**Tabular Playground Series**

**Group name: CS636Project2**

Team members:

* **Rutvij Hiteshbhai Mavani**
* **Ayush Vashistha**
* **Bhirahaspathi Sairam Seeni**
* **Varun Jagannathan**

Graphical user interface, text, application

Description automatically generated

**Introduction:**

The data has been split into two groups:

* training set (train.csv)
* test set (test.csv)

The training set should be used to build your machine learning models. For the training set, we provide the outcome (also known as the “ground truth”) for each passenger. Your model will be based on “features” like passengers’ gender and class. You can also use feature engineering to create new features.

The test set should be used to see how well your model performs on unseen data. For the test set, we do not provide the ground truth for each passenger. It is your job to predict these outcomes. For each passenger in the test set, use the model you trained to predict whether or not they survived the sinking of the Synthanic.

**Abstract:**

The dataset is used for this competition is synthetic but based on a real dataset (in this case, the actual Titanic data and generated using a CTGAN. The statistical properties of this dataset are very similar to the original Titanic dataset. The task is to predict whether or not a passenger survived the sinking of the Synthanic (a synthetic, much larger dataset based on the actual Titanic dataset). For each PasengerId row in the test set, you must predict a 0 or 1 value for the Survived target.

**Classifcation Technique: XGBoost**

It is an implementation of gradient boosting machines created by [Tianqi Chen](https://homes.cs.washington.edu/~tqchen/), now with contributions from many developers. It belongs to a broader collection of tools under the umbrella of the Distributed Machine Learning Community or [DMLC](https://github.com/dmlc) who are also the creators of the popular [mxnet deep learning library](https://github.com/dmlc/mxnet).

The library is laser focused on computational speed and model performance, as such there are few frills. Nevertheless, it does offer a number of advanced features.

The implementation of the model supports the features of the scikit-learn and R implementations, with new additions like regularization. Three main forms of gradient boosting are supported:

* **Gradient Boosting** algorithm also called gradient boosting machine including the learning rate.
* **Stochastic Gradient Boosting** with sub-sampling at the row, column and column per split levels.
* **Regularized Gradient Boosting** with both L1 and L2 regularization.

The implementation of the algorithm was engineered for efficiency of compute time and memory resources. A design goal was to make the best use of available resources to train the model. Some key algorithm implementation features include:

* **Sparse Aware** implementation with automatic handling of missing data values.
* **Block Structure** to support the parallelization of tree construction.
* **Continued Training** so that you can further boost an already fitted model on new data.

XGBoost is free open source software available for use under the permissive Apache-2 license.

The two reasons to use XGBoost are also the two goals of the project:

1. Execution Speed.
2. Model Performance.

**Project Contribution:**

**Rutvij Mavani:** I downloaded the training data file and read that CSV file on R Studio. Apart from doing this I also preprocessed it to implement various machine learning algorithms on it. I divided this training data set into 2 parts namely training set and test set using the ratio of 70:30, so that we can train model on train set and get the accuracy of trained model on this test set. I also handled missing values of Age and Embarked fields.

**Ayush Vashistha:** I analysed the test and train data and determined the target and features of the test and train data. After determining the target and feature I decided to implement classification algorithms as per the problem statement . I applied various classification algorithms such as logistic regression, decision tree, random forest and XGBoost. I got the maximum accuracy for XGBoost among all other classification techniques which I applied (I have only included the XGboost in the final R CODE).

**Varun Jagannathan:** I added the travelers attribute in training set which is addition of SibSp and Parch and 1 so we can get number of passenger traveling on particular PassengerId. I studied various parameters being used in XGBoost and tune those parameters to get maximum possible accuracy using XGBoost algorithm itself.

**Bhirahaspathi Sairam Seeni:** I preprocessed actual test data file to implement trained machine learning model on it for predicting whether passenger is survived or not. I implemented this trained model on actual test data to predict whether passenger is survived or not. I also made CSV file containing PassengerId and Survived attributes. The Survived attribute contains predicted values of whether passenger is survived or not.