Case Study: Titanic dataset

# Context

The Titanic had 2224 passengers and crew on board of which 1502 drowned. The goal is to build a predictive model that analyses data on some members on board and predicts what sort of people could survive. Some important features that could affect survival chances are sex, passenger class, age, etc. This is a binary classification problem where the goal is to predict ‘survived’ or ‘not survived’. The methodologies used in this document are logistic regression and K-nearest neighbors. Theoretical discussions on these methodologies are not a part of the document. However, some nuances will be covered as the modelling proceeds.

# Dataset & Wrangling

The dataset used contains the following features/target variable:

* Passenger Id
* Survived: 0 = No, 1 = Yes
* Pclass: Ticket class 1 = 1st, 2 = 2nd, 3 = 3rd
* Name
* Sex
* Age
* Sibsp: # of siblings/spouses aboard the Titanic
* Parch: # of parents/children aboard the Titanic
* Ticket: Ticket number
* Fare: Passenger fare
* Cabin: Cabin number
* Embarked: Port of Embarkation

Data manipulation and wrangling are the same as described in the model. The data has already been split into training and test data. However, the test data is missing the ‘Survival’ target, so it is difficult to test predictions. Hence initial training and testing are both done on the training dataset. However, once the initial predictions are obtained using XGBoost Classifier that is assumed as the true value for survival in the test dataset. Upon obtaining the supposed truth for the survival values, the test and training data are merged to create one super dataset which is then used for splitting and validation.

Some key highlights for data manipulation:

* Sex was converted to binary
* Title was extracted from the name and converted to ordinal feature and Name was dropped
* Passenger ID was dropped
* Age and Fare were converted to ordinal feature using brackets
* Sibsp and Parch were merged to a new binary feature checking if the person is alone or not
* Ticket number and cabin were dropped
* Embarked was converted from string to ordinal numerical values

# Questions

**Design and execute back-testing of the logistic regression model. Calculate back-testing error metrics to evaluate model performance on “testing data set”.**

The target variable is missing in the test dataset. This target variable for the test dataset is obtained using XGBoost (simply to maintain independence). A super dataset is created using the training and test dataset and then k-fold cross-validation is done as presented in the code alongside the document. The back-testing error metrics are presented in the code as well.

**Calculate the R2. What is an acceptable R2 level for logistic regression models? How would you test the robustness of the model to validate if R2 has not been elevated due to the lack of technical soundness of the model design? What other statistical metrics can you calculate to support the validity of R2?**

R2 is a goodness of fit parameter defined for regression models usually. Logistic regression is a classification problem, hence pseudo R2 can be defined. Multiple definitions exist but for this document, McFadden’s R2 is computed which is essentially the ratio of log-likelihood of the model to the log-likelihood of the null model subtracted from 1.

The value comes out to be XXXXX.

R2 can be artificially inflated if the features are linearly correlated. One needs to assess the VIF and ensure that it is below a threshold (usually 4 or 5). Another reason for R2 to artificially inflate is if the number of features is greater than the sample size of the data.

Other metrics to compute the validity of R2 include accuracy, precision, recall (they come from the confusion matrix) and Area Under Curve (AUC).

**Using the “training data set” perform variable selection using the methodology of your choice. What kind of statistics would you use to evaluate the contribution of each variable to the model? What kind of statistics would you calculate to find out if there is multicollinearity present? If it was present, what are the steps to remediate this issue?**

In the model, the feature selection is done visually and manually by establishing relationships between different features themselves and to the survival rate as well. Some features that do not have a trend or are very sparse are dropped off. For instance, the ‘cabin’ feature is very sparse and cannot be estimated because it is alphanumeric, ticket’ is just another alphanumeric feature that does not have any trend and logically there is no intuition for it to add information about who could have survived. The feature ‘Name’ for instance also does not add any value to the survival chances. There are other statistical techniques such as the Wald Test or the Likelihood test and some embedded techniques like Lasso, but they have not been used in the model.

For Logistic regression looking at the coefficients can determine the contribution of each feature to the result. Other statistics like Wald and p-values test or AIC can also be used to evaluate the contribution of each variable in the model. Finally, the Shapley values derived from game theory provide a good indication of how much each variable contributes to the model. This has been shown in the code.

Variance Inflation factor (VIF) can detect multicollinearity. In this case, the VIF is not very useful because all regressors are either ordinal or categorical.

However, if multicollinearity is present, some ways to address include either eliminating one of the features or performing a PCA to retain the most important variables. Lasso or Ridged regression could also work by adding a penalty term to the loss function.

**Using model validation standards, how would you assess and compare i) the conceptual soundness and ii) the assumptions and limitations of each model (logistic regression vs the model of your choice)? If you had to make a recommendation and could only use one, which model would it be and why?**

The other model of choice is KNN. It is easy to interpret and shows a similar accuracy which is why this model was chosen. As for conceptual soundness, both KNN and Logistic Regression are used to solve classification problems. Both are easily interpretable though KNN does not have an objective function and relies on instance. Both are sensitive to outliers but if the numerical data is converted to ordinal features, then this problem is curtailed. One key difference is that Logistic regression assumes a linear relationship between the features and the log of the outcome while KNN can capture non-linear relationships as well.

*Assumptions:* Logistic Regression assumes none to little multicollinearity among features and assumes a linear relationship between features and the log of the outcome. For KNN it goes by default that the data should be normalized else one feature may significantly impact the outcome.

Limitations: Linear regression cannot handle non-linear relationships and may result in biased prediction if the data is not adequately representative of the real-world scenarios. KNN performs poorly if the feature space is very large or the data is not scaled. Scales poorly with large datasets as distance needs to be calculated for every data point.

Given the understanding of both models, I would prefer Logistic regression over KNN given that all assumptions are met. The reasons include:

1. Logistic Regression has a functional form hence easier to interpret and compute
2. KNN has a few arbitrary hyperparameters, for example the choice of K and the distance metric which is not the case for Logistic Regression
3. KNN scales poorly with large datasets and given the model is in production scalability is an important feature.
4. Logistic regression is less sensitive to outlier data as compared to KNN.

**Imagine that the model of your choice is already in production. Describe the tests you would design to meet the following requirements for ongoing monitoring:**

1. **data inputs must be subject to data controls on an ongoing basis,**
2. **monitor the model performance over time,**
3. **evaluate the robustness, completeness, and accuracy of the tests you performed**

The following need to be taken care of when the Logistic Regression model is already in production:

1. The data source must be authenticated before passing the data to the model. Only authorized endpoint key holders must be allowed to pass data to the model. Check that the data being passed is complete and the data format is as expected. If feature transformations have been done in the model, the same transformations need to be applied to any other data input. Also, significant changes in data should be detected by monitoring statistical measures over time.
2. Track key performance indicators over time. For instance, AUC, R2, etc. Regular recalibration of the model in consultation with business requirements. Deploy a statistical test tracking if a feature is still relevant and independent over time.
3. The robustness of these tests can be tested by passing invalid or noisy data and checking if the tests detect them. For completeness, ensure that all data errors and feature importance are logged regularly. For accuracy, model performance and impact should also be logged.

**Describe how you would evaluate the data quality checks performed by the modeler before any of these models were created. What other tests would you perform to ensure the model development data is appropriate, complete, and accurate?**

Some tests to verify that the modeler has done data quality checks include checking for missing values if they still exist or how they were treated, checking for consistency in scaling of features or the logic behind generating new features, checking if the target variable has a sampling bias and finally checking the source of the data and the table the data has been queried from. Also, check SQL queries used by the modeler to aggregate/generate model data.

Finally, check for multicollinearity using VIF and if the feature engineering includes the requirements of the business.

**Based on your evaluation of the two models and modeling process (logistic regression and the model of your choice) are there any findings? If yes, what are they and how would you propose to remediate them? (Hint: Think in terms of model risk).**

Based on the model presented in the code some findings and their remediation can be stated:

1. Age and class have been multiplied but no explanation has been provided as to why this was needed. *Suggestion:* Check if this feature is important using Recursive feature elimination or maybe a p-test in case of logistic regression. Use the statsmodels library for the same.
2. The title feature may be highly correlated with the Sex feature or may just be a branch of the Sex feature. *Suggestion:* Check model performance in the absence of the title feature.
3. Test data does not have an actual target variable. *Suggestion:* Check if the target variable is available in the original data source. If not, use a third source to obtain target variable and use it as the ground truth.
4. Validation and testing has been done on the same dataset. *Suggestion:* Perform a test-train split and do k-fold cross-validation.
5. For KNN hyperparameter tuning has not been performed on the choice of distance metric and the value of K. *Suggestion:* Perform a hyperparameter tuning and use cross-validation or grid search to select optimum value of K.