**CIS 415 – Big Data Analytics in Business**

**Project**

**Student Name: Suprya Barbara Azhakath**

**Student ASURite: Sazhakat**

**Section 42360**

**Term, Year: Summer, 2024**

**Contents**

[**1.** **Project Initiation** 2](#_Toc165162694)

[**2.** **Analysis, Design, Environment Setup** 3](#_Toc165162695)

[**3.** **Small Data Solution** 4](#_Toc165162696)

[**4.** **Big Data Solution** 5](#_Toc165162697)

[**5.** **Reflection / Learnings from the project** 6](#_Toc165162698)

# **Project Initiation**

**Project Topic**

2.4 Campaign Effectiveness Analysis

**Project Description**

Analyze the effectiveness of marketing campaigns

**Key Variables**

Campaign\_success

**Why did you choose this topic?**

It’s a fun way to better understand which campaigns are successful and optimizing future marketing efforts.

**How does this project align with your future professional goals?**

Professionally, I’m interested in the analytics marketing campaigns and what number of page visits converts to a sale. Split testing different marketing campaigns and measuring ROI has been and is something that has intrigued me in the past, present, and future.

**What ML algorithm(s) are you considering to use in this project?**

I plan on using Predictive modeling algorithms like Regression models and Classification/ clustering. These are great for classifying outcomes or binary outcomes.

**What skills you expect to learn/strengthen through this project?**

The skills I expect to learn from this outcome if gaining a better familiarity with the ML algorithms I just listed, because I find these very commonly used and I want to personally get better at them. In addition, I want to strengthen my ability to interpret and communicate my findings to future stakeholders or partners. In addition, gain experience in analyzing data trends and patterns.

# **Analysis, Design, Environment Setup**

**Check the small and big datasets.**

**Do you feel that the synthetic values generated for all columns (variables) are reasonable?**

The synthetic values generate for all the columns are reasonable to me, and they are typical metrics that might be used in campaign effectiveness analysis. It’s good to get relevant information and practice.

**Which is your Y variable that you want to predict?**

**Which are the X variables that you plan to use to predict Y?**

My Y value I want to predict is Campaign\_Success.

My X value I want to predict is Campaign\_Reach, Click\_Throough\_Rate, Customer\_Engagment, Campaign\_Cost, Campaign\_Success

**Go through the solution code generated by ChatGPT.**

**Briefly explain the step-by-step flow of the code.**

It didn’t say to paste code so if I was supposed too I’m sorry.

Usually you have the data loading, then data preprocessing, feature selection. After that comes the Model Training, Model Evaluation

**Do you feel ChatGPT generated reasonable code?**

Yes, ChatGPT does generate reasonable code. Sometimes It can overcomplicate. Overall, quite suited to the task and objective of Campaign Effectiveness Analysis.

**Write code to generate descriptive statistics for the small dataset.**

**Copy-paste the descriptive statistics of the small dataset here.**

Descriptive Statistics for Small Dataset:

Campaign\_Reach Click\_Through\_Rate Conversion\_Rate \

count 100.000000 100.000000 100.000000

unique NaN NaN NaN

top NaN NaN NaN

freq NaN NaN NaN

mean 122365.400000 5.390000 2.812000

std 45071.171615 2.397874 1.310353

min 50769.000000 1.000000 0.500000

25% 85745.500000 3.300000 1.800000

50% 117328.000000 5.350000 2.700000

75% 166998.500000 7.300000 3.925000

max 199503.000000 10.000000 5.000000

Customer\_Engagement Campaign\_Cost Campaign\_Success

count 100.00000 100.000000 100

unique NaN NaN 3

top NaN NaN High

freq NaN NaN 52

mean 67.28000 5945.990000 NaN

std 21.46789 2364.370664 NaN

min 30.00000 2114.000000 NaN

25% 47.50000 4127.750000 NaN

50% 70.00000 5701.000000 NaN

75% 87.00000 7990.750000 NaN

max 99.00000 9903.000000 NaN

**Which IDE are you using to build and test the Small data solution? Google Colab or a different environment?**

I will be using Google Colab, mostly because that’s what my laptop likes and I can move around with it compared to my desktop.

**Did you successfully run your code in the small data environment?**

Yes, I was able to run the code successfully in a small environment.

**Which models did you train and evaluate (e.g. linear regression, random forest etc.)**

In this project I used Logistic Regression and Decision Tree Classification Models.

**What evaluation metric you used? (e.g. F1-score, AUC, RMSE, MSE)**

F1 Score

**List (or copy-paste) the model evaluation results (e.g. model 1 -> F1: 0.46, model 2 -> F1: 0.64)**

Model 1: Logistic Regression

Confusion Matrix:

[[9 1 3]

[3 0 1]

[3 0 0]]

Classification Report:

precision recall f1-score support

0 0.60 0.69 0.64 13

1 0.00 0.00 0.00 4

2 0.00 0.00 0.00 3

accuracy 0.45 20

macro avg 0.20 0.23 0.21 20

weighted avg 0.39 0.45 0.42 20

F1 Score: 0.41785714285714287

**Based on the small data set, which algorithm/model seems to be performing the best?**

In this case with the small dataset, Logistic Regression performed decent with a F1 score of 0.41

**Are you happy with the “best” model? Or you would prefer to check if there are better models?**

With the small dataset I’m happy with the performance of the Logistic Regression model. I would still prefer to check out decision models like clusters and classification.

**Briefly share the challenges you faced in this step**

I believe the biggest challenge was creating adequate data sets for both small and large data, and then scaling the features and reducing overfit. In addition, I’m not used to mockup data, so this is new to me. I’m very happy with my logistic regression model and would love to try tree classification.

# **Big Data Solution**

**Did you make any change to the PySpark code after moving it from Google Colab to GCP Dataproc Jupyter? If yes, briefly mention the changes made to the code.**

Yes, I had to use GCS paths like using “gs://” and basically had to adapt to GCP Dataproc’s cluster environment by using “from pyspark” related libraries.

**List (or copy-paste) the model evaluation results for SMALL DATA (e.g. model 1 -> F1: 0.46, model 2 -> F1: 0.64)**

Model 1 (Logistic Regression) - > F1:0.4178571428571428 , model 2 (Decision Tree) - > F1: 0.3795454545454546

**List (or copy-paste) the model evaluation results for BIG DATA (e.g. model 1 -> F1: 0.46, model 2 -> F1: 0.64)**

Model 1 (Logistic Regression) - > F1: Model 1 (Logistic Regression) - > F1:0.3353686487818195 , model 2 (Decision Tree) - > F1:0.37914652515381575

**Based on the BIG dataset, which algorithm/model seems to be performing the best? Is it same as the best performing algorithm on the SMALL dataset.**

Based on the results from different models on both small and big data, the best preforming algorithm was the Decision Tree, while the least best was Logistic Regression. In comparison, for small data the best preforming algorithm was Logistic regression, while the least preforming algorithm was in fact the Decision tree.

**Briefly share the challenges you faced in this step**

I believe the biggest challenges for me was ensuring the accuracy of the model, especially since I already have the F1 score for model 1 on Small Data. I wanted to make sure that remained consistent with running the code again and comparing it with logistic regression. Especially since Big Data is involved, I want to make sure there was no overfit to maintain this accuracy, so I was consistently looking over my code.

# **Reflection / Learnings from the project**

**Briefly reflect and share your experience and learnings from this project.**

So far, I’ve learned how differently working with both small and big data impacts performance. Logging into GCP everything worked exactly as it had in google Colab, so I was able t use the knowledge from GCP Lab 5 to test out both small and big data. Now I know a little bit more about the google cloud and PySpark. The examples are from the F1 scores, and which model performed better or what. In addition, I learned how to generate synthetic data for a business objective more unsupervised learning model. I got to get acquainted with the metadata and understand how well my dataset can perform. Overall, this was a successful introductory project. The skills I learned from this project will come In handy in the future when I’m working in the cloud of need to work with either big or small data.