Titanic Data analysis using Pandas, Numpy, Tensorflow

The Titanic dataset and the related story on how the boat sank on his first trip from England to New York is well-known. The practical dataset is often used in Data Science, Big Data, and Machine Learning to illustrate how to clean the data, do an initial exploratory data analysis, and ultimately do some predictive analysis on who survived the trip.

A brief search on google for Titanic, Kaggle competition.. Should lead to resources to get familiar to the dataset, solutions, etc..

Top submissions on Kaggle predicted with accuracy between 83 and 90%. Prediction with 80% accuracy are easy, while accuracy close to 90% requires fine-tuning.

For this programming assignment, you should do the following:

1. Use either the Google Colab environment or install relevant libraries such as numpy, pandas, matplotlib, seaborn, and tensorflow within a Notebook Python environment such as Anaconda.

When running the version check the Tensorflow version should be 2.0 or above ..for the other libraries such as pandas, matplotlib, seaborn, numpy.. etc

It is generally better to use the latest stable version such as python 3.8+ ...{i.e. close to the latest but supported as latest built {/ e.g. Python 3.9} experimental versions such as 3.10a1 (2021-05-03)/} often has some bugs or incompatibilities as you may have experienced with Spark}

The current version of Tensorflow is 2.4.1 and includes Keras :

https://www.tensorflow.org/api_docs/python/tf

{note: if you use your own and do not have a relevant/approved GPU such as NVidia certified models, do not install TensorFlow with GPU ... this would crash everything}

i.e. easier to run on Google Colab https://www.youtube.com/watch?v=PitcORQSjNM

... to check ...run ...

import tensorflow as tf

tf.__version__

2. Load the data for Titanic {either load the full data and then perform a train_test_split or load a fraction of data for training, and fraction for testing.

Data is uploaded to blackboard already for download.. or raw data can be downloaded from the web.

3. Do some exploratory analysis of the data with:

... info()

...describe()

.. until you have a feel for the data

And a few graphs outlining the nature of the attributes in the dataset.

4. Do some data cleaning to eliminate redundant columns if necessary {they would highly extend computing time if included in latest models}

5. Try a few different models {e.g. Logistics, Random Forest, Decision tree, Classification} to analyze the data and run them. Which one would have best predictions [on training data? On test data?] - ? How long does it take to run e.g. %%timeit

6. Do a few visualizations with the finalized data {i.e. after deleting extra attributes, and treating missing values.

6. Would you {and an assumed passenger e.g Woman / 30 year old} survived the trip? With your or existing models {E.g. look at the Jack and Rose notebook case. Jack actually died but Rose actually lived ...

Most models gave 15% chance to survive for Jack and 97% for Rose }

jack = pd.Series([0, 'Jack', 3, 'male', 23, 1, 0, 5.0000, 'S'])

rose = pd.Series([1, 'Rose', 1, 'female', 20, 1, 0, 100.0000, 'S'])

0	0	Jack	3	male	23.0	1	0	5.000	S	0.150512
1	1	Rose	1	female	20.0	1	0	100.000	S	0.970346

7. Short write-up [1 page] of you findings in layman terms... who survives .. who dies

Which combination of attributes is best?

- Young or old people
- Man or Female
- Boarding at one of the three embarquement/boarding places
- Rich or poor {correlates to decks}