box plus minus
DBPM	Defensive box plus minus
BPM	Box plus minus
VORP	Value over replacement
ORtg	Offensive rating
DRtg	Defensive rating
Avg Shot Dist	Average shot distance

Sports Reference LLC, 2016a.

Task B. Gradient Ascent (25 Marks)

Instructions

You will work on the **Boston housing dataset**, to predict the median value of owner-occupied homes in $1000's as the target $\boldsymbol{y}$ (column "*medv*"), using linear regression.

You can download the data set "*Boston_housing_data.csv*" and the corresponding data description file "*Boston_housing_data_description*" on Canvas.

1. **(5 marks)** You decide to use this dataset for a linear regression task, that is to use the existing information to predict the "*medv*". But before that, you want to first conduct exploratory data analysis (EDA).

   After EDA, please choose **three most relevant features** and present your results including:
   - How your EDA is conducted;
   - Carefully explain your feature selection rules and criteria;
   - There is no need to train a regression model for this question.

2. **(20 marks)** Based on slides from 27 to 32 of week 5 slides, write **your own** Gradient Ascent algorithm, to estimate the parameters of the given regression problem **with the intercept**.

Use three features: "*rm*", "*dis*" and "*tax*", and $\boldsymbol{\beta} = [0, \dots, 0]^T$ as your initialization point. You may write a python function named such as `Gradient_Ascent_Algo`, with various inputs, e.g. data matrix $\boldsymbol{X}$, target $\boldsymbol{y}$, an initial parameter vector $\boldsymbol{\beta}^{(0)} = [0, \dots, 0]^T$, learning rate, the number of GD iterations T, stopping criteria and other arguments you see appropriate.

You need to:

- find the optimal learning rate for your own Gradient Ascent algorithm and explain why this is the optimal value;
- use **leave one out cross validation** to select the optimal learning rate;
- **design and justify** your approach.

You can use the "Reference code 1" to load and standardize your data and use the "Reference code 2" as a template to start your Gradient Ascent algorithm.

Reference code 1:

```
import pandas as pd
import numpy as np
import matplotlib.pylab as plt

data = pd.read_csv('Boston_housing_data.csv')
X = data[['rm','dis','tax']]
y = np.array(data[["medv"]])
```

```
# Standardization
X = (X-X.mean(axis=0))/X.std(axis=0)
y = (y-y.mean(axis=0))/y.std(axis=0)

# Adding constant
X = np.column_stack((np.ones(len(X)), X))
```

Reference code 2:

```
"""
Build the gradient ascent function for MLE
"""
# m denotes the number of training examples here, not the number of features
def Gradient_Aescent_Algo(X, y, beta, alpha, numIterations):
    # sample size
    N = len(X)
    XTrans = X.transpose()
    # create a vector to save all the likelihood values at each iteration
    likelihood_values = np.zeros((numIterations,1))
    beta_values = np.zeros((numIterations,2))

    for i in range(0, numIterations):
        # predicted values from the model
        f_X = np.dot(X, beta)

        # calculte the likelihood

        # save all the likelihood values at each iteration

        # calcualte the gradient using matrix representation

        # update the parameters simulteneously with learning rate alpha
        beta = beta + alpha * gradient
        # save all the estimated parametes at each step
        beta_values[i,:]= beta.transpose()
    return beta, likelihood_values, beta_values
```