Proposal for the Starbuck‘s project

1. Domain Background

The domain addressed in this proposal is clearly a marketing/sales domain of a consumer-oriented fortune 500 company (rank 125) Starbucks | 2021 Fortune 500 | Fortune. Starbucks makes most of its earnings by selling coffee or food, and hast 30,000 stores in over 80 markets, as can be also seen in the link.

2. Problem Statement

The problem at hand is bigger than described in the data sets made available by Starbuck's. The overall problem is, **to identify how can we get the maximum revenue of each individual or groups of individuals, while finding more characteristics of these individuals to address in the future more of these individuals or groups of individuals.**

This bigger problem can then be split into several sub problems, where each sub-problem may be challenged through a method of its own. Sub problems may be described as part of the overall problem:

(a) *Maximizing the revenue for individuals:*
   i. which group of individuals (**or individual**, in short **oI**) spend(s) overall more money, when achieving/getting regular incentives
   ii. which group of individuals (**oI**) spends less, while getting regular incentives (and should probably not targeted at all)
   iii. Can we increase spendings of individuals by sending incentives more often?
   iv. ...

(b) *Characteristics of individuals:*
   i. which group of individuals (oI) is attracted by which incentive
   ii. how much money does every group spend
   iii. How long does a group take to redeem an incentive
   iv. Does a group redeem an incentive at all, and how can we probably encourage them?
   v. ...

3. Data sets and inputs

The data sets and inputs are the three tables delivered by Starbuck's

(a) portfolio → **descriptive information about the incentive:** - id (string) - offer id - offer_type (string) - type of offer ie BOGO, discount, informationaldifficulty (int) - minimum - required spend to complete an offer - reward (int) - reward given for completing an offer - duration (int) - time for offer to be open, in days - channels (list of strings)

(b) profile → **demographic information about the customer:**
- age (int) - age of the customer - became_member_on (int) - date when customer created an app account - gender (str) - gender of the customer ('O' for other) - id (str) - customer id - income (float) - customer's income

(c) transcript → **descriptive information about a customers transactions and incentives:** - event (str) - record description (ie transaction, offer received, offer viewed, etc.) - person (str) - customer id - time (int) - time in hours since start of test. The data begins at time t=0 - value - (dict of strings) - either an offer id or transaction amount

4. Solution Statement

To tackle the problems stated above, one would require to use several methods and combine all of them to get a final insight:

(a) Unsupervised learning to look for similarities between customers (e.g. HDSBSCAN, kmeans or other), which use a specific incentive

(b) Supervised ML-Models like Catboost or ExplainableBoostingMachine, which are the latest GradientBoosting/GA2M models, to forecast the use of an coupon or the time until it is redeemed. In addition, one have to decide, if the problem would be a regression and/or classification problem. Catboost would be good for categorical features like the coupon, while EBM offers the highest intelligibilty because its final model is a fancy linear regression containing interaction effects

(c) Hint: All of the previous things can be achieved with and without AWS

5. **Benchmark Model**

When it comes to forecast the:

(a) use of a coupon with yes/no, a simple model would be a Decision Tree

(b) probability of using a specific coupon, it could be a **Multiclass One vs Rest Logistic Regression** (there may be others)

(c) spendings maybe a **Linear Regression**, while the latter can get further simplified by just passing it the mean of a time period or the whole training data.

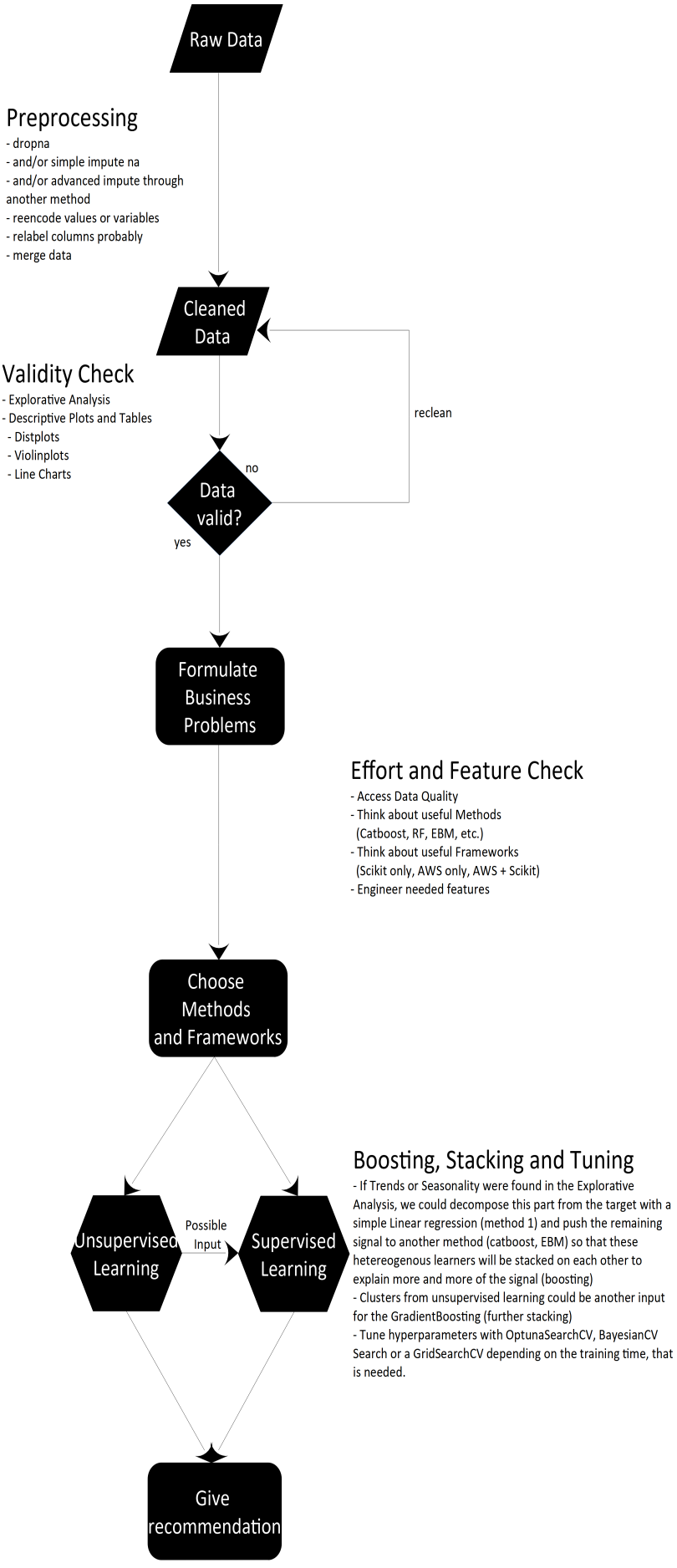
6. **Evaluation Metrics**

The author considers to use micro technical metrics and business metrics to judge the final outcome. A macro evaluation will be made on, how well methods may work together and capture specific aspects of the problem, to be a decision support system for Starbucks.

Micro					Macro
regression	R^2	MAPE	MAE	RMSE	How well the methods can capture problem intrinsic behavior and act well together to a final decision support system.
classification	Accuracy	Confusion Matrix	ROC/AUC	Lift-Chart	
Business	Shapley-Values	Permutation Importance	Recommendation rules		

7. **Project Design**

On the third page you will find the final project design outlined. Remarks: The before-mentioned aspects highlighted different aspects that could be undertaken to fulfill the capstones requirement. The author of the study has only a rough plan of **what he wants do** and reserves himself the right to reduce the effort, for example only doing a classification in AWS instead of doing unsupervised and/or an extra classifier, and go extra routes or make lesser checks than highlighted in the document. I did not formulated extra rounds into the following picture/graph as it is normally the case, when you are done with an ML model to realize it wasn't the one you needed. I also skipped for now the part to reshape the data into a time-series, but as I said, this maybe also an option, as I do not know if I have the time for all of this.


Raw Data
Preprocessing
- dropna
- and/or simple impute na
- and/or advanced impute through another method
- reencode values or variables
- relabel columns probably
- merge data
Cleaned Data
Validity Check
- Explorative Analysis
- Descriptive Plots and Tables
- Distplots
- Violinplots
- Line Charts
reclean
no
Data valid?
yes
Formulate Business Problems
Effort and Feature Check
- Access Data Quality
- Think about useful Methods (Catboost, RF, EBM, etc.)
- Think about useful Frameworks (Scikit only, AWS only, AWS + Scikit)
- Engineer needed features
Choose Methods and Frameworks
Unsupervised Learning
Possible Input
Supervised Learning
Boosting, Stacking and Tuning
- If Trends or Seasonality were found in the Explorative Analysis, we could decompose this part from the target with a simple Linear regression (method 1) and push the remaining signal to another method (catboost, EBM) so that these hetereogenous learners will be stacked on each other to explain more and more of the signal (boosting)
- Clusters from unsupervised learning could be another input for the GradientBoosting (further stacking)
- Tune hyperparameters with OptunaSearchCV, BayesianCV Search or a GridSearchCV depending on the training time, that is needed.
Give recommendation