

**Postgraduate Certificate in Cloud Native Computing**

**Postgraduate Certificate in Software Design with Artificial Intelligence**

**Applied Scripting Languages**

**Assignment 1**

Student ID: A00267948

Student Name: Daniel Foth

*Brief Description:*

*This assignment aims to design, implement and test a Python program to analyse and visualize a csv data set about information on bike sharing. In this document, the data, testing, visualisation and reflective learning will be described in detail.*

Contents

[Introduction 3](#_Toc22211610)

[Data 4](#_Toc22211611)

[Design 5](#_Toc22211612)

[Testing 6](#_Toc22211613)

[Analysis and Visualisation 7](#_Toc22211614)

[Conclusion 8](#_Toc22211615)

[Appendix 1: Reflective Learning Log 9](#_Toc22211616)

[Appendix 2: References 10](#_Toc22211617)

# Introduction

*This assignment aims to design, implement and test a Python program to analyse and visualize a csv data set about information on bike sharing. In the Data section below you can find information about the data, how its laid out and where it was sourced from.*

*The design for the program should be modular and allow for the importing of custom data sets to be processed via user input. It should be able to output graphs for linear regression and normal distribution. More information can be found in the design section below.*

*There are multiple ways to test code. As part of this assignment manual, unit and user testing shall be completed and described in the testing section below.*

*The analysis and visualisation of the data will help evaluate how successfully the data was analysed as part of the conclusion below.*

*Lastly a reflective log will be available detailing how the code was written and what issues were overcome.*

# Data

The data set was taken from <https://archive.ics.uci.edu/ml/datasets/bike+sharing+dataset>

This data set was composed by Hadi Fanaee-T and records the number of users interacting with a bike sharing system. It can give us an interesting view of how the bikes are utilised in different weather conditions. It also shows us how many registered/casual users use the services.

The data set includes 2 csv files. hours.csv and days.csv. hours.csv has 17 columns and 17389 entries. This will be the default dataset used for the analysis. days.csv has 16 columns and 731 entries.

The following columns are present:

|  |  |
| --- | --- |
| Column | Description |
| instant | Index of the record |
| dteday | The date of the record |
| season | The season the record was taken in. (1=Springer, 2=Summer, 3=Fall, 4=Winter) |
| Yr | The year of the record. (0=2011, 1=2012) |
| mnth | The month of the record. (from 1 to 12) |
| hr | (not in days.csv) The hour of the record. (from 1 to 12) |
| holiday | Defines if a day is a holiday. (0=false, 1=true) |
| weekday | Defines if a day is a weekday. (0=false, 1=true) |
| workingday | Defines if a day is a working day meaning it’s a week day but not a holiday. (0=false, 1=true) |
| weathersit | This defines what the weather was like at the time of taking the record. (value from 1-4)   1. Clear, Few clouds, Partly cloudy, Partly cloudy 2. Mist & Cloudy, Mist & Broken clouds, Mist & Few clouds, Mist 3. Light Snow, Light Rain & Thunderstorm & Scattered clouds, Light Rain & Scattered clouds 4. Heavy Rain & Ice Pallets & Thunderstorm & Mist, Snow & Fog |
| temp | Normalized temperature in Celsius. (max is 41) |
| atemp | Normalized feeling temperature in Celsius. (max is 50) |
| hum | Normalized humidity. (max is 100) |
| windspeed | Normalized wind speed. (max is 67) |
| casual | Number of causal users per record |
| registered | Number of registered users per record |
| cnt | Total number of users per record |

To compose this data Hadi Fanaee-T used 3 sources:

1. Original Source: <http://capitalbikeshare.com/system-data>
2. Weather Information: <http://www.freemeteo.com>
3. Holiday Schedule: <http://dchr.dc.gov/page/holiday-schedule>

# Design

<https://github.com/DanielsHappyWorks/ASL-CA1-Data-Science-from-Scratch>

Modular

Utilities

Chart?

# Testing

Python version

Code Testing - dir

Manual Tetsing

User Testing

# Analysis and Visualisation

Data output/input in cli

LR

Distribution

# Conclusion

Interesting outputs

Data set better suited for poly

# Appendix 1: Reflective Learning Log

*Date:*

*Work Completed*

*Understanding Achieved:*

List:

* Commits on Oct 16, 2019
  + Added code to turn a csv file into a dictionary
* Commits on Oct 24, 2019
  + updated scaling for mobile to fix formatting issues
* Commits on Oct 26, 2019
  + added DataFrame class
  + Added support for loading in custom CSVs and printing them to cli
  + added menu for printing and processing data
  + Added TO DO statements, expanded menus to list all functions of program, added placeholder functions for the missing functionality.
* Commits on Oct 29, 2019
  + Added support for user specified column typing (int/float/string), added util fo check user inputs are as expected and added comments to functions inDataFrame class
  + Implemented detailed header printing
  + updated to dos
  + Added exception handling for string to integer conversions, Updated TextUtils with comments and more generic code
* Commits on Oct 30, 2019
  + Added 1st iteration of the linear regression algorithm with the abili ty to plot the graph
  + moved estimation functions to MathUtil and added support for export of graph, linear regression algorithm needs more testing
  + Fixed LR algorinth, added more files for export and changed directory to always optuput to ./output/date, tested dataset and found to interesting correlation (need to find a new one?)
* Commits on Nov 5, 2019
  + Updates based on user testing: More validation for inputs, changed gr aph colour scheme to make it more redable, fixed issue with graphs not clearing, fixed issue with datasets breaking on reload, also refactored duplicated code in main for maintainability
  + Updated comments in all files
  + Decreased code duplication and added ability to export all graph data at once, fixed some minor issues with selections too
  + Added graceful handling of FileIO for opening/writing files
  + Created constants file for different strings in the program, fixed issue with exporting all with default data types, organised directories and added custom delimeter support
  + Moved Utils into Utils directory
* Commits on Nov 8, 2019
  + Added most tests for ExceptionUtils
* Commits on Nov 11, 2019
  + updated directory structure and exceptions tests
  + Fixed paths to work with Python 3.7.4 (default, Aug 9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)] on win32
  + Added tests for the maths util, while testing figured out that slope and intercept formulas were the wrong way around, but still used correctly for calculating predictions
  + Added tests for text utils, fixed file names with underscores so pytest can now recognise them
  + file name change
  + Added tests for the data frame class, fixed issue where the invalid rows werent dropped correctly
  + Fixed some directory names
* Commits on Nov 14, 2019
  + Added some support for exporting normal distributions
  + Made adjustments to distributions

# Appendix 2: References

1. Fanaee-T, H. (2019). *UCI Machine Learning Repository: Bike Sharing Dataset Data Set*. [online] Archive.ics.uci.edu. Available at: https://archive.ics.uci.edu/ml/datasets/bike+sharing+dataset [Accessed 18 Nov. 2019].
2. GeeksforGeeks. (2019). *Linear Regression (Python Implementation) - GeeksforGeeks*. [online] Available at: https://www.geeksforgeeks.org/linear-regression-python-implementation/ [Accessed 18 Nov. 2019].
3. En.wikipedia.org. (2019). *Simple linear regression*. [online] Available at: https://en.wikipedia.org/wiki/Simple\_linear\_regression [Accessed 18 Nov. 2019].
4. Mathsisfun.com. (2019). *Normal Distribution*. [online] Available at: https://www.mathsisfun.com/data/standard-normal-distribution.html [Accessed 18 Nov. 2019].