OVERALL

This support document file contains three main sections:

- Data cleaning process with respective codes and conclusion (carried from Data at Scale coursework)
- Model implementation instructions
- Figure creation notes

DATA CLEANING

The process first started with checking on 'null' and 'duplicated' value in the database.
Python (Pandas) was used to provide quick overview of the data (attached 'D@S – Data Cleaning' file) along with SQL for further diagnosis. Based on that:

- **Customer database:** 2556/2920 (~88%) 'null' value of customer first name ('first') and date of birth ('dob'). Due to that, no direct analysis on customer profile was included in main report.
- **Receipts database:** no 'null' or 'duplicated' value found. All customer IDs and store codes recorded matched with Customer and Store database.
  However, there were 141 pairs of transaction that having different receipt ID but same customer ID, purchased date, value and quantity. These receipts were considered duplicate and removed from clean database (138 lines impacted).
- **Receipt lines database:** no 'null' or 'duplicated' value found. All receipt IDs and product codes from this database matched with Receipt and Product database.
  Follow above assumption, all receipt lines from 138 duplicate receipt ID were removed from analysis (513 lines impacted)

SQL queries for clean receipt database:

- Compute receipt table added with value and quantity for each receipt
- Compute duplicate table
- Create new database excluding the duplicate receipts

```
CREATE TABLE ml2.receipts_clean AS(
WITH
receipt_check AS (
        SELECT receipt_id, purchased_at, store_code, till_number,
        customer_id, SUM(value) AS value,
        SUM(qty) AS qty
        FROM ml2.receipts a JOIN ml2.receipt_lines
        USING (receipt_id)
        GROUP BY receipt_id),
duplicate AS (
        SELECT a.receipt_id AS rep_1, b.receipt_id AS rep_2
        FROM receipt_check a, receipt_check b
        WHERE a.purchased_at=b.purchased_at
        AND a.customer_id=b.customer_id
        AND a.receipt_id<b.receipt_id
        AND a.value=b.value
        AND a.qty=b.qty)
SELECT * FROM ml2.receipts WHERE receipt_id NOT IN (
SELECT rep_1 FROM duplicate))
```

SQL queries for clean receipt_line database:

- Compute receipt table added with value and quantity for each receipt
- Compute duplicate table
- Create new database excluding the duplicate receipts

```
CREATE TABLE ml2.receipt_lines_clean AS(
WITH
receipt_check AS (
        SELECT receipt_id, purchased_at, store_code, till_number,
        customer_id, SUM(value) AS value,
        SUM(qty) AS qty
        FROM ml2.receipts a JOIN ml2.receipt_lines
        USING (receipt_id)
        GROUP BY receipt_id),
duplicate AS (
        SELECT a.receipt_id AS rep_1, b.receipt_id AS rep_2
        FROM receipt_check a, receipt_check b
        WHERE a.purchased_at=b.purchased_at
        AND a.customer_id=b.customer_id
        AND a.receipt_id<b.receipt_id
        AND a.value=b.value
        AND a.qty=b.qty)
SELECT * FROM ml2.receipt_lines WHERE receipt_id NOT IN (
SELECT rep_1 FROM duplicate))
```

MODEL IMPLEMENTATION INSTRUCTION

Model evaluation and implementation includes 5 Jupyter notebooks:

- **MLPA - ConsultingCorp_analysis:** the file includes provided codes for Consulting Corp report. Some amendments were made including:
  - Replace 'quantile' with 'percentile_cont' function due to running error issue
  - Change charts' colors to fit with report theme
  - Add in cumulative % of sales contribution by customers with different average gap between visits
- **MPLA – Finding optimal output:** the file includes steps in evaluating appropriate decrease level in visit frequency to be labelled as churn. Similar process was made for spending decrease to conclude that spending was not an optimal option
- **MPLA – Final evaluation code:** main file of the report, including feature generation & selection, model evaluation and insights report
- **MPLA – Model implementation:** apply the final model to dataset – with instruction to rerun weekly

Each file contains detailed instruction for each step.

FIGURE CREATION NOTES

Figure	Sources
Figure 1. Cumulative % of customers based on avg. gap of visit by ConsultingCorp	Juyper notebook: MPLA – ConsultingCorp_analysis (seaborn)
Figure 2. % of expected churn with different gap definition by ConsultingCorp	Juyper notebook: MPLA – ConsultingCorp_analysis (seaborn)
Figure 3. Cumulative % of sale based on avg. gap of visit	Juyper notebook: MPLA – ConsultingCorp_analysis (seaborn)
Figure 5. Churn rate and model AUC changes with % of normal visit habit	Juyper notebook: MPLA – Finding optimal output (matplotlib)
Figure 6. Correlation between input and output feature	Juyper notebook: MPLA – Final evaluation code (seaborn)
Table 8. Accuracy and AUC on different feature sets	Juyper notebook: MPLA – Final evaluation code (sklearn)
Table 10. Model evaluation results	Juyper notebook: MPLA – Final evaluation code (sklearn)
Figure 11. Receiver operating characteristic curve (ROC) between models in test set	Juyper notebook: MPLA – Final evaluation code (sklearn)
Figure 12. Accuracy and AUC by tuned RF in 'all' and 'window aggerates' sets	Juyper notebook: MPLA – Final evaluation code (sklearn)
Table 13. Churner and Non-churners pen portraits	Juyper notebook: MPLA – Final evaluation code (pandas groupby)
Figure 14. Feature importance in test set by SHAP	Juyper notebook: MPLA – Final evaluation code (shap)
Figure 15. Features' average value (standardised) in churned and non-churned groups	Juyper notebook: MPLA – Final evaluation code (pandas groupby & plotly)
Table 17. Key stats of churned ranking group	Juyper notebook: MPLA – Final evaluation code (pandas groupby)
Figure 16, 18, 19, 20. Average values or churned ranking group	Juyper notebook: MPLA – Final evaluation code (seaborn)