

## **Project: Abalone Age Prediction**

### **Introduction:**

For the Big Data Concepts, I chose a Hands-on project. The main aim of this project is to apply the Big Data concepts and techniques I have learned during the course; therefore, I will be implementing a data pipeline (data cleaning, storage, building ML models, and visualizing the results), proposing some data-cleaning techniques by defining and removing outliers and exploring the Google Cloud platform, which is a big data cloud data platform environment.

First, I will be taking the data from its source, applying data cleaning techniques, removing outliers by analyzing each feature with the target feature, splitting the data into train and test datasets, importing these datasets into Cloud Storage buckets in GCP, utilizing GCP's NoSQL database- BigQuery, building Machine Learning models on the train data, visualizing the evaluation metrics in GCP's visualization tool- Looker Studio, and publishing the results.

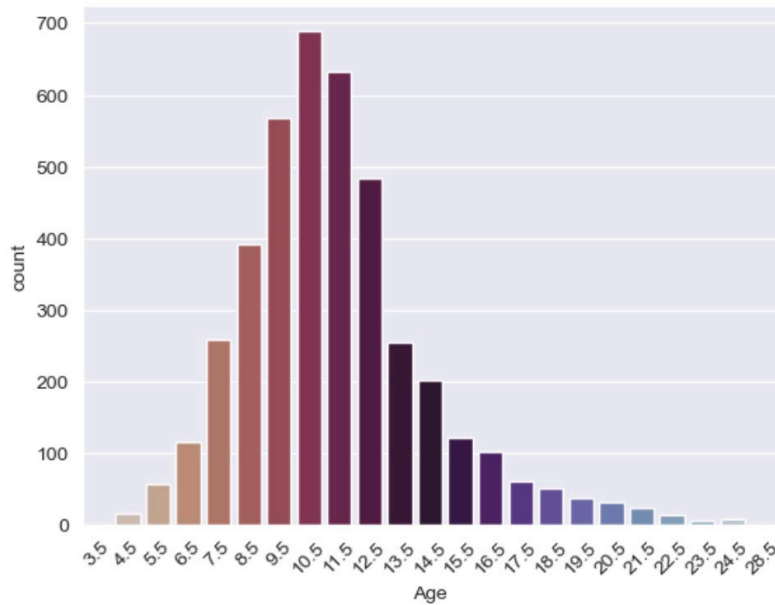
### **Background:**

Cloud computing is important as it provides various advantages like data scalability and flexibility. Building local resources for computation is too expensive, so various space and life sciences organizations are using cloud infrastructure for their various needs. It provides enough computational power to process that huge amounts of data quickly and can also be scaled easily. I have always been interested in life sciences organizations and their advances, and I believed that this would be a wonderful opportunity to work on this data as there has been huge scope for cloud services on such data. I used the Google Cloud Platform, as it provides NoSQL databases, data visualization tools, and buckets for storing the data and publishing the results.

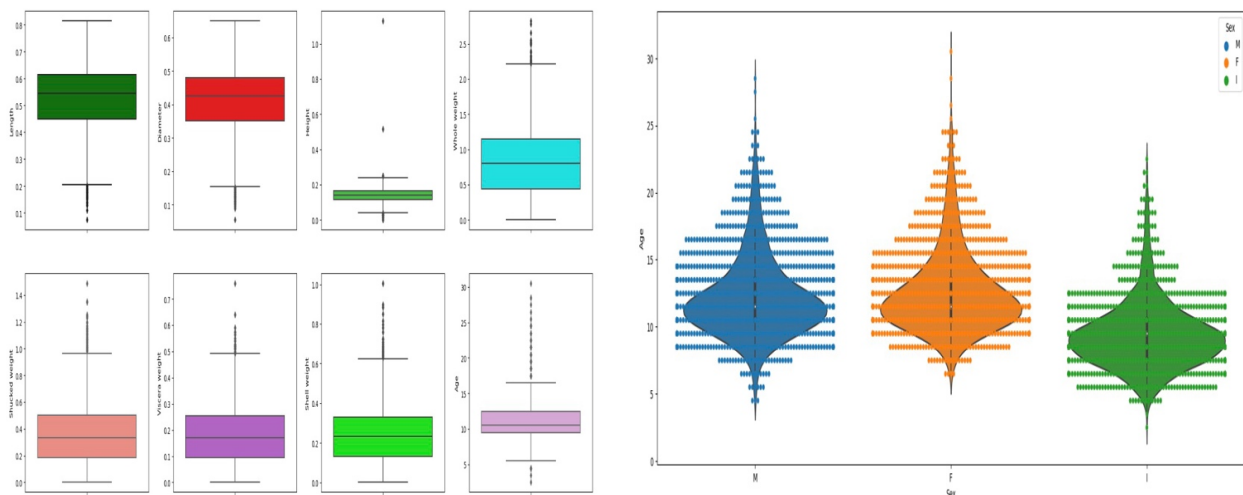
For this project, I will be using the Abalone dataset from UCI ML repository. Abalones are marine snails; these can be found along the coasts of almost all continents. Estimating the age of the Abalone based on its morphological characteristics is the focus of this dataset. The age of Abalone is determined by recoloring, counting the number of rings using a magnifying glass, and slicing the shell through the cone—a laborious and time-consuming process. To predict the age of Abalone, various easier-to-obtain estimates such as length, diameter, shucked weight, and shell weight are used. But these metrics alone may not be enough to determine the age, additional information, such as climatic examples and location (and, consequently, food accessibility), may be needed. The abalone dataset consists of 4177 rows and nine features. Rings are the target variable, and we will obtain the age of the Abalone by adding 1.5 to the ring value in each row. The sex variable in this dataset is interesting because it contains three categories Male, Female, and Infant. But the infant is not considered a sex of the Abalone but instead is about its age. The Abalone age prediction is interesting because we predict various ranges of ages rather than a binary classification.

## Methodology:

I started by taking the dataset from the UCI Machine Learning repository. Initially, I ingested the data into my local. I used Jupyter notebook from Anaconda for basic data cleaning and preprocessing. First, I checked for missing and duplicate values in the data and removed the missing values for better results. The dataset contains Rings as a variable, the age of the abalone can be found when we add 1.5 to the ring value. After this, I plotted a bar chart in Seaborn to analyze the Target variable “Age”.

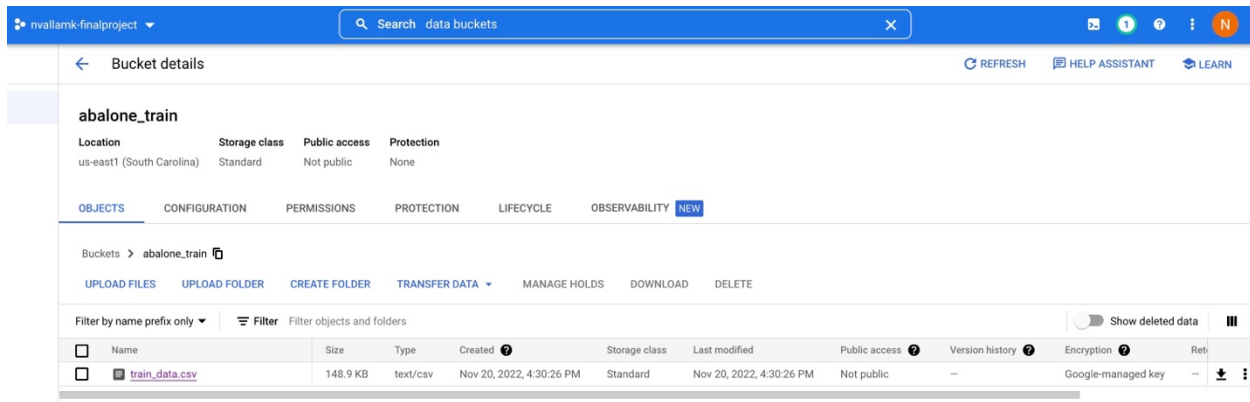


I have also plotted a box plot for each feature of this dataset to better understand the mean and outliers in a variable. Also, the Sex variable in Abalone is particularly interesting as it has three different categories Male, Female, and Infant. The infant is not really sex, but it represents young Abalone for which we cannot determine sex yet.

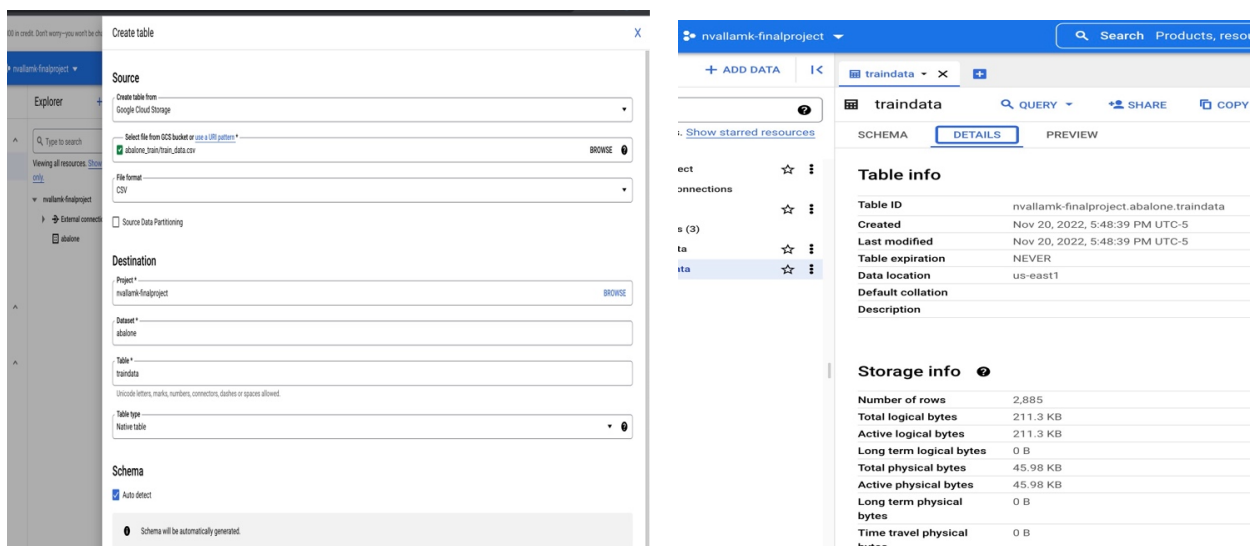


Next, I removed outliers in the dataset by comparing each variable to the target variable. For this preprocessing step, I have plotted Implot in seaborn. Looking at the plot I deleted all the rows that are away too far from the trends in the data.

In this next step, I split the data into train, and test datasets and exported each data into my local environment in form of CSV files. After this, I redeemed the coupon provided by Instructor to set up the environment in the Google Cloud Platform. I have created a project nvallamk-finalproject in the GCP. Then I used Google Cloud Storage and created two buckets for training and test datasets by selecting a location as us-east1, and standard storage options. After creating the buckets, I uploaded the training dataset into the abalone\_train bucket and the test dataset into abalone\_test1 bucket.



Next, I used BigQuery workspace in Google Cloud Platform to build Machine Learning models for the abalone age prediction. BigQuery is a NoSQL and serverless data warehouse in GCP, it can process very large amounts of data within a few seconds. In the BigQuery console, we created a dataset abalone. Under this dataset, we have created two tables one for each train and one test dataset. While creating the tables we pulled the data from the buckets that we created in the Cloud Storage console, instead of uploading the dataset again into BigQuery.



Now the tables have been successfully created, next we will use the powers of BigQuery to implement ML models on the train table. We implement these by writing SQL queries for each model. We created three models for this prediction task, the created models are Logistic Regression, Random Forest, and Boosted Tree from XGBoost. Random Forest take the most amount of time (8 minutes) to train the dataset, while logistic regression took the least.

The screenshot shows the BigQuery console interface. At the top, there are tabs for unsaved queries. The main editor displays a SQL query to create or replace a model named 'nvallamk-finalproject.abalone.rfmodel' using the 'RANDOM\_FOREST\_CLASSIFIER' model type. The query selects 'Age' as the label and uses data from 'nvallamk-finalproject.abalone.traindata'. Below the editor, the 'Query results' section is active, showing 'EXECUTION DETAILS'. The details include: Elapsed time of 8 min 32 sec, Slot time consumed of 32 sec, and a table of stages (Preprocess, Train, Evaluate) all with a count of 0. Training iterations are also shown as Completed: 1 and Planned: 1.

```

1 create or replace model `nvallamk-finalproject.abalone.rfmodel`
2 options(
3   model_type = 'RANDOM_FOREST_CLASSIFIER'
4 )
5 as
6 select Age as label, * ,
7 from `nvallamk-finalproject.abalone.traindata`;

```

Stages	Count
Preprocess	0
Train	0
Evaluate	0

Training iterations	Count
Completed	1
Planned	1

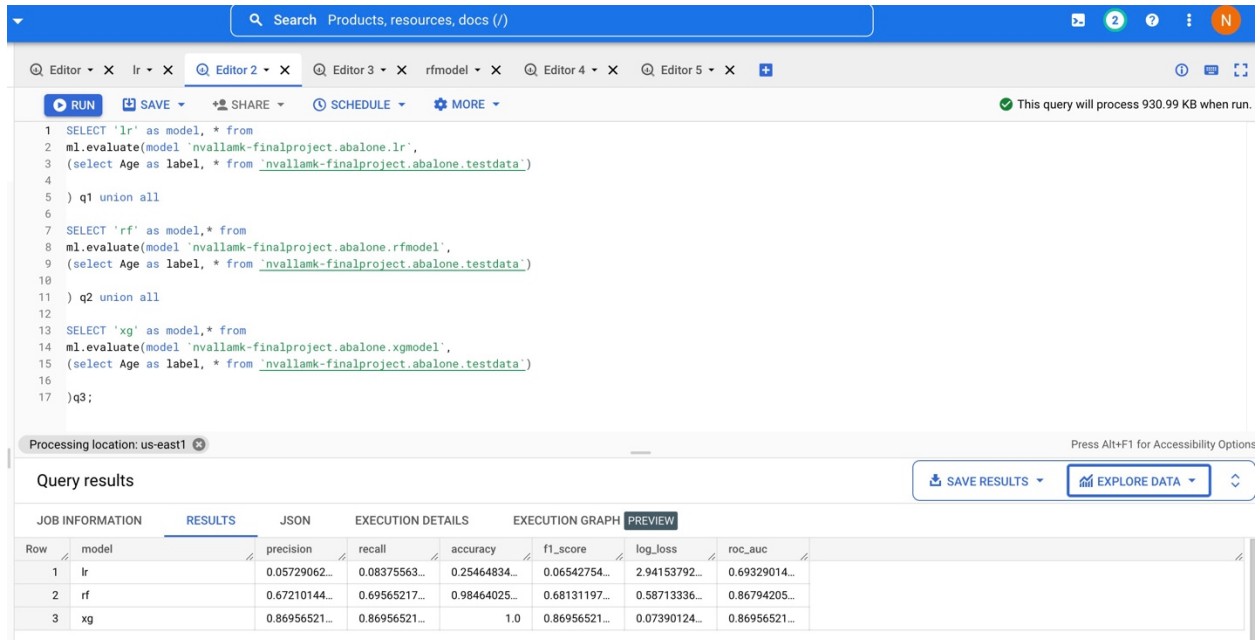
The above picture shows the SQL query of the Random Forest Classifier in the BigQuery console, in a similar, we have also created Logistic Regression and Boosted Tree models. After successfully training the models, we evaluated them by using ml. evaluate in BigQuery. ML.evaluate gives all the evaluation metric values for each model so that we can compare the accuracies of each model. After evaluating the models, I predicted the target using each model. The prediction accuracy has high for younger ages than for older ages in the Random Forest model.

The screenshot shows the BigQuery console with a query that uses the 'ml.predict' function to evaluate the 'nvallamk-finalproject.abalone.rfmodel' on 'abalone.testdata'. The 'Query results' section is active, displaying a table with columns: Row, predicted\_label, predict... label, predict... prob, int64\_field\_0, Sex, Length, and Diameter. The results show a list of predicted labels and probabilities for different age groups.

Row	predicted_label	predict... label	predict... prob	int64_field_0	Sex	Length	Diameter
		14.5	0.00096996...				
		13.5	0.00097028...				
		12.5	0.00097102...				
		11.5	0.00097246...				
		10.5	0.00097572...				
		9.5	0.00098233...				
		8.5	0.00099596...				
		7.5	0.00103191...				
		6.5	0.00111398...				
		5.5	0.97175103...				

## Results:

For the results of this project, I have written a query to evaluate all the ml models used. We used ml. evaluate and aggregated the evaluation metrics for all models.



```
1 SELECT 'lr' as model, * from
2 ml.evaluate(model 'nvaliamk-finalproject.abalone.lr',
3 (select Age as label, * from 'nvaliamk-finalproject.abalone.testdata')
4 ) q1 union all
5
6
7 SELECT 'rf' as model,* from
8 ml.evaluate(model 'nvaliamk-finalproject.abalone.rfmodel',
9 (select Age as label, * from 'nvaliamk-finalproject.abalone.testdata')
10 ) q2 union all
11
12
13 SELECT 'xg' as model,* from
14 ml.evaluate(model 'nvaliamk-finalproject.abalone.xgmodel',
15 (select Age as label, * from 'nvaliamk-finalproject.abalone.testdata')
16 ) q3;
17
```

Row	model	precision	recall	accuracy	f1_score	log_loss	roc_auc
1	lr	0.05729062...	0.08375563...	0.25464834...	0.06542754...	2.94153792...	0.69329014...
2	rf	0.67210144...	0.69565217...	0.98464025...	0.68131197...	0.58713336...	0.86794205...
3	xg	0.86956521...	0.86956521...	1.0	0.86956521...	0.07390124...	0.86956521...

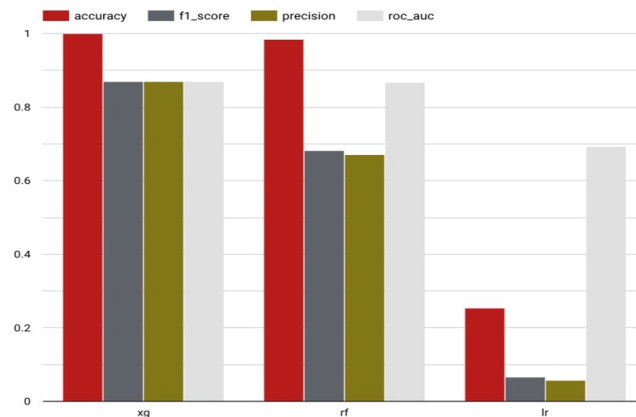
In the results of the query, we can see all the evaluation metrics for all three models. Logistic regression has the least accuracy of 0.25, whereas Random Forest has decent accuracy. The Boosted tree gives an accuracy of 1.0, this indicates that this model is overfitting the data. Also, while comparing the roc\_auc, we observe that Random Forest has 0.86 which is a decent score, and logistic regression has the least roc score of 0.69.

I have also predicted the Age of the Abalone with all three models. As expected, the prediction results for the Random Forest model were the best among the other models. The younger ages were predicted with high accuracy and precision in comparison to older-aged Abalones. I have stored the predicted results of each model in a new one for easier access. I'm attaching one example of the predicted results below. Here, we can observe the prediction label probabilities and slight variation in the predicted label compared to the actual age.

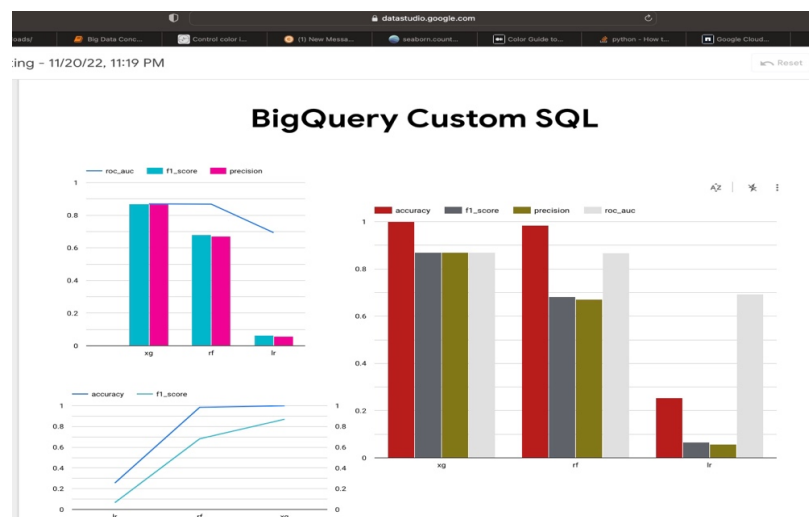
index		predicted_label	predicted_label_probs										int64_field_0	Sex	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell_weight	Age
0	21.5	21.5	[{"label": 28.5, "prob": 0.019140588119626045} {"label": 24.5, "prob": 0.22146698832511902} {"label": 23.5, "prob": 0.050181034952402115} {"label": 22.5, "prob": 0.2043485939502716} {"label": 21.5, "prob": 0.23163117468357086} {"label": 20.5, "prob": 0.025689540430903435} {"label": 19.5, "prob": 0.016259780153632164} {"label": 18.5, "prob": 0.01563879710316658} {"label": 17.5, "prob": 0.015223050490021708} {"label": 16.5, "prob": 0.014932443397896416} {"label": 15.5, "prob": 0.014719514176248594} {"label": 14.5, "prob": 0.014550264943039417} {"label": 13.5, "prob": 0.01441761665046215} {"label": 12.5, "prob": 0.014317568391561508} {"label": 11.5, "prob": 0.014238948002457619} {"label": 10.5, "prob": 0.014194132760167122} {"label": 9.5, "prob": 0.014168058522045612} {"label": 8.5, "prob": 0.01415531660547477} {"label": 7.5, "prob": 0.01414898567040052} {"label": 6.5, "prob": 0.014145683497190475} {"label": 5.5, "prob": 0.014143323766251537} {"label": 4.5, "prob": 0.01414429023861865} {"label": 3.5, "prob": 0.014143873006105423}]										811	F	0.49	0.365	0.13	0.6835	0.165	0.1315	0.205	22.5
			[{"label": 28.5, "prob": 0.019151076674461365} {"label": 24.5, "prob": 0.2215883582830429} {"label": 23.5, "prob": 0.05020853504538536} {"label": 22.5, "prob": 0.20446057617664337} {"label": 21.5, "prob": 0.23175810277462006} {"label": 20.5, "prob": 0.02515559270977974} {"label": 19.5, "prob": 0.016268691048026085} {"label": 18.5, "prob": 0.01564441787800312} {"label": 17.5, "prob": 0.015231393277645111} {"label": 16.5, "prob": 0.014940626919269652} {"label": 15.5, "prob": 0.014727581292390828} {"label": 14.5, "prob": 0.014560640789568424} {"label": 13.5, "prob": 0.014425517991185188} {"label": 12.5, "prob": 0.014325414784252644} {"label": 11.5, "prob": 0.014246751554310322} {"label": 10.5, "prob": 0.014201912097632885} {"label": 9.5, "prob": 0.01417582259359182} {"label": 8.5, "prob": 0.01416307687793994} {"label": 7.5, "prob": 0.0141469739227473736} {"label": 6.5, "prob": 0.014153436826301575} {"label": 5.5, "prob": 0.014162075164020061} {"label": 4.5, "prob": 0.014152041636407375} {"label": 3.5, "prob": 0.014151624403893948}]										2180	F	0.625	0.42	0.165	1.0595	0.358	0.165	0.445	22.5

I have also visualized the evaluation metrics such as precision, accuracy, roc\_auc, and f1\_score of all models in the data visualization tool available for Google Cloud Platform- Looker Studio. We can create various charts in Looker studio such as Scatter plots, Bar charts, Time series charts, etc. As I will just be comparing the evaluation metrics of the Machine Learning models, I utilized line and bar graphs of Looker Studio.

## BigQuery Custom SQL



From the above graph of evaluation metrics, we can see that XGBoost has higher values for all the metrics, but as we can observe this model is overfitting. Random forest gives the actual metrics. After this, I created a bucket and pushed the visualizations of evaluation metrics into this bucket. I have created this bucket with public access so that users can click on the link provided to view the evaluation metric results. The link to the bucket is given here <https://datastudio.google.com/reporting/50e2655a-d79d-46e6-a77a-869feb9376a3>.



## **Discussions:**

### **Interpretation of the results:**

The Machine Learning models we employed in BigQuery have given their predictions for the Age variable. These predictions were evaluated using various metrics in the results. Now, we are going to compare the results in a detailed manner. In the evaluation metrics, we took all the metrics that are available in ML. Evaluate of BigQuery. The provided metrics were precision, accuracy, recall, f1\_score, roc\_auc, and log\_loss. The basic comparison metric is accuracy. From the results, we can observe that Logistic Regression has the worst accuracy of 0.25. XGBoost has the best accuracy of 0.9. But, accuracy does not give a good picture of model performances, accuracy may be high but due to various mathematical techniques involved and the uncertain nature of data, the model may result in better accuracy but fails in correctly evaluating the data. So, we use other metrics to truly understand the model performances. One of the popular evaluation metrics is the roc\_auc value of the models. So, from AUC values we can see the Logistic regression has 0.69 which is not very bad, and XGBoost has a 0.89 AUC value which is a good score. Another important metric is Log\_loss, it works by penalizing the model for the wrong classification of the target variable. Log\_loss works best for multi-class classification as our problem at hand, the lower the value better the prediction. From the results logistic regression has the highest value (2.94), indicating that it does not predict the target very well.

### **Technologies and Skills employed from the course:**

I have used various technologies discussed during the course in this project. I have started with data cleaning, data preprocessing and components analysis, the module Processing and Analytics helped me a better understanding and implement these concepts. I have learned about the Google Cloud Platform from one of the initial modules Cloud computing, this module helped me in understanding various cloud platforms available and the benefits of GCP. I have created Machine Learning models in BigQuery which is a NoSQL database server. I learned about this database server from the module Ingest and Storage. While publishing my evaluation metric visualizations, I had to upload the results in a Cloud Storage bucket and give the public access to all the users, I learned to edit the access of the buckets and generate a link for sharing from one of the qwiklabs of GCP during the assignments.

### **Barriers or Failures faced during the project:**

I faced various barriers during the implementation of this project. The main problem was understanding Looker Studio, the visualization tool of Google Cloud Platform. Drawing charts is one of the important parts of the project as it makes communication of results easier for the users, so I used online resources to work in Looker Studio. Another hurdle was implementing Machine Learning models in SQL, as this is the first time implementing an ML model in SQL, I had to look up various documentation and tutorials of GCP's BigQuery to successfully implement and evaluate ML models.



## Conclusions:

The main aim of this project is to predict the age of Abalones using the powers of the big data cloud platform. Initially, I performed various data preprocessing techniques on the dataset such as data cleaning, and outlier handling. Next, I performed a basic data analysis of all the features present in the dataset. Then I split the data into train and test datasets. These datasets were successfully ingested into Google Cloud storage buckets. After this, I implemented three different Machine Learning models in the NoSQL server- BigQuery. These models were evaluated using various evaluation metrics and predicted the target variable of test data to analyze the performance of models. Then the results of evaluation metrics are visualized using line and bar charts in the Looker Studio, a data visualization tool of GCP. Finally, I successfully published the results using the Looker Studio report-sharing option as well as bucket public viewing of buckets.

## References:

<https://archive.ics.uci.edu/ml/datasets/abalone>

<https://cloud.google.com/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax-evaluate-overview>

<https://towardsdatascience.com/create-a-machine-learning-model-with-google-cloud-bigquery-ml-using-sql-9e2c0ce7fd2d>

<https://windsor.ai/google-data-studio-advanced-tips/>

[https://www.cloudskillsboost.google/focuses/3692?catalog\\_rank=%7B%22rank%22%3A5%2C%22num\\_filters%22%3A0%2C%22has\\_search%22%3Atrue%7D&parent=catalog&search\\_id=14163071](https://www.cloudskillsboost.google/focuses/3692?catalog_rank=%7B%22rank%22%3A5%2C%22num_filters%22%3A0%2C%22has_search%22%3Atrue%7D&parent=catalog&search_id=14163071)