**Report on FIFA World Cup 2022**

**Cloud Feasibility Study**

There are several different cloud platforms available, each with its own set of features and services. Some of the most popular cloud platforms include:

Amazon Web Services (AWS): AWS is a comprehensive cloud platform that offers a wide range of services, including computing, storage, database, analytics, machine learning, security, and more.

Microsoft Azure: Azure is a cloud platform that provides a range of services, including computing, storage, database, networking, analytics, machine learning, and more.

Google Cloud Platform: Google Cloud Platform offers a range of cloud services, including computing, storage, database, networking, analytics, machine learning, and more.

Oracle Cloud: Oracle Cloud is a cloud platform that provides a range of services, including computing, storage, database, analytics, and more.

IBM Cloud: IBM Cloud is a cloud platform that offers a range of services, including computing, storage, database, analytics, and more.

Google Collab: Colab is a free Jupyter notebook environment that runs entirely in the cloud. It allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.

Predicting the winner of the 2022 Qatar World Cup using machine learning would likely involve building a model that takes into account various factors such as team strength, team performance, and historical data of past World Cup games played.

We would consider AWS and Collab cloud platforms for this project. Hence in terms of feasibility for deploying a machine learning solution for predicting the World Cup winner on the AWS and Collab cloud platforms, these considerations should be taken account of:

Cost: They both have pay-as-you-go pricing models, which means that you will only be charged for the resources you use. However, it's important to consider the potential costs of using these platforms, particularly if you expect your machine learning model to be heavily used or if you need to store and process large amounts of data.

Scalability: They both offer scalable resources, which means that you can increase or decrease the amount of resources you use as needed. This can be useful for handling sudden spikes in demand for your machine learning model.

Ecosystem: They both large ecosystems of machine learning tools and services, including pre-trained models and libraries for data analysis and visualization. This can make it easier to build and deploy your machine learning model.

Integration with existing tools and systems: If you already have certain tools or systems in place, it's important to choose a cloud platform that can easily integrate with them. AWS and Collab both offer a range of integrations with other tools and services.

Support and documentation: Both AWS and Collab offer good support and documentation, which can be helpful if you are new to machine learning or if you encounter any issues while building and deploying your model.

Overall, both AWS and Collab are good choices for deploying a machine learning solution for predicting the World Cup winner. It's important to carefully consider your specific needs and requirements when choosing a cloud platform for your project.

**Data analysis and opportunity identification**

**Dataset**

Let’s first take a review on the dataset. The dataset available for this project is the ‘international\_matches.csv’, which contains 23,921 rows and 25 columns. The columns (or features) are:

* Date – date that the matches where played
* home\_team – team that is home
* away\_team – country that is away
* home\_team\_continent – continent of the home team
* away\_team\_continent – continent of the away team
* home\_team\_fifa\_rank – rank of the home team
* away\_team\_fifa\_rank – rank of the away team
* home\_team\_total\_fifa\_points – total fifa points of the home team
* away\_team\_total\_fifa\_points – total fifa points of the away team
* home\_team\_score – number of goals scored after a match by the home team
* away\_team\_score – number of goals scored after a match by the away team
* tournament – type of fifa tournament
* city – city where the match was played
* country – country where the match was played
* neutral\_location – whether the location was neither the home or away team country
* shoot\_out – city where the match was played
* home\_team\_result – result of the match in respect to the home team
* home\_team\_goalkeeper\_score – goalkeeping strength of the home team
* away\_team\_goalkeeper\_score – goalkeeping strength of the away team
* home\_team\_mean\_defense\_score – defensive strength of the home team
* home\_team\_mean\_offense\_score – offensive/attacking strength of the home team
* home\_team\_mean\_midfield\_score – midfield strength of the home team
* away\_team\_mean\_defense\_score – defensive strength of the away team
* away\_team\_mean\_offense\_score – offensive/attacking strength of the away team
* away\_team\_mean\_midfield\_score – midfield strength of the away team

The dataset contains data between year 1993 to 2022. It contains all the matches played in different FIFA tournaments ranging from FIFA World Cup qualification, Friendly, African Cup of Nations qualification, Amílcar Cabral Cup, CFU Caribbean Cup qualification, United Arab Emirates Friendship Tournament, Malta International Tournament, Lunar New Year Cup, African Cup of Nations, CFU Caribbean Cup, UEFA Euro qualification, Kirin Cup, FIFA World Cup, Oceania Nations Cup qualification, Baltic Cup, Gulf Cup, Simba Tournament, CECAFA Cup, Confederations Cup, Dynasty Cup, King's Cup, Nehru Cup, SAFF Cup, Copa Paz del Chaco, Korea Cup, USA Cup, Copa América, Merdeka Tournament, South Pacific Games, UNCAF Cup, Oceania Nations Cup, Windward Islands Tournament, Gold Cup, AFC Asian Cup qualification, UEFA Euro, AFF Championship, AFC Asian Cup, King Hassan II Tournament, Cyprus International Tournament, Dunhill Cup, COSAFA Cup qualification, COSAFA Cup, Tournoi de France, Gold Cup qualification, SKN Football Festival, Arab Cup qualification, Arab Cup, UNIFFAC Cup, Nordic Championship, WAFF Championship, Millennium Cup, Cup of Ancient Civilizations, Prime Minister's Cup, EAFF Championship, TIFOCO Tournament, Afro-Asian Games, AFC Challenge Cup, Copa del Pacífico, AFC Challenge Cup qualification, African Nations Championship, VFF Cup, Dragon Cup, Nile Basin Tournament, Nations Cup, Copa Confraternidad, Pacific Games, Superclásico de las Américas, ABCS Tournament, Kirin Challenge Cup, OSN Cup, Copa América qualification, Pacific Mini Games, Intercontinental Cup, AFF Championship qualification, UEFA Nations League, CONCACAF Nations League qualification, African Nations Championship qualification, CONCACAF Nations League, Three Nations Cup, Mahinda Rajapaksa Cup, Navruz Cup, and CONMEBOL–UEFA Cup of Champions.

**Exploratory Data Analysis**

After performing exploratory data analysis (or in other words Descriptive Analysis) the following facts were discovered:

• The average least team strength of any country (or team) is 50. This mean no matter how worse the performance of a country team is, their team strength cannot be less than 50.

• Top 10 teams by rank are Brazil, Belgium, France, Argentina, England, Italy, Spain, Portugal, Mexico, and Netherlands



• Top 10 teams by their offensive strength are Argentina, France, England, Brazil, Portugal, Belgium, Italy, Spain, Poland, and Uruguay



• Top 10 teams by their defensive strength are Spain, Netherlands, Portugal, England, Brazil, France, Italy, Germany, Argentina, and Morocco



• Top 10 teams by their midfield strength are Germany, France, Spain, Belgium, Brazil, Italy, Portugal, Croatia, Argentina, and England



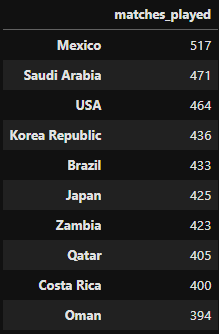
• Top 10 teams by their goalkeeper strength are Slovenia, Germany, Belgium, Brazil, Italy, Costa Rica, France, Poland, Denmark, and Hungary



• The overall 10 best teams are France, Brazil, Germany, Italy, Spain, Belgium, England, Argentina, Portugal, and Netherlands



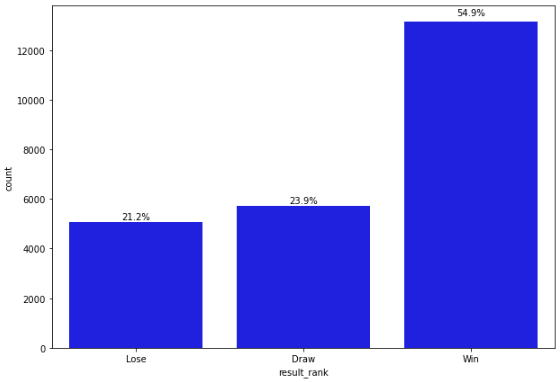
• Top 10 teams with the most played matches are Mexico, Saudi Arabia, USA, Korea Republic, Brazil, Japan, Zambia, Qatar, Costa Rica, and Oman



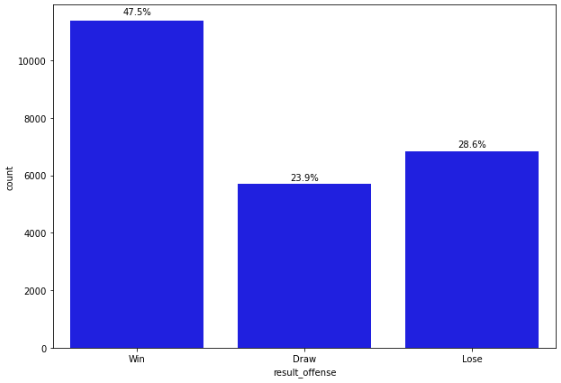
• Top 10 teams with the most wins are Mexico, USA, Saudi Arabia, Egypt, Brazil, Trinidad and Tobago, Costa Rica, Japan, Korea Republic, and Qatar



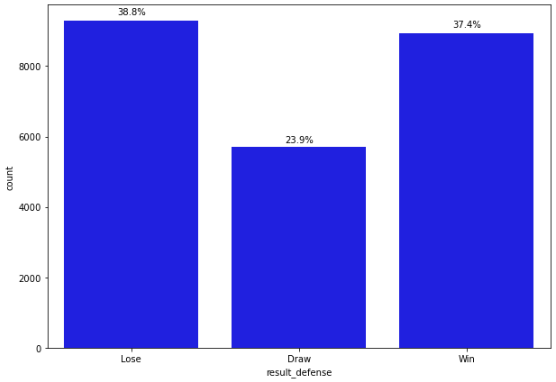
• When a team with higher rank play against another team that has a lower rank, there is a win probability, draw probability, and lose probability of 54.9%, 23.9%, and 21.2% respectively



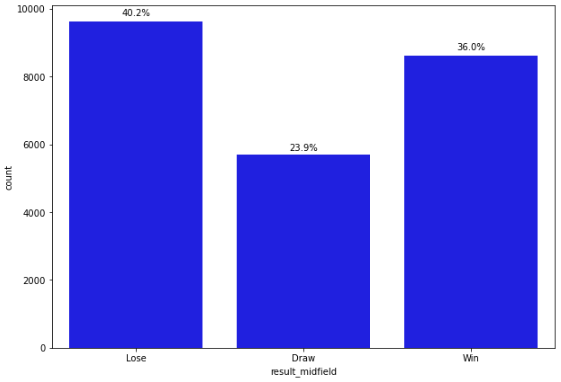
• When a team with higher offense strength play against another team that has a lower offense strength, there is a win probability, draw probability, and lose probability of 47.5%, 23.9%, and 28.6% respectively



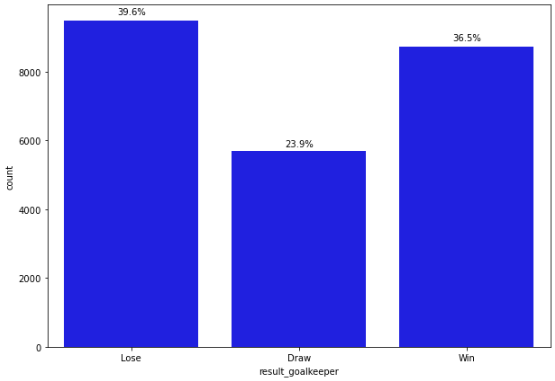
• When a team with higher defense strength play against another team that has a lower defense strength, there is a win probability, draw probability, and lose probability of 37.4%, 23.9%, and 38.8% respectively



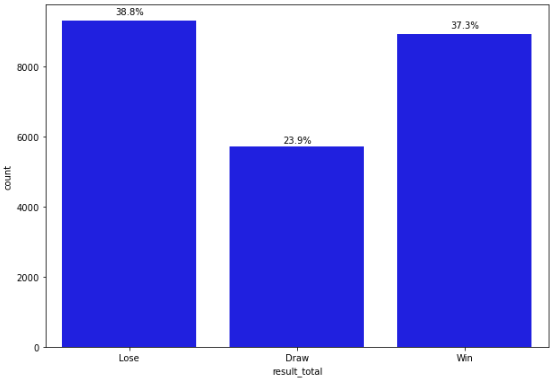
• When a team with higher midfield strength play against another team that has a lower midfield strength, there is a win probability, draw probability, and lose probability of 36.0%, 23.9%, and 40.2% respectively



• When a team with higher goalkeeper strength play against another team that has a lower goalkeeper strength, there is a win probability, draw probability, and lose probability of 36.5%, 23.9%, and 39.6% respectively

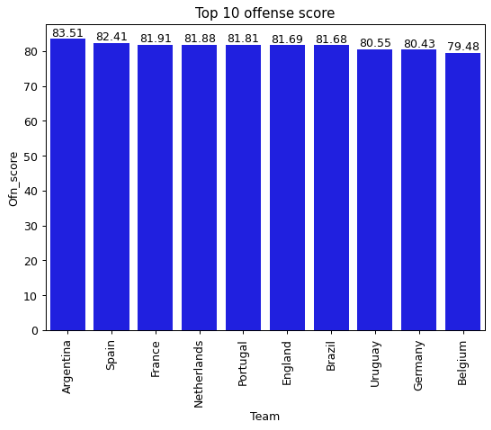


• When a team with higher overall team strength play against another team that has a lower overall team strength, there is a win probability, draw probability, and lose probability of 37.3%, 23.9%, and 38.8% respectively

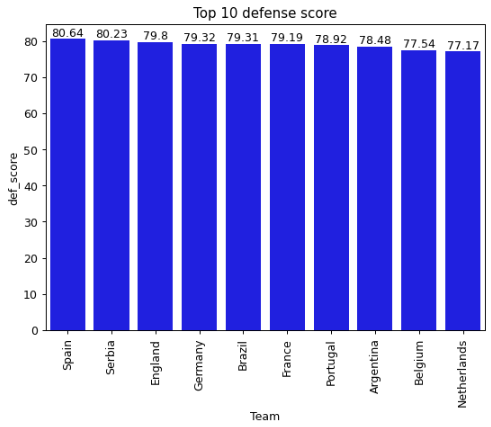


**Teams that were present at the 2022 Qatar world cup**

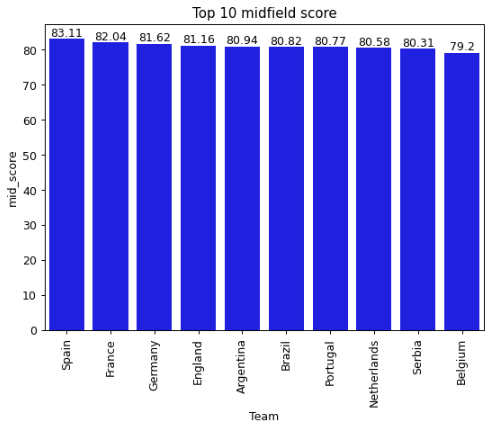
• Top 10 teams by offensive / attacking strength are Argentina, Spain, France, Netherlands, Portugal, England, Brazil, Uruguay, Germany, and Belgium



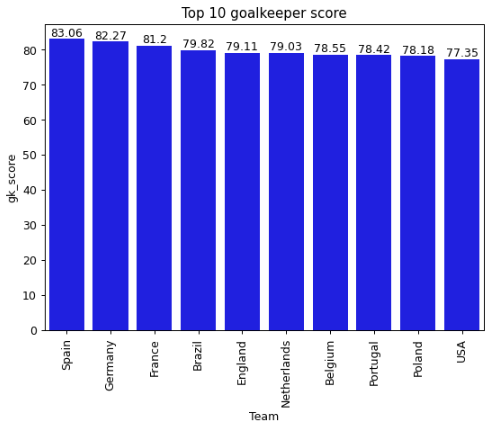
• Top 10 teams by defensive strength are Spain, Serbia, England, Germany, Brazil, France, Portugal, Argentina, Belgium, and Netherlands



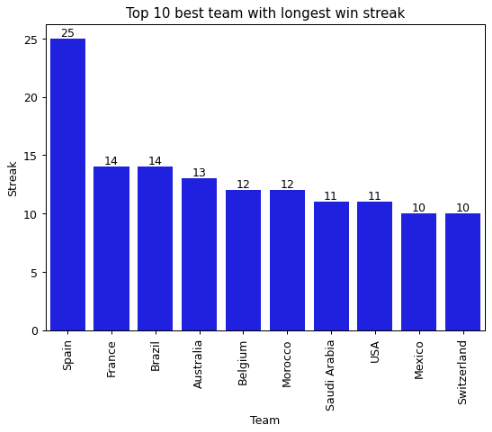
• Top 10 teams by midfield strength are Spain, France, Germany, England, Argentina, Brazil, Portugal, Netherlands, Serbia, and Belgium



• Top 10 teams by goalkeeper strength are Slovenia, Germany, Belgium, Brazil, Italy, Costa Rica, France, Poland, Denmark, and Hungary



• Top 10 teams with the longest win streak are Spain, France, Brazil, Australia, Belgium, Morocco, Saudi Arabia, USA, Mexico, and Switzerland.



**Data pre-processing**

The major preprocessing that was done in this project is the treatment of missing data. The features below contain a huge amount of missing values: home\_team\_goalkeeper\_score, home\_team\_mean\_defense\_score, home\_team\_mean\_offense\_score, home\_team\_mean\_midfield\_score, away\_team\_goalkeeper\_score, away\_team\_mean\_defense\_score, away\_team\_mean\_offense\_score, away\_team\_mean\_midfield\_score.

Before going about how this was done, it is important to understand what treating missing value is, and why it is very important in this project.

In data preprocessing for machine learning, it is common to encounter datasets that contain missing values and this project was not an exception. These missing values can arise for various reasons, such as missing data due to failure of measurement or data collection processes, or because the data was not recorded or stored properly. Regardless of the reason, missing values can pose a problem for many machine learning algorithms, which often require complete and consistent datasets to function properly. Therefore, it is often necessary to fill in the missing values in some way before training the model(s). In this project, the eight features mentioned previously have half of the data missing.

Several different approaches can be used to fill in missing values, and the appropriate method will depend on the specific characteristics of the dataset and the goals of the analysis. Some common methods for filling in missing values include:

Imputation: This involves replacing the missing values with estimates based on the other values in the dataset. One simple imputation method is to use the mean or median of the non-missing values to fill in the missing values. Other more sophisticated methods include using linear regression or other machine learning models to make the imputations.

Interpolation: This involves using known values at nearby points to estimate the missing values. For example, if a dataset contains a series of measurements taken at regular intervals, then linear interpolation can be used to estimate the missing values by fitting a straight line between the known values and using that line to predict the values at the missing points.

Extrapolation: This involves using known values to estimate values outside the range of the known values. Extrapolation can be risky because it involves making predictions beyond the range of the data, which may not be accurate. However, it can be useful in certain situations where there is a strong pattern in the data that is expected to continue beyond the range of the known values.

Interpolation and extrapolation can also be performed using more sophisticated methods, such as spline interpolation or polynomial extrapolation, which can provide more accurate estimates of the missing values.

Dropping rows or columns: In some cases, it may be appropriate to simply drop rows or columns that contain too many missing values. This can be a good approach if the missing values are not uniformly distributed throughout the dataset and removing the rows or columns with missing values does not significantly reduce the size of the dataset.

K-nearest neighbors imputation: This method involves using the values of the K nearest neighbors of a missing value to estimate the missing value. The K nearest neighbors can be determined using a distance measure, such as Euclidean distance, to find the points in the dataset that are most similar to the point with the missing value.

Predictive modeling: Another approach to filling in missing values is to use machine learning models to predict the missing values based on the other values in the dataset. For example, a random forest or gradient boosting model could be trained to predict the missing values based on the other features in the dataset.

In this project, among the above mentioned ways of filling missing value, we used imputation (mean imputation to be precise) where the mean values for each of the above-mentioned were imputed based on the countries (or teams). Since the values will vary from country to country, we took into account the mean values for these countries and imputed them for the missing ones. And we found it reliable.

**Model Selection and Training**

The main libraries used for this project includes: sagemaker, boto3, numpy, pandas, matplotlib, seaborn, and sklearn.

Amazon SageMaker: is a fully-managed platform that enables developers and data scientists to build, train, and deploy machine learning models at scale. It provides pre-built machine learning algorithms and popular libraries, as well as a simple way to create custom machine learning models using your own algorithms or third-party libraries.

With SageMaker, you can train and deploy your machine learning models in a variety of environments, including the cloud, on-premises, and at the edge. SageMaker also provides tools for monitoring and optimizing your machine learning models, including automatic model tuning and machine learning model explainability.

SageMaker is integrated with a variety of AWS services, such as Amazon S3 for storing data and model artifacts, and Amazon EC2 for training and hosting models. It also integrates with other AWS services, such as AWS Glue and AWS Lake Formation, to enable end-to-end machine learning workflows.

Boto3: is the Amazon Web Services (AWS) Software Development Kit (SDK) for Python, which allows Python developers to write software that makes use of services like Amazon S3 and Amazon EC2.

NumPy: is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Pandas: is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

Matplotlib: is a data visualization library in Python that provides a wide range of static, animated, and interactive visualizations in Jupyter notebooks.

Seaborn: is a data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Scikit-learn (sklearn): is a free machine learning library for Python that provides simple and efficient tools for data mining and data analysis. It features various algorithms, including classification, regression, clustering, and dimensionality reduction via a consistent interface.

In carrying out model selection and training, we chose XGBoost (eXtreme Gradient Boosting) algorithm. XGBoost (eXtreme Gradient Boosting) is a popular and efficient open-source implementation of the gradient boosting algorithm. We will be considering the characteristics, advantages and disadvantages of the XGBoost algorithm.

Advantages of XGBoost Algorithm

1. Fast training speed: It is implemented in C++ and is designed to be highly efficient, allowing it to train large models quickly.
2. Scalability: It can efficiently handle large datasets with billions of examples and features.
3. Flexibility: It supports a variety of regularization techniques, such as L1 and L2 regularization, as well as a choice of loss functions. It also supports parallel and distributed training, allowing it to scale to very large datasets.
4. Good performance: It has shown strong performance in a wide range of machine learning tasks and has won numerous Kaggle competitions.
5. Built-in cross-validation: It includes a built-in mechanism for performing cross-validation, which can be used to tune hyperparameters and select the best model.
6. Wide adoption: It is widely used in industry and academia and has a large and active user community, making it easy to find support and resources.

Disadvantages of XGBoost Algorithm

1. It requires more memory and computational resources compared to some other machine learning algorithms, which can make it more challenging to deploy on low-power devices or in resource-constrained environments.
2. It can overfit to the training data if the model is not properly regularized or if there is too much noise in the training data.
3. It can be sensitive to the scale of the features in the training data, so it is often necessary to normalize or standardize the features before training a model.
4. It can be difficult to interpret because it is a complex ensemble model, making it more challenging to understand how the model is making predictions.
5. It is not suitable for real-time predictions because it requires the entire training dataset to make a prediction, which can be slow for large datasets.

In using the SageMaker Python SDK for training a XGBoost algorithm to predict the winner of the 2022 Qatar world cup, the following steps were taken:

1. The data was collected and prepared training the model.
2. The training data was then upload to the Amazon S3, a storage service offered by Amazon. The SageMaker Python SDK provides a convenient method for uploading data to S3.
3. An Amazon SageMaker Estimator object was created. An Estimator represents a machine learning model that can be trained and deployed on Amazon SageMaker.
4. The Docker image for the algorithm was specified, including the
5. Type and number of resources to be used for training, and the hyperparameters for the algorithm.
6. The fit method of the Estimator to train the model was called, and the S3 location of the training data and the name of the SageMaker training job was specified.

**Model evaluation and visualization**

Assessing the performance of a machine learning model is an important step in the machine learning process because it helps you to understand how well the model is able to make predictions on unseen data, and whether the model is overfitting or underfitting the training data.

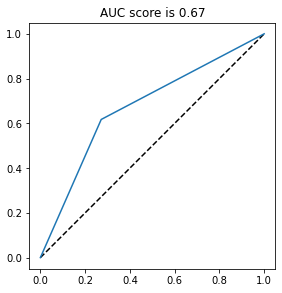
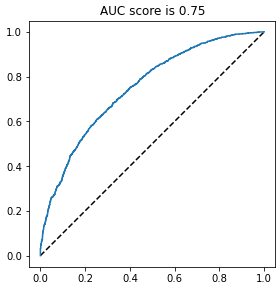
There are several metrics that can be used to evaluate the performance of a model, but we will focus on ROC-AUC (receiver operating characteristic - area under the curve) score, AUC-ROC CURVE, and Confusion Matrix.

ROC-AUC-Score is a metrics commonly used to evaluate the performance of a binary classification model. It is the area under the curve of the receiver operating characteristic (ROC) curve, which plots the true positive rate (TPR) on the y-axis and the false positive rate (FPR) on the x-axis. The ROC-AUC score ranges from 0 to 1, with a higher score indicating better model performance. A perfect model would have an ROC-AUC score of 1, while a model that is no better than random guessing would have an ROC-AUC score of 0.5. The ROC-AUC score is a useful metric because it is insensitive to class imbalance, meaning it can be used to compare models even when the classes are not equally represented in the data. It is also less sensitive to the choice of probability threshold than other metrics, such as accuracy, which makes it a more reliable evaluation metric.

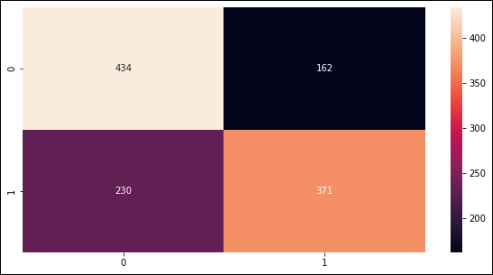
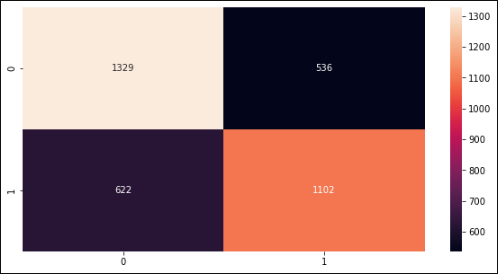
AUC-ROC Curve which is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. It is a plot of the true positive rate (TPR) against the false positive rate (FPR) at various classification thresholds.

Confusion Matrix is a table that is used to evaluate the performance of a classification model. It is a table of counts that shows the number of correct and incorrect predictions made by the model. The confusion matrix is a useful tool for understanding the behavior of a classification model and for identifying areas for improvement. It allows you to see the true positives, true negatives, false positives, and false negatives that the model has produced, and to calculate various evaluation metrics such as precision, recall, and the F1 score.

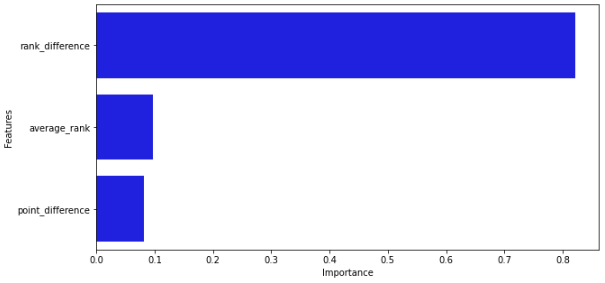
In this project, in other to predict the winner of the world cup, we needed to first predict if a team will win a match or not, therefore we considered a binary problem of either win or lose. After training we got a ROC-AUC-Score of 0.74 using XGBoost algorithm from xgboost library and 0.67 using XGBOOST from the AWS SageMaker SDK.

ROC-AUC Curve for XGBOOST AWS SageMaker ROC-AUC Curve for local XGBOOST

Confusion Matrix for XGBOOST AWS SageMaker Confusion Matrix for local XGBOOST



Feature Importance for the XGBOOST model

Rank differences, which is the difference in the ranks of the two teams (or countries) that played a match, is very significant in determining who will win the match

**Model deployment**

The model was deployed using AWS SageMaker to an Amazon SageMaker Endpoint. An Endpoint is a hosted REST API that you can use to make predictions with the trained model.

To deploy a model using Amazon SageMaker, you can use the following steps:

1. Create an Amazon SageMaker notebook instance: This is a fully managed machine learning (ML) compute instance running the Jupyter Notebook App. It provides access to the ML tools and libraries that you need to train and deploy models.
2. Train your model: In the Jupyter Notebook, you can use SageMaker's built-in high-level Python libraries to train your model using your own data, or using one of the many publicly available datasets.
3. Save your trained model: After you have trained your model, you can use the sagemaker.session.Session.upload\_data method to save your model artifacts to an Amazon S3 bucket.
4. Deploy your model: Use the sagemaker.model.Model class to create a SageMaker Model object, which you can then deploy using the deploy method. This will create a SageMaker Endpoint, which is an HTTPS endpoint that you can use to host your model for real-time inference.
5. Test your deployed model: You can use the sagemaker.predictor.RealTimePredictor class to send test data to your deployed model and get inferences in real-time.

These are the basic steps for deploying a model using Amazon SageMaker. For more information, you can refer to the SageMaker documentation.

Best professional practices can be employed to comply with ethical and privacy concerns

1. Obtain informed consent: Before collecting or using any personal data, it is important to obtain informed consent from the individual(s) whose data you will be collecting. This means that you should clearly explain the purpose for collecting the data, how it will be used, and who will have access to it.
2. Protect data privacy: Personal data should be collected and used only for the purpose for which it was collected, and should be kept secure and confidential. This means taking appropriate measures to protect against unauthorized access, use, disclosure, or destruction of the data.
3. Anonymize data: If you need to use personal data for research or other purposes, but do not want to compromise the privacy of the individuals involved, you can anonymize the data by removing any identifying information.
4. Use de-identification: De-identification is the process of removing or obscuring any information that could be used to identify an individual. This can be done through techniques such as aggregation, suppression, and generalization.
5. Implement data governance policies: It is important to establish clear policies and procedures for how data will be collected, used, and protected. This can help ensure that ethical and privacy concerns are properly addressed.
6. Monitor and audit data usage: Regularly monitoring and auditing how data is being used can help ensure that it is being used appropriately and in accordance with ethical and privacy standards.
7. By following these best practices, you can help ensure that you are complying with ethical and privacy concerns when working with data.

**FIFA World Cup Qatar 2022 simulation and prediction**

A simulation of the fifa world cup Qatar 2022 was created from the round of 16 down to the finals, and we were able to generate the winning probability of the top 5 teams.

The table below shows the result of the top ten team likely to win the world cup with their winning percentage from the 1000 simulation. In all simulation trials the below result are most likely to occur.

|  |  |  |
| --- | --- | --- |
| S/N | Country (Team) | Win (%) |
| 1 | BRAZIL | 23.1 |
| 2 | FRANCE | 12.3 |
| 3 | ARGENTINA | 10.7 |
| 4 | PORTUGAL | 9.0 |
| 5 | SPAIN | 7.7 |

It is important to note the football is very game that is very difficult to predict, as our result here is quite different from the outcome of the FIFA WORLD CUP QATAR 2022, with Argentina emerging as the WINNER having defeated France in the finals. We were close though.

It is also important to note that this result may vary if more simulation trials are executed, by nevertheless, both Brazil and France have higher chances of winning the world cup.

**REFERNCES**

<https://apmonitor.com/pds/index.php/Main/XGBoostRegressor>

<https://www.quora.com/What-are-the-advantages-and-disadvantages-of-XGboost-I-know-they-are-very-good-for-supervised-learning-but-what-are-there-limitations-Do-they-still-perform-well-on-noisy-data>

<https://docs.aws.amazon.com/sagemaker/latest/dg/whatis.html>

https://www.kaggle.com/code/rodrigolacerda/winner-world-cup-simulation-2022