HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and Communication Technology

Introduction to Business Analytics

Conversion Optimization

Group Members:
Nguyen Vu Thien Trang - 20194459
Nguyen Van Thanh Tung - 20190090
Chu Hoang Duong - 20194429

Supervisor:
Prof. Nguyen Binh Minh

Table of Contents

Chapter 1

Scenario

1.1 Big Picture

Marketing is the one of the most important stage that paves the way for broadening the reputation of the products/brands and returning sales. As a consequence, a company does a lot of marketing to boost the brand's reputation and get sales, and various kinds of campaigns are initiated to market products. Exploiting the trend of the meta-verse and social networking, various companies choose to approach their customers or audiences through social networking platforms such as Facebook, YouTube, TikTok, Instagram, etc. Facebook, which leads in the number of users, provides paid ads for business companies, where businesses can setup ads with customized settings for the target audience and budget spent.

To that end, the goal of this project is to learn how the independent variables (the settings) influence the returned values and whether businesses can utilize this dependence to enhance their reputations and revenues.

For this Business Analytic project, we use the given dataset [1] of an incognito business with their tracked returns when broadcasting their campaigns with different settings through Facebook Ads Manager. Since the data was encoded entirely in anonymous form, we cannot reference the data to other customer demographic data or detailed campaign features. It follows that we would mainly focus on the dependencies and relationships between variables in the table and see if these correlations can be optimized for generating more sales and gaining more reputation.

In the context of big data, we have designed a distributed system combining different frame works suitable for storing and processing data, thereby taking advantage of computational resources, extracting insights of data.

1.2 Contributions

- We discover the properties of distributions of the returned variables and use regression analysis to find their correlation.

[1] https://www.kaggle.com/datasets/loveall/clicks-conversion-tracking

- We investigate if the distribution of returns differs when breaking down the data into different combinations of setting variables. We test the hypotheses to see if the difference in the distribution is really affected by the independent variables or just by chance.
- We build models to predict the returned variables with different settings and explore the dependency between independent variables and dependent variables.
- Based on the analysis, we research whether the business can exploit the dependency between variables to enhance its reputation and boost sales.

The source code is released at `https://github.com/nguyenvuthientrang/BusinessAnalysis.git`

1.3 Assigned Tasks

In term of assigning tasks, team members are guaranteed the same amount of work, specifically as follows:

- **Nguyen Vu Thien Trang** Sanity Check, Data Engineering, Quantitative Exploratory Data Analysis, Building Machine Learning Models
- **Nguyen Van Thanh Tung** design System Architecture and Data Pipeline; setup Hadoop cluster, Spark, Jupyter Lab on 3 machines; Integrating Code into System
- **Chu Hoang Duong** make Data Visualization, make, Hypothesis Testing; actively support other members in other tasks

Chapter 2

System Architecture and Data Pipeline

2.1 System Architecture

The Conversion Optimization problem mentioned above needs to be handled with many stages, to handle each stage well, we designed a system with the following architecture:

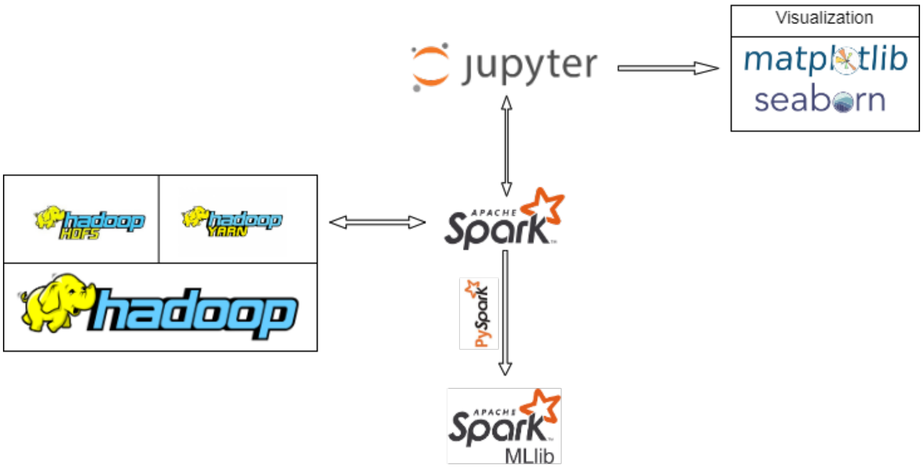


Figure 2.1: System Architecture

The system is composed of different frameworks, each of which will handle different tasks, allowing data to pass through them and be processed efficiently.

2.1.1 Hadoop Cluster

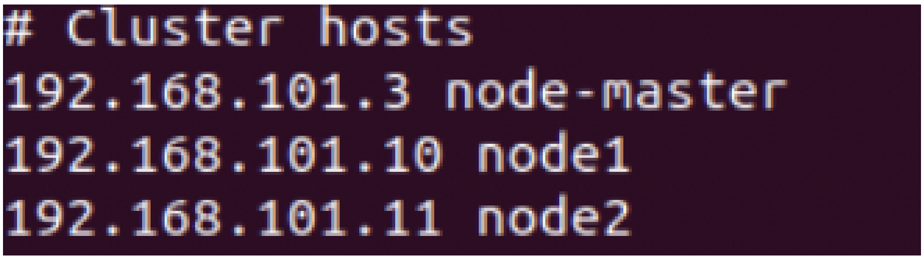
Hadoop is a framework permitting the storage of large volumes of data on node systems. The Hadoop architecture allows parallel processing of data using several components:

- **Hadoop HDFS** to store data across slave machines

- **Hadoop YARN** for resource management in the Hadoop cluster

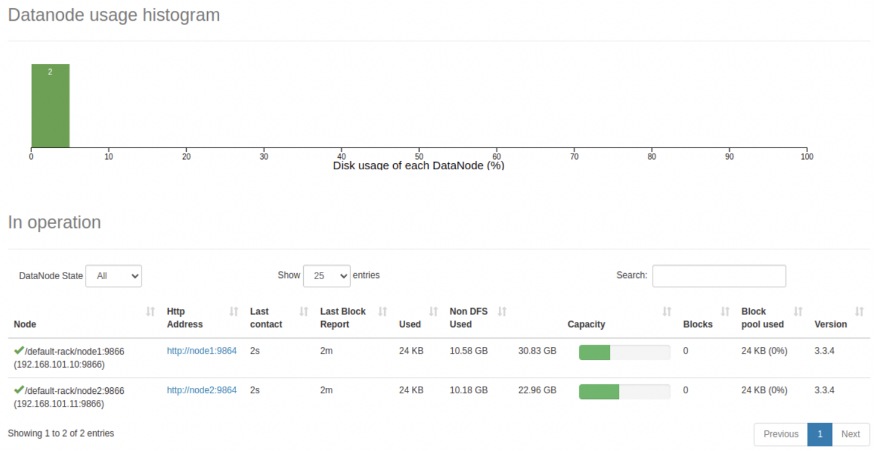
Within the limitations of this project, and also because of resource limitations, we designed a small Hadoop cluster consisting of 3 machines, in which 1 node will act as the master node and the other 2 nodes will be the data nodes. These nodes are connected to each other in an internal network, identified by the IP address in the network so they can communicate with each other.

We can see the cluster information shown in the following figures:



```
# Cluster hosts
192.168.101.3 node-master
192.168.101.10 node1
192.168.101.11 node2
```

Figure 2.2: Node - Local IP address



Datanode usage histogram

In operation

DataNode State All

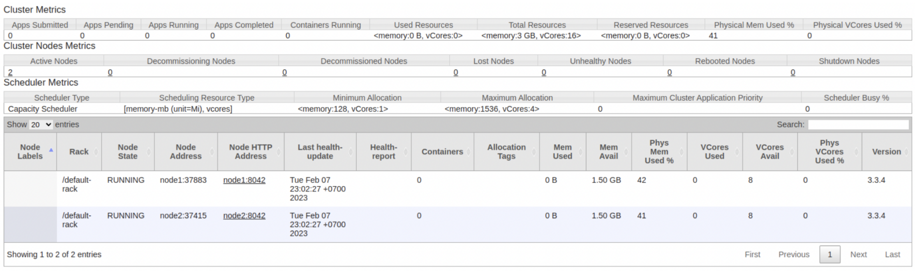
Show 25 entries

Search:

Node	Http Address	Last contact	Last Block Report	Used	Non DFS Used	Capacity		Blocks	Block pool used	Version
✓/default-rack/node1:9866 (192.168.101.10:9866)	http://node1:9864	2s	2m	24 KB	10.58 GB	30.83 GB	🟩	0	24 KB (0%)	3.3.4
✓/default-rack/node2:9866 (192.168.101.11:9866)	http://node2:9864	2s	2m	24 KB	10.18 GB	22.96 GB	🟩	0	24 KB (0%)	3.3.4

Showing 1 to 2 of 2 entries

Figure 2.3: HDFS - Datanodes Report



Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Used Resources	Total Resources	Reserved Resources	Physical Mem Used %	Physical VCores Used %
0	0	0	0	0	<memory:0 B, vCores:0>	<memory:3 GB, vCores:16>	<memory:0 B, vCores:0>	41	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
2	0	0	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority	Scheduler Busy %
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:128, vCores:1>	<memory:1536, vCores:4>	0	0

Show 20 entries

Search:

Node Labels	Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report	Containers	Allocation Tags	Mem Used	Mem Avail	Phys Mem Used %	VCores Used	VCores Avail	Phys VCores Used %	Version
	/default-rack	RUNNING	node1:37883	node1:8042	Tue Feb 07 23:02:27 +0700 2023		0		0 B	1.50 GB	42	0	8	0	3.3.4
	/default-rack	RUNNING	node2:37415	node2:8042	Tue Feb 07 23:02:27 +0700 2023		0		0 B	1.50 GB	41	0	8	0	3.3.4

Showing 1 to 2 of 2 entries

Figure 2.4: YARN - Cluster Nodes Report

In this project, our main purpose of using HDFS is to store data and YARN is to utilize resources of the cluster.

2.1.2 Spark

Apache Spark is a multi-language engine for executing data engineering, data science, and machine learning on single-node machines or clusters.

We use Spark for two main purposes:

- First is extract data from HDFS and this data will go to frameworks like **seaborn, matplotlib** for visualization.
- Second is using **Spark MLlib** to be able to manipulate machine learning tasks.

Since we already had an HDFS for data storage and YARN for Hadoop cluster resource utilization, our spark tasks were set up to run on YARN.

2.1.3 Jupyter Lab

Jupyter Lab plays an important role in the system. Because we use frameworks like python's Seaborn, python's Matplotlib, the data processing and data visualization will take place by python on Jupyter Lab. Besides, Jupyter Lab also acts as a Spark Driver [1] that connects Pyspark to YARN server to perform tasks.

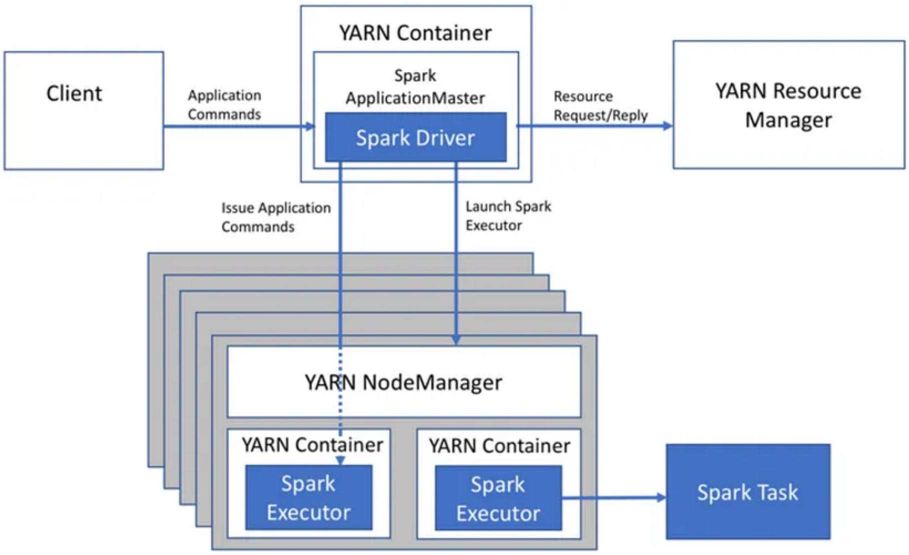


Figure 2.5: Jupyter Lab play Spark Driver role

[1] https://medium.com/@goyalsaurabh66/running-spark-jobs-on-yarn-809163fc57e2

2.2 Data Pipeline

Below is the data flow as it passes through our system. If in the above section, we have clearly explained each component of the system, then in this part we will go into how the data will be processed once it has been collected as a CSV file.

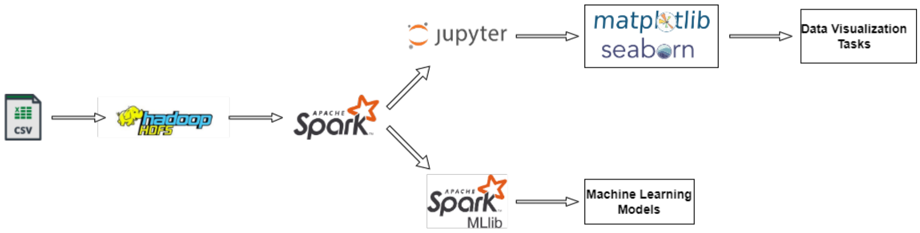


Figure 2.6: Data Pipeline

We choose CSV file because this is a common file format, we can completely replace CSV file with other file formats without changing the data flow.

Once the data has been collected on the machines, aggregated as csv, they will be loaded into HDFS. When Spark connects to YARN, it can also extract CSV files stored on HDFS, putting data into different frame works for different purposes:

- **Data Visualization** by Seaborn, Matplotlib on Jupyter Lab
- **Machine Learning engineering** on Spark MLlib

In summary, we have a complete system and data flow for storing and processing data. This system can be scaled up easily by increasing the number of nodes of the Hadoop cluster, which makes it convenient when the data volume scaling up in the future.

Framework	Task
HDFS	Data Storage
YARN	Resource Management
Spark MLlib	Machine Learning Engineering
Jupyter Lab	Spark Driver, Python environment
Seaborn, Matplotlib	Data Visualization

Table 2.1: System Components

Chapter 3

Data Engineer and Analysis

3.1 Data Summary

The data used in this project is from an anonymous organisation's social media ad campaign. The data contains total of 1143 observations in 11 variables. No demographics on customers or details on the campaign was linked to the data.

```
database
 |-- fb_campaign:
 |     |-- 1. fb_campaign_id: string (primary_key)
 |     |-- 2. age: string
 |     |-- 3. gender: string
 |     |-- 4. interest: string
 |     |-- 5. xyz_campaign_id: string
 |-- coversions:
 |     |-- 1. fb_campaign_id: string (primary_key)
 |     |-- 2. Spent: double (primary_key)https://www.overleaf.com/project/63de269640d9ac33c
 |     |-- 3. Impressions: integer
 |     |-- 4. Clicks: integer
 |     |-- 5. Total_Conversion: integer
 |     |-- 6. Approved_Conversion: integer
```

In theory of normal forms, the original dataframe can be breakdown into 2 tables. How ever, for effective queries we join the two table based on the *fb campaign* key.

3.2 Sanity Check

Before working on any feature engineering and exploratory data analysis, we check the cleaness and the integrity of the data.

- **Missing data**: The dataframe contains no null/nan values
- **Redundant data**: The data frame contains no duplicate values, the number of unique records is equal to the number of rows.

- **Constraint violation**: From the definition of each column, we have inferred some constraints that the data should meet:
  - If Spent = 0 $\implies$ Impression, Clicks, TotalConversion, Approved-Conversion = 0
  - ApprovedConversion < TotalConversions < Clicks < Impressions

  However, both the constraints was violated. For the case Spent = 0 but still gaining Impressions and Conversions, we assume that the auto-recommender system of Facebook recommend the ads for non-targeted audience, hence still returning conversions. We also found values that there are some none-clicked ads but still have conversions. Thus, not all enquiries/sales comes from the ads.

3.3 Feature Engineering

Since the target of this project is to optimize the conversion rate, we add some features that are defined by converting a values to others. It is noteworthy that only the Clicks follow the condition on the amount of Spent, hence we create 2 Reputation Gain features, which are Click Through Rate (CTR) and Cost per Click (CPC).

$$CTR = \frac{Clicks}{Impressions}$$

$$CPC = \frac{Spent}{Clicks}$$

We also focus on the sales conversion rate. Conversions can be generated from non-clicked ads, hence we only formulate the rate to convert from Impression to the two Conversions (TCR, ACR). We also want to analyse the approval rate (AR) after inquiries.

$$TCR = \frac{TotalConversion}{Impressions}$$

$$ACR = \frac{ApprovedConversion}{Impressions}$$

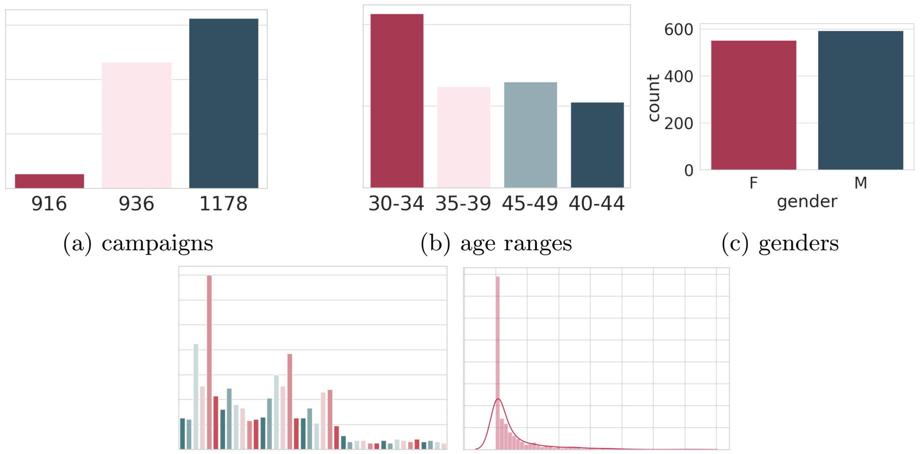
$$AR = \frac{ApprovedConversion}{TotalConversion}$$

The final dataframe is structured as follow:

```
|-- conversion_data:
|    |-- 1. ad_id: string (primary_key)
|    |-- 2. fb_campaign_id: string
|    |-- 3. age: string
|    |-- 4. gender: string
|    |-- 5. interest: string
|    |-- 6. xyz_campaign_id: string
|    |-- 7. Spent: double
|    |-- 8. Impressions: integer
|    |-- 9. Clicks: integer
|    |-- 10. Total_Conversion: integer
|    |-- 11. Approved_Conversion: integer
|    |-- 12. CTR: integer
|    |-- 13. CPC: integer
|    |-- 14. TCR: integer
|    |-- 15. AR: integer
|    |-- 16. ACR: integer
```

3.4 Exploratory Data Analysis

The features of the used data include independent variables and dependent variables. We have five independent variables; besides the impressions (the views on the ads), the number of clicks on the ads, the number of inquiries on the product in the ads, and the final sales of the products, the values of these campaign returns are dependent variables that we want to optimize in this project.



(a) campaigns

(b) age ranges

(c) genders

(d) interest

(e) budgets

Figure 3.1: independent variables

The correlogram shows that all the return values are going down with the money spent on the ads. The correlation seems to be stronger for Spent, Clicks and Impressions, and not as strong for the two left conversions. The graph already explains quite well the relationship between the numerical variables. Otherwise, a more systematic tool that can evaluate how the dependent variables depend on the independent variables is necessary. And this is called regression analysis.

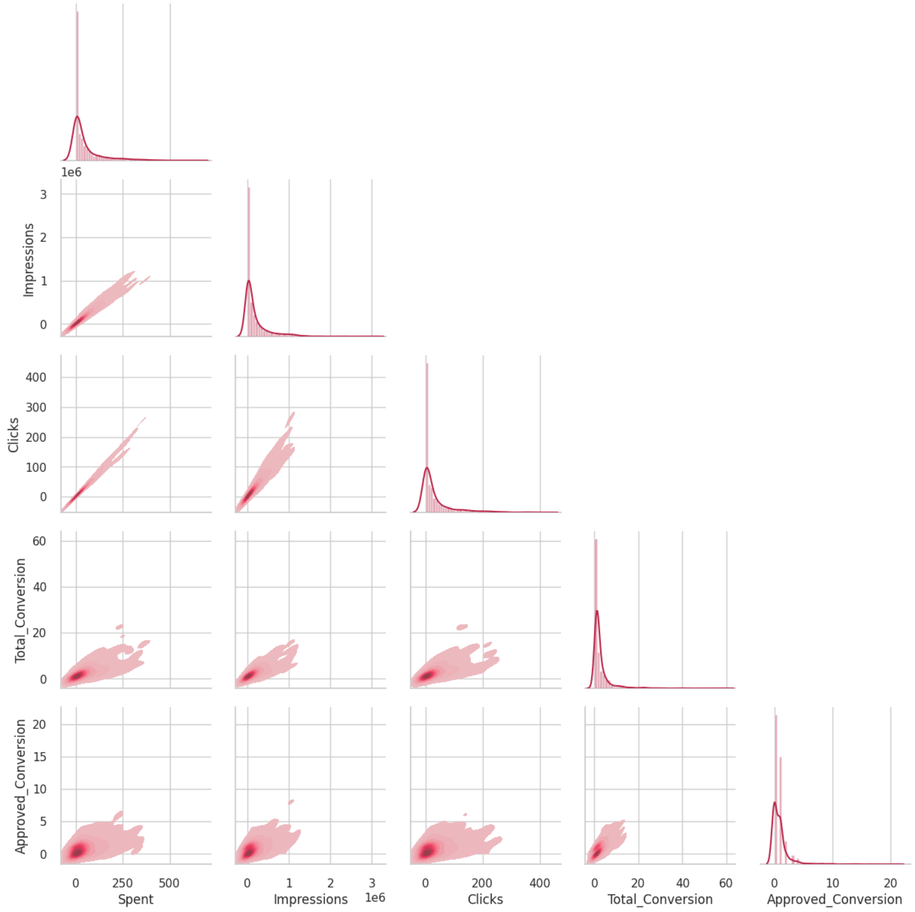


Figure 3.2: Correlation of dependent variables

Chapter 4

Machine Learning and Hypothesis Testing

4.1 Usecase 1: Personalize Spent on Audience

Given a budget or a range of amount available for spending on ads, return the amount of spent that most effective or can balance the most between spent and return values (impressions, clicks, etc.)

4.1.1 Tracker

Normally, testing strategy can directly tracks the outcome given different versions of input. Since we have no control on the campaigns, we cannot directly track the return values with different versions of the setting. As a result, it necessitates the appearance of a virtual tracker capable of predicting the outcome given a predefined customization. We exploit the RandomForestRegreesor playing the role of the virtual tracker in this section.

	Impressions	Clicks	TotalConversion	ApprovedConversion
Accuracy	96.43	97.83	70.13	53.69
numTrees	10	50	10	20
maxDepth	10	20	10	20

Table 4.1: Detail of the RandomForestRegressor.

Leveraging the reduction in entropy or the purity gain when splitting the data using a variable when building the tree by RandomForestRegressor, we get the calculated feature importance on Clicks and Impressions. As illustrated in Figure 4.1, for both calculations, the importance of Spent dominates the other factors.

4.1.2 Metric

As in other testing strategies, a metric is necessary to evaluate which version or which value is the best. We define the point that balances the money spent and

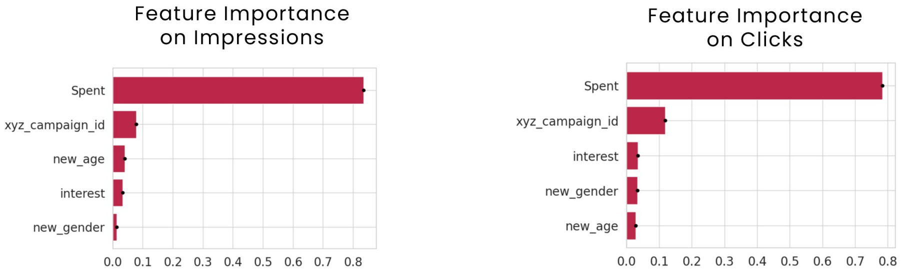


Figure 4.1: Feature Importance calculated the RandomForestRegressor on Impressions and Clicks.

the return values. We state that the optimized value is the one that leads to the strongest increase in the return value when stepping by one unit of Spent. Based on the derivative, we formulate the metric as follows:

$$\frac{f(s) - f(s - \triangle s)}{\triangle s}$$

where s is the Spent and $\triangle s$ is one unit of Spent. The value that achieve the highest metric is the optimal point.

4.1.3 Pipeline

We summarize our pipeline for the first use case as follows:

- We first define a target audience group and select a target return value between clicks and impressions (conversions are not recommended because the model cannot learn its distribution well).
- We then loop over the given budget range with a predefined step size and track the ouput with the trained regressor.
- The best value for Spent is the point that has the highest jump on the return values when increasing by a unit. The result of the method is shown in Figure 4.2.

4.2 Use case 2: Customer Clustering

We notice that the CTR(Click through Rate) and CPC(Cost per Click) have two modes, and the AR have three modes, hence we pose use case 2, clustering the customers based on these modes.

We use the well-known K-means to cluster the data. By clustering the CPC (Cost per Click) and CTR (Click through Rate) into 2 clusters, we get a more potential audience group that have higher CTR, and another group with no Clicks. Similarly, after clustering based on AR(Approval Rate), the audience is broken down into three groups with different levels of approval rate.

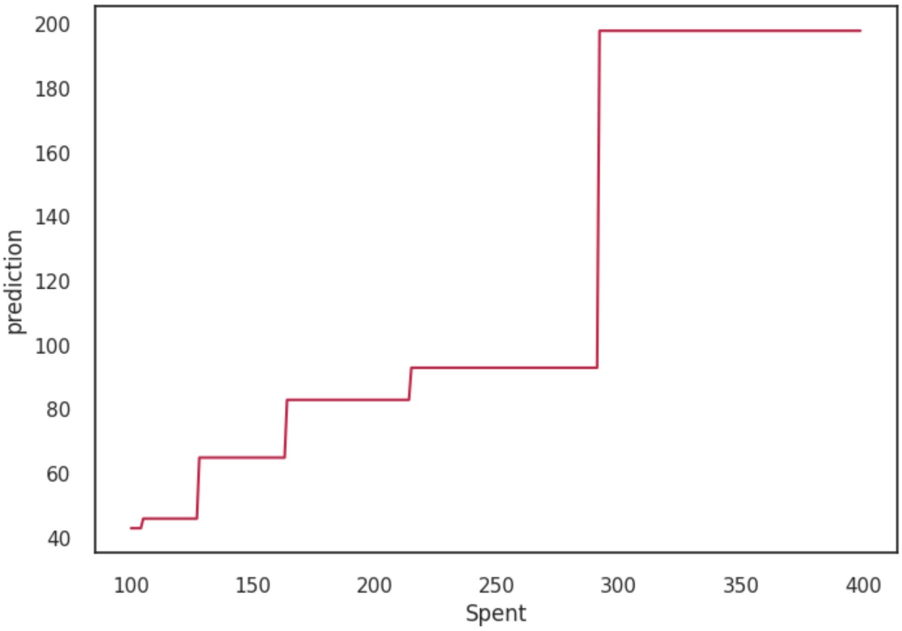


Figure 4.2: The Clicks predicted by the Regressor when looping in the budget range $(100, 400)$ with step size 1. The best Spent is 292 with a predicted number of 198 Clicks.

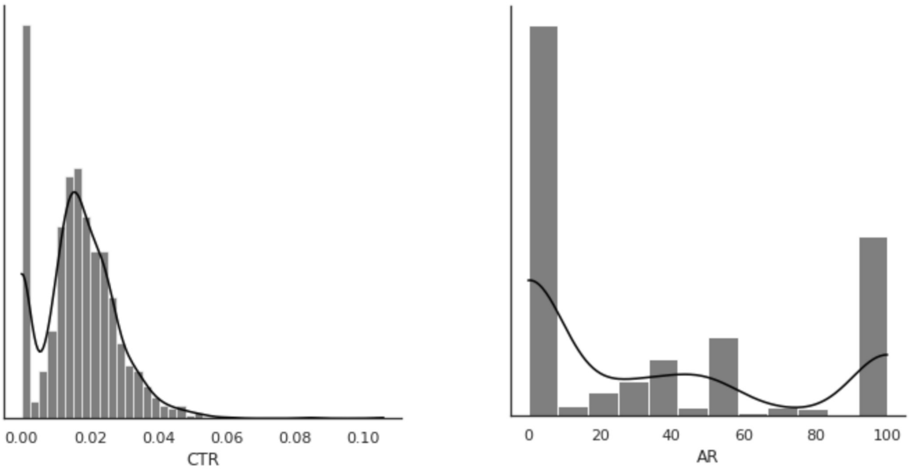


Figure 4.3: The distributions of CTR (Click-through-Rate) and AR (Approval Rate).

4.3 Hypothesis Testing: Explore Potential Customers

We added TCR (Total Conversion Rate) and ACR (Approved Conversion Rate) but have not worked on them. This section attempts to examine these two in

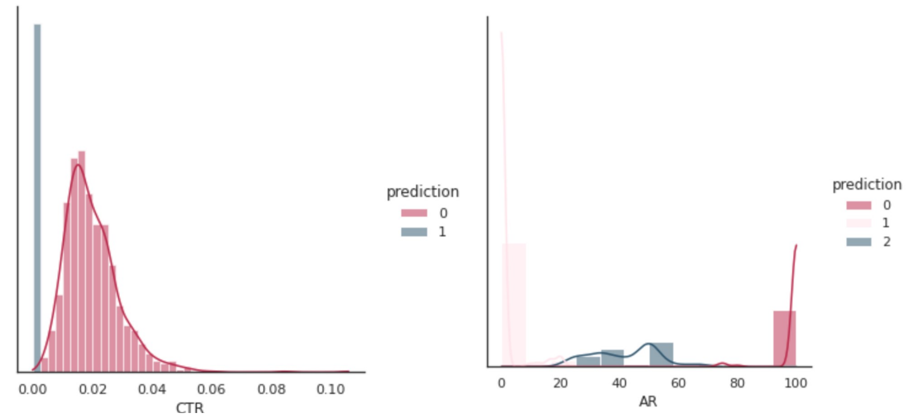


Figure 4.4: The distributions of the clusters when clustering based on (CTR, CPC) and (AR).

depth by breaking them down by customer features. As shown in Figure 4.5, there are differences in TCR and ACR between different age groups. Particularly, the group ”30–34” tends to have a higher TCR and ACR. As a consequence, we should target this age group more. Nevertheless, there are chances that the superiority of ”30-34” over the others is due to chance. Thus, we test our hypothesis by comparing the means of different age groups.

- **Null Hypothesis**: There is no difference in conversion rate among the age group.
- **Alternative Hypothesis**: There is difference in conversion rate among the age group

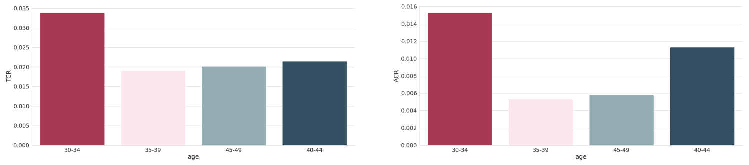


Figure 4.5: The average TCR (Total Conversion Rate) and ACR (Approved Conversion Rate) group by age .

After calculating the p-value based on t-test, we induce that the age group ”30-34” are more likely to be converted statistically.

Chapter 5

Conclusion

Through this project, we successfully installed and deployed Hadoop cluster as well as using Spark to process data.

In this project, we suggest solutions to optimize the return values and conversion rate:

- Get the best value for Spent by using a virtual tracker and derivative.
- Focus on the cluster of audience that have higher Click through Rate and Approval Rate.
- The age group "30-34" are more likely to be converted from Impressions.

Bibliography

[1] Apache Software Foundation. (2010). Hadoop. Retrieved from https://hadoop.apache.org

[2] SSH connection from https://www.ucl.ac.uk/isd/what-ssh-and-how-do-i-use-it

[3] Hadoop Cluster Setup from https://www.linode.com/docs/guides/how-to-install-and-set-up-hadoop-cluster/

[4] Running Spark on YARN from https://spark.apache.org/docs/latest/running-on-yarn.html

[5] Apache Software Foundation. (n.d.). Spark SQL and DataFrames - Spark 3.2.0 Documentation., from https://spark.apache.org/docs/latest/sql-

[6] Kluyver, T., Ragan-Kelley, B., Fernando Px27;erez, Granger, B., Bussonnier, M., Frederic, J., ... Willing, C. (2016). Jupyter Notebooks – a publishing format for reproducible computational workflows. In F. Loizides B. Schmidt (Eds.), Positioning and Power in Academic Publishing: Players, Agents and Agendas (pp. 87–90).

[7] Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . others. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825–2830.

[8] Shafranovich, Y. (2005). Common format and MIME type for comma- separated values (CSV) files.

[9] https://spark.apache.org/docs/latest/ml-guide.html

[10] https://spark.apache.org/docs/latest/rdd-programming-guide.html

[11] https://www.kaggle.com/code/chrisbow/an-introduction-to-facebook-ad-analysis-using-r

[12] https://www.kaggle.com/code/mansimeena/facebook-ad-campaigns-analysis-sales-prediction

[13] https://www.kaggle.com/code/chiticariucristian/clustering-facebooks-ads-campaigns