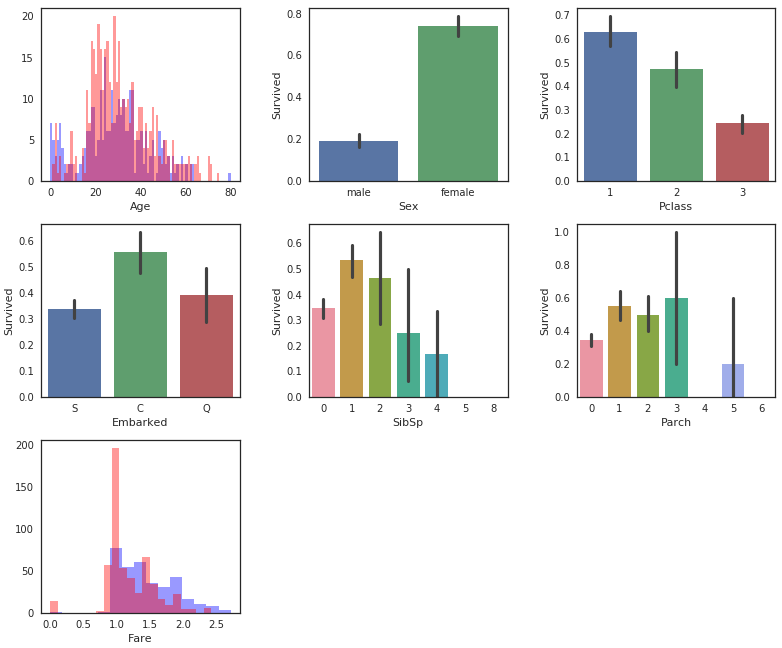
Data Science Project Structure

Exploratory Data Analysis

1. Loading Data and Modules – Combining information if necessary from several data sources.
   * Importing Pandas, Numpy, Seaborn, among others.
   * Renaming collumns if necessary
2. Initial Exploration – Looking at the data is as many different ways as possible because some connections will seem to be obvious, but others will require a more in debt examination
   * Head, Shape, Describe, Describe(include=[‘object’]), Isnull().sum(),info()
   * Analysis of the target variable
   * Relations between target variable and other variables in the data
   * Creation of new variables
   * Relations between features and target variable. From the first graphs we will be able to better combine the features to study the phenomena under investigation.
     + Use Correlations and Pairplots if possible (numerical variables)
     + Examination in details with barplot, factorplot, violinplot, distplot, countplot, boxplot and kdeplot justified with crosstabs, groupby, value\_counts()/value\_counts(normalize=True) functions and pivot\_table

Data Cleaning

1. Filling Missing values – Mean imputation, drop the variables or the rows, input a string “Unknown” as a variable to consider
2. Variables transformation
   1. Continues variables such as Age into Interval Variables (data.loc or data.iloc)
   2. Categorical Variables into numerical variables (replace function)
   3. Whenever a variable contains several distinct values (unique()), pick the relevant ones and consider less relevant as others
   4. Drop irrelevant variables
   5. Creation of dummy variables if necessary
3. Check once more the relations using visual exploration

Prepare for Modelling

1. Check correlation between features (heatmap)
2. Diagnose Multicollinearity

Modelling

1. Train and Test datasets
2. Run several different classifiers such as Statistical Models (Linear Models), Machine Learning Models and Ensemble models
   1. For K Nearest Neighbors obtain the best number of neighbors. Use a for loop and compare train and test datasets
   2. For Decision Trees and Random Forest obtain the best number of max\_leaf\_nodes. Use a for loop and compare train and test datasets.
3. Hyper-Parameter Tuning for the better models (GridSearch)
4. Confusion Matrix for the better models using

Validation

1. Ranking the models
2. Ranking the features where it’s possible
3. Cross Validation, aiming to access overfitting/under fitting.
   1. Use KFold, cross\_val\_score to get accuracies mean and standard deviation.
4. Confusion Matrix for the better models using