Preview train data

PassengerId Survived Pclass ... Fare Cabin Embarked

0 1 0 3 ... 7.2500 NaN S

1 2 1 1 ... 71.2833 C85 C

2 3 1 3 ... 7.9250 NaN S

3 4 1 1 ... 53.1000 C123 S

4 5 0 3 ... 8.0500 NaN S

The number of samples into the train data is 891.

Preview test data

PassengerId Pclass ... Cabin Embarked

0 892 3 ... NaN Q

1 893 3 ... NaN S

2 894 2 ... NaN Q

3 895 3 ... NaN S

4 896 3 ... NaN S

The number of samples into the test data is 418.

"""Checking if there are missing values"""

Checking if there are missing values in train data.

PassengerId 0

Survived 0

Pclass 0

Name 0

Sex 0

Age 177

SibSp 0

Parch 0

Ticket 0

Fare 0

Cabin 687

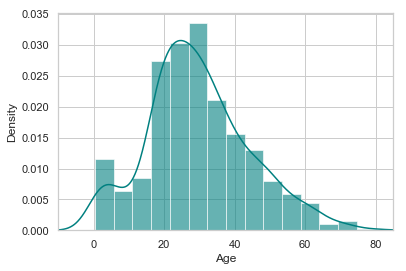
Embarked 2

We can see here that 177 passengers are missing an age value, 687 are missing a Cabin value and 2 are missing a Embark value.

"""Age as a missing Value"""

Next, I check the percent of missing "Age" value. 19.87% of the passengers are missing the "Age" value. Next, I will take a look at the mean and median as well as a histogram of the "Age" variable.

"""Histogram"""



We can see here that "Age" is right skewed. Filling in the missing values using the mean might give us biased results that are older than desired. The mean of "Age" is 29.70 years old. To deal with this, we'll consider using the median to input the missing values. The median of "Age" is 28.00 years old. The median is lower than the mean and "Age" is right skewed, we will be using the median when filling in the missing "Age" values.

“””Missing Cabin Values"""

Next, we will investigate the percent of missing "Cabin" values

The Percent of missing cabin records is 77.10%. Since we are missing 77% of these values inputting a value for prediction is a bad idea, so will ignore this variable in our model.

"""EMBARKED MISSING VALUES"""

Let’s consider the percent of missing "Embarked" values. The percent missing "Embarked" records is 0.22%. There are only 2 port missing values. In this case I would suggest using the port from which most passengers embarked from to fill in the missing values.

Now let’s find out how many passengers embarked from each of the three ports. C = Cherbourg, Q = Queenstown, S = Southampton.

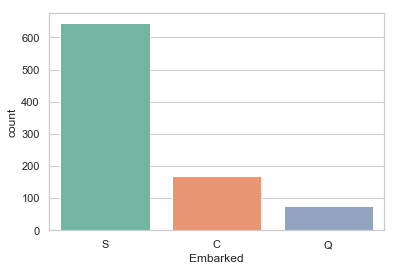
Passengers grouped by port of embarkment (C = Cherbourg, Q = Queenstown, S = Southampton):

S 644

C 168

Q 77

Let’s create a histogram to visually represent the number of passengers which embarked from each port.

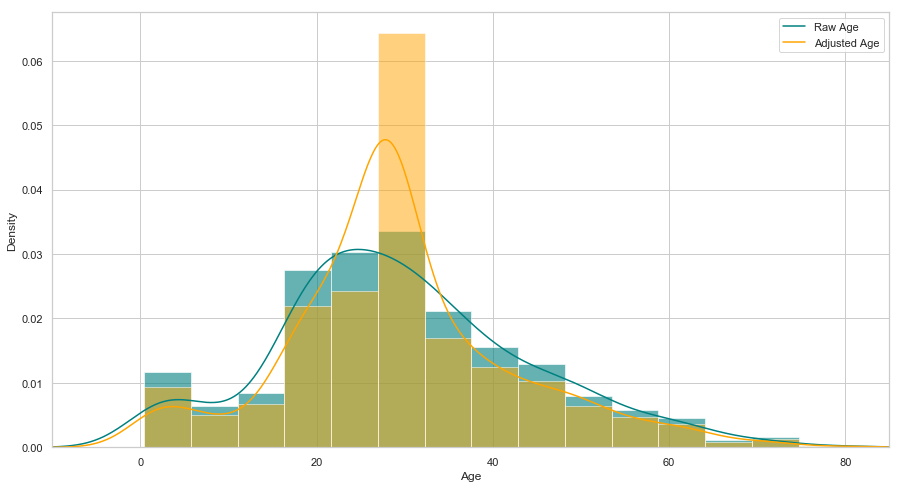


This is showing the distribution of where the passengers embarked from. Most passengers embarked from Southampton so we will input the 2 missing values with Southampton.

"""Assessment of missing values in training data"""

If "Age" is missing for a given row, I'll input the median age which is "28". If "Embarked" is missing for a riven row, I'll input "S" which is the most common boarding port. I'll ignore "Cabin" as a variable as a large percent was missing. As a note the "Cabin" variable appears to be associated with the passenger's class and fare paid.

Now I will graph the Age variable again as a histogram, however, this time I will be comparing the values before the train data set was adjusted to the train data set after it was adjusted.



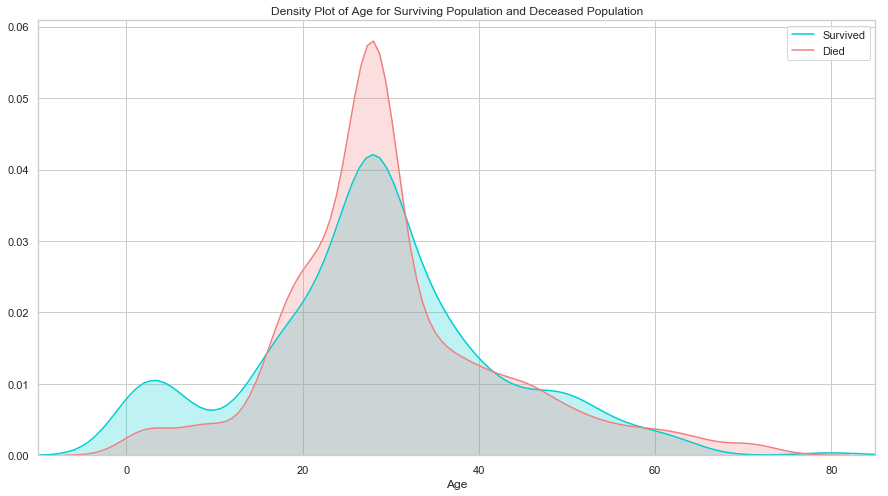
The adjusted "age" is not skewed right anymore it is much more "normal" as seen by the yellow graph.

"""Handeling additional variables"""

The SibSp and Parch variables relate to traveling with family, to handle multicollinearity I will combine these two variables into one categorical predictor, whether or not an individual was traveling alone. After creating these categorical variables, I will apply the same process to the test data set so that the missing values for that set can be added.

"""DATA ANALYSIS"""

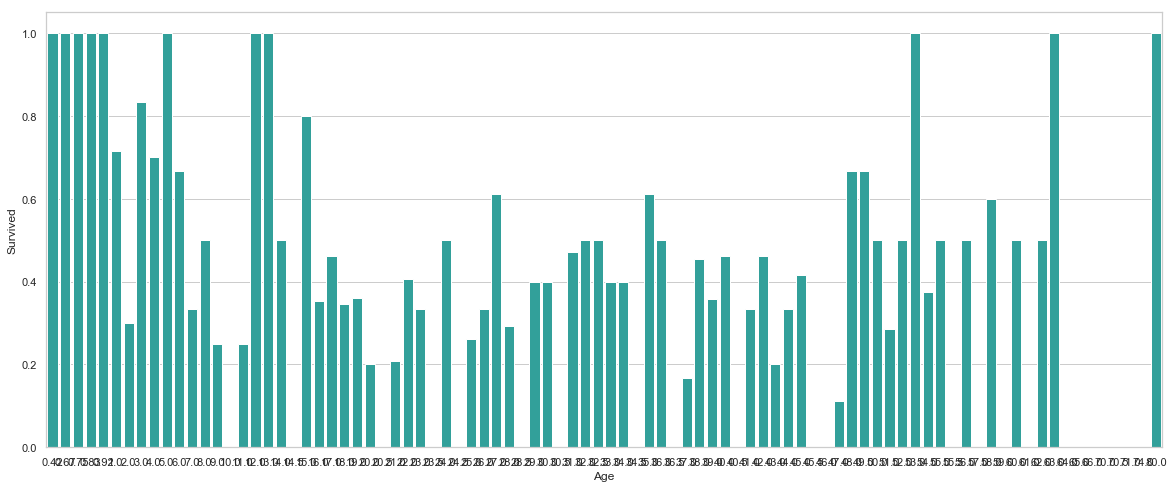
"""Looking into AGE"""

"""Creating a Histogram of that compares the survivors and non survivors ages""" 

The Age distribution of survivors is very similar to that of who died, the biggest difference is that a great number of children survived, this shows that passengers prioritized saving children. However, it should also be noted that the "Age" 28 was used to fill in many of the missing age values, therefore that will have an impact on the number of deaths/survivors. Also, the 28-age range had the largest number of members in it so it makes sense that the largest number of survivors and deaths come from this age range.

""Creating a bar chart of age of survivors"""

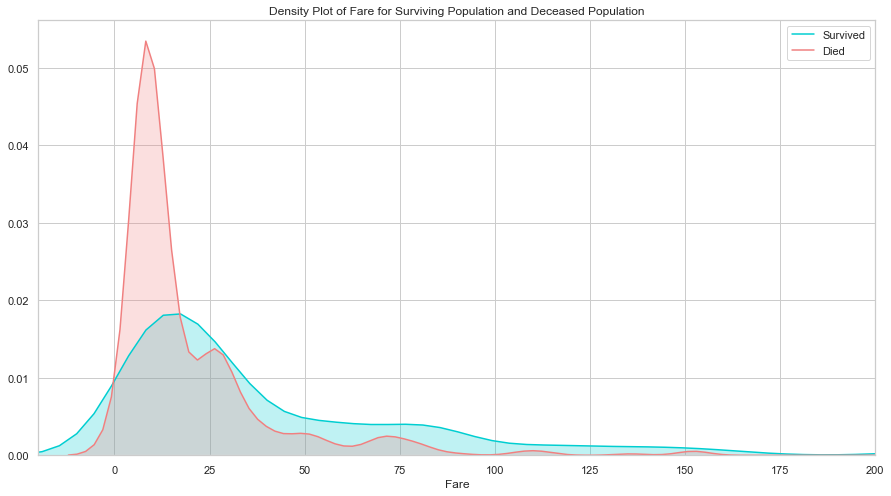
Now we will create a bar chart to represent the survivors



The number of younger survivors is an interesting data point to consider, so I am going to create another variable called minors which will include passengers under 16.

"""Considering FARE"""

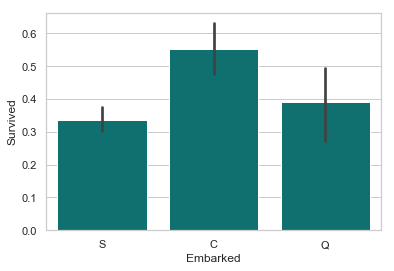
Now let’s consider the FARE price and how it can relate to the number of survivors.



This shows that passengers that had a lower fair were significantly more likely to die. This can also possibly be related to class as well.

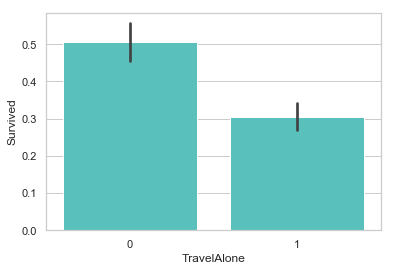
"""Passenger Class"""

We will now create a bar chart for the class of the passengers.



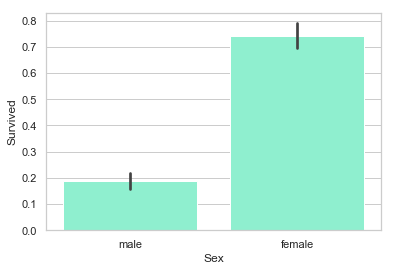
Passengers who boarded in Cherbourg, France, appear to have the highest survival rate. Passengers who boarded in Southampton were marginally less likely to survive than those who boarded in Queenstown. This is probably related to passenger class, or maybe even the order of room assignments (e.g. maybe earlier passengers were more likely to have rooms closer to deck). It's also worth noting the size of the whiskers in these plots. Because the number of passengers who boarded at Southampton was highest, the confidence around the survival rate is the highest. The whisker of the Queenstown plot includes the Southampton average, as well as the lower bound of its whisker. It's possible that Queenstown passengers were equally, or even more, ill-fated than their Southampton counterparts.

"""Traveling alone or with family"""



Individuals traveling alone without family were more likely to die. Considering the time period those traveling alone are most likely Male.

"""Gender"""



A much greater number of females survived than males.

There is a Section after this.

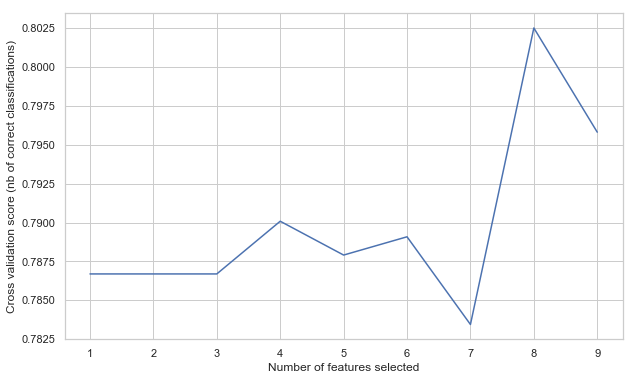
NOTE: This section makes more sense if viewed in the python code as you will have the functions and code to look at on top of the comments.

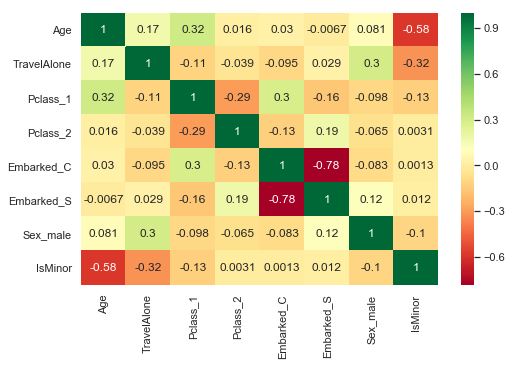
"""Logistical Regression"""

This following section has to do with Logistical regression and training the model. At the end of the section I will use the trained model to determine who within the “test” set will survive the sinking of the titanic.

After building a Logistical Regression the Selected features: ['Age', 'TravelAlone', 'Pclass\_1', 'Pclass\_2', 'Embarked\_C', 'Embarked\_S', 'Sex\_male', 'IsMinor'].

Now plot the number of features vs. the cross-validation scores.





We are keeping 8 features.

"""Evaluating the model. Model evaluation based on simple train/test split using train\_test\_split() function"""

First, I create X (features) and y (response) and use train/test split with different random state values. I can change the random state values that changes the accuracy scores.

Then I check the classification scores of the logistic regression.

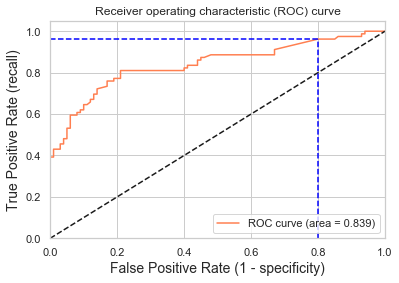
Train/Test split results:

LogisticRegression accuracy is 0.782

LogisticRegression log\_loss is 0.504

LogisticRegression auc is 0.839

Set index of the first threshold for which the sensibility > 0.95



Using a threshold of 0.071 guarantees a sensitivity of 0.962 and a specificity of 0.200, i.e. a false positive rate of 80.00%.

"""Model evaluation based on K-fold cross-validation using cross\_val\_score() function"""

Using cross\_val\_score function

We are passing the entirety of X and y, not X\_train or y\_train, it takes care of splitting the data

cv=10 for 10 folds

scoring = {'accuracy', 'neg\_log\_loss', 'roc\_auc'} for evaluation metric - although they are many

K-fold cross-validation results:

LogisticRegression average accuracy is 0.802

LogisticRegression average log\_loss is 0.454

LogisticRegression average auc is 0.850

"""Model evaluation based on K-fold cross-validation using cross\_validate() function with same parameters as previous K-fold validation function"""

K-fold cross-validation results:

LogisticRegression average accuracy: 0.802 (+/-0.025)

LogisticRegression average log\_loss: 0.454 (+/-0.034)

LogisticRegression average auc: 0.850 (+/-0.025)

Note Fare was left out as a feature, let’s see what happens when it is considered.

K-fold cross-validation results:

LogisticRegression average accuracy: 0.796 (+/-0.028)

LogisticRegression average log\_loss: 0.455 (+/-0.034)

LogisticRegression average auc: 0.849 (+/-0.025)

Observe that the model is slightly deteriorated. The "Fare" variable does not carry any useful information. Its presence is just noise for the logistic regression model.

"""GridSearchCV evaluating using multiple scorers simultaneously"""

best params: LogisticRegression(C=2.8000100000000003, class\_weight=None, dual=False,

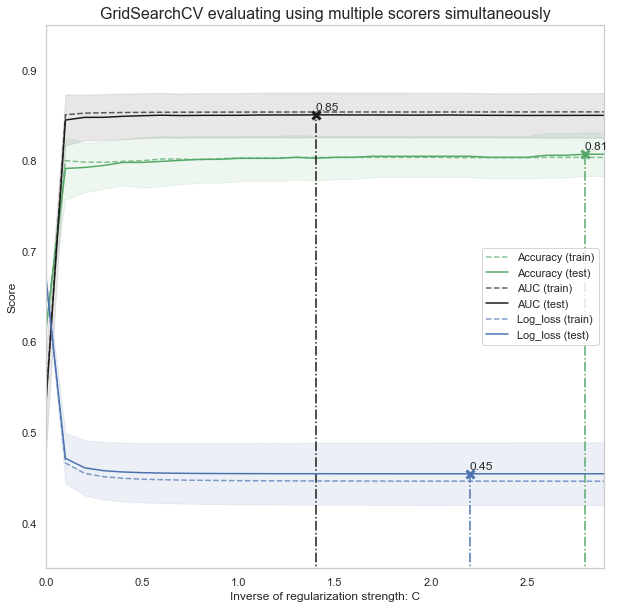
fit\_intercept=True, intercept\_scaling=1, max\_iter=100,

multi\_class='warn', n\_jobs=None, penalty='l2', random\_state=None,

solver='warn', tol=0.0001, verbose=0, warm\_start=False)

best params: {'C': 2.8000100000000003}

best score: 0.8069584736251403



"""GridSearchCV evaluating using multiple scorers, RepeatedStratifiedKFold and pipeline for preprocessing simultaneously"""

====================

best params: Pipeline(memory=None,

steps=[('scale', StandardScaler(copy=True, with\_mean=False, with\_std=False)), ('clf', LogisticRegression(C=5.00001, class\_weight=None, dual=False,

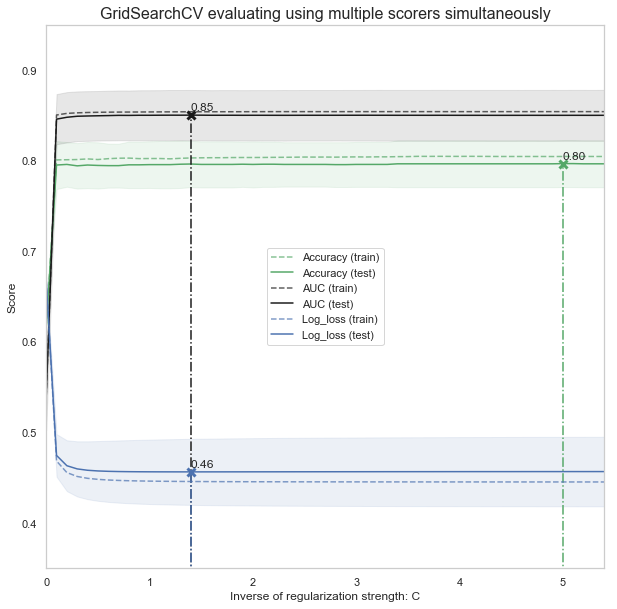
fit\_intercept=True, intercept\_scaling=1, max\_iter=100,

multi\_class='warn', n\_jobs=None, penalty='l2', random\_state=None,

solver='warn', tol=0.0001, verbose=0, warm\_start=False))])

best params: {'clf\_\_C': 5.00001}

best score: 0.7966329966329966.



"""The last step: Using the trained model to predict who in the test data set will survive the sinking of the Titanic """

Here we use the trained model predict who in the test set will survive the sinking of the Titanic. NOTE: if the python code is run it will create a result.csv in the same location as the “train.csv”, “test.csv” and the “Titanic analysis.py”.