Configuration Manual

MSc Research Project
Fintech

Ruchik More
Student ID: x20234074

School of Computing
National College of Ireland

Supervisor: Mr. Victor Del Rosal

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Ruchik More		
Student ID:	x20234074		
Programme:	MSc Fintech	**Year:**	**2022-2023**
Module:	Research Project		
Lecturer:	Mr. Victor Del Rosal		
Submission Due Date:	8/14/2023		
Project Title:	Understanding Factors Influencing UPI (Unified Payments Interface) user adoption levels and sentiments in India.		

Word Count:5071 **Page Count:** 18

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Ruchik More

Date: 8/14/2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□x
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□x
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□x

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Ruchik More
Student ID: x20234074

1 Introduction

My Research work is submitted as a part of the Msc Fintech module along with this Configuration Manual. The Configuration Manual comprises of the steps that were implemented during the study and the technology and hardware configuration of the machine used. In addition to that, it aims to provide guidance for future research and instruct other researchers about how to reproduce the findings.

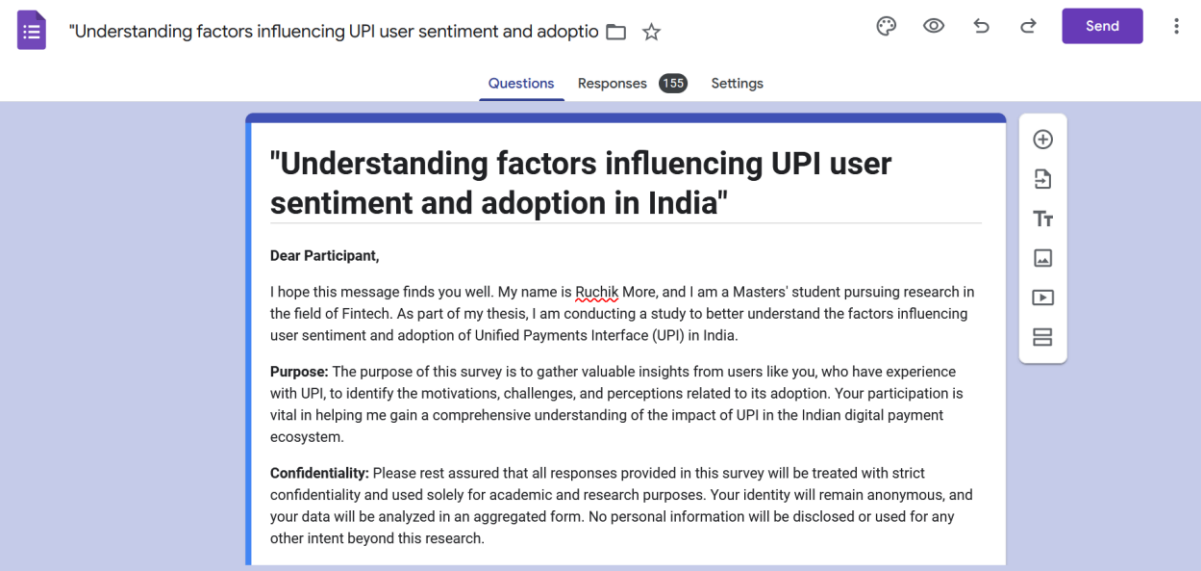
2 System Specification

2.1 Hardware Specifications

OS Name: Microsoft Windows 11 Home Single Language
Processor: AMD Ryzen 5 5500U with Radeon Graphics, 2100 Mhz, 6 Core(s), 12 Logical Processor(s)
(RAM): 16.0 GB
Size: 476.02 GB (511,125,221,376 bytes)

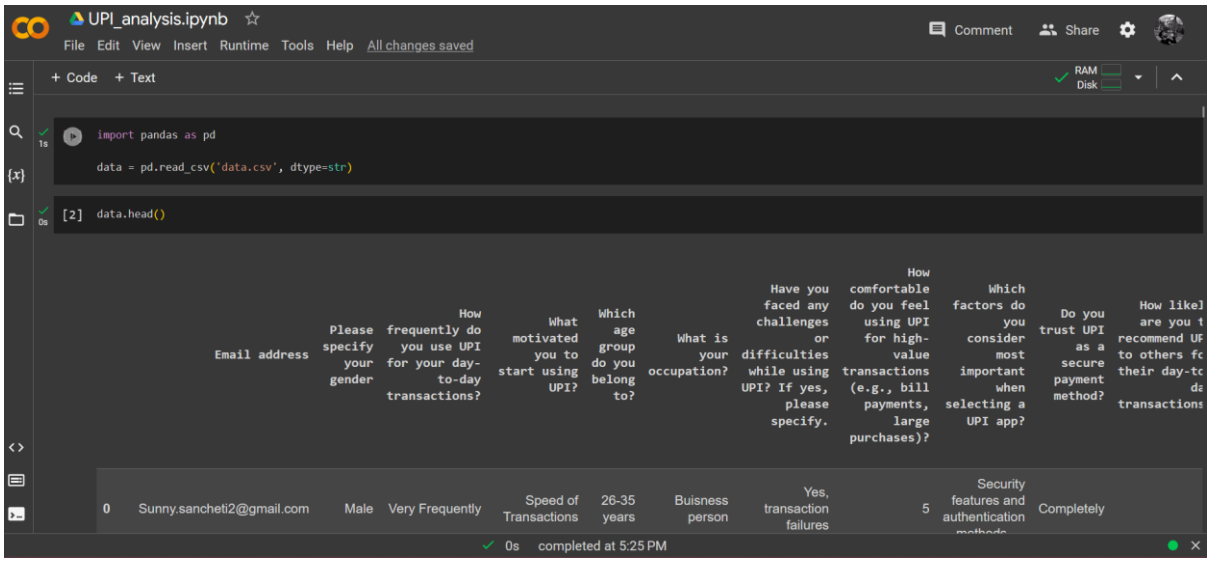
2.2 Software and Tools

Google Forms: The study built a structured Google form which was given out to the respondents to collect the data for the study. The gathered data was saved on a google spread sheet and then exported to a password protected Excel csv file.



Microsoft Excel 2021: Microsoft Excel was used to extract and save the data from Google sheets to a comma separated file (csv).

Google Colaboratory: Google Colaboratory was used as an environment to run Python on the statistical data generated from Google forms and saved in the csv.

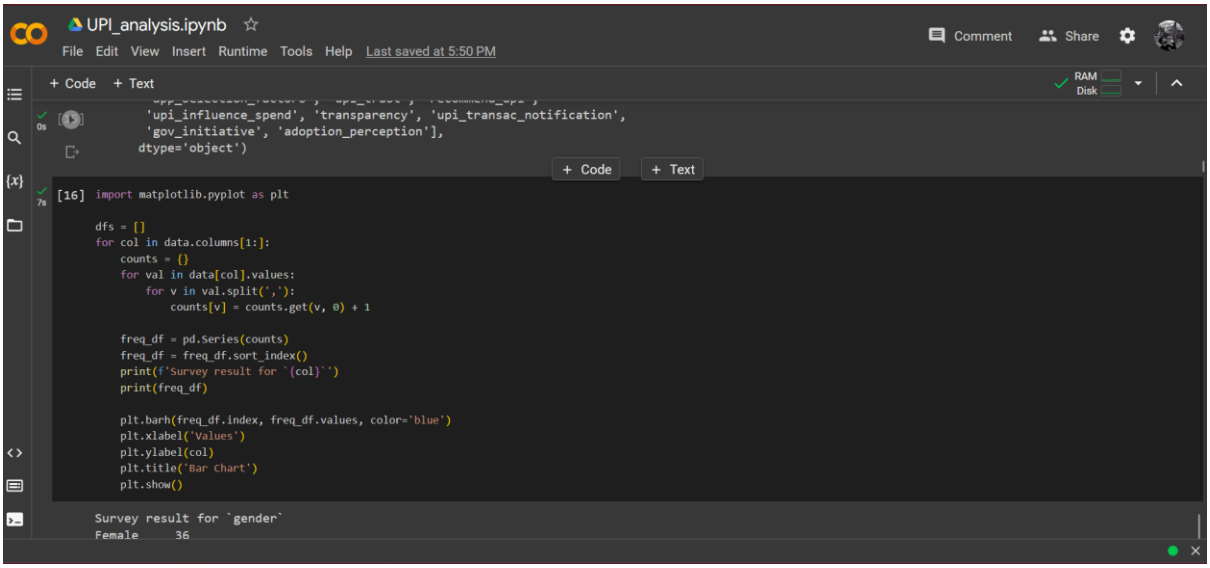


3 Data Cleaning and Pre Processing

In this stage, after the data being imported from the Google sheets, data was cleaned to omit incomplete or erroneously filled responses. Any potential duplicate responses were tackled and dealt with. The columns imported from the Google forms were renamed to a more readable and relevant format. Also, in cases of multiple responses being recorded for particular questions, comma separated values were dealt with to be coherent and one which make sense.

4 Descriptive Statistics was used to explain the participant's responses on the research.

For the same, matplotlib.pyplot was imported from the Python libraries. It was then followed by computing the frequency of each answer choice and then visualising these frequencies using horizontal bar charts.



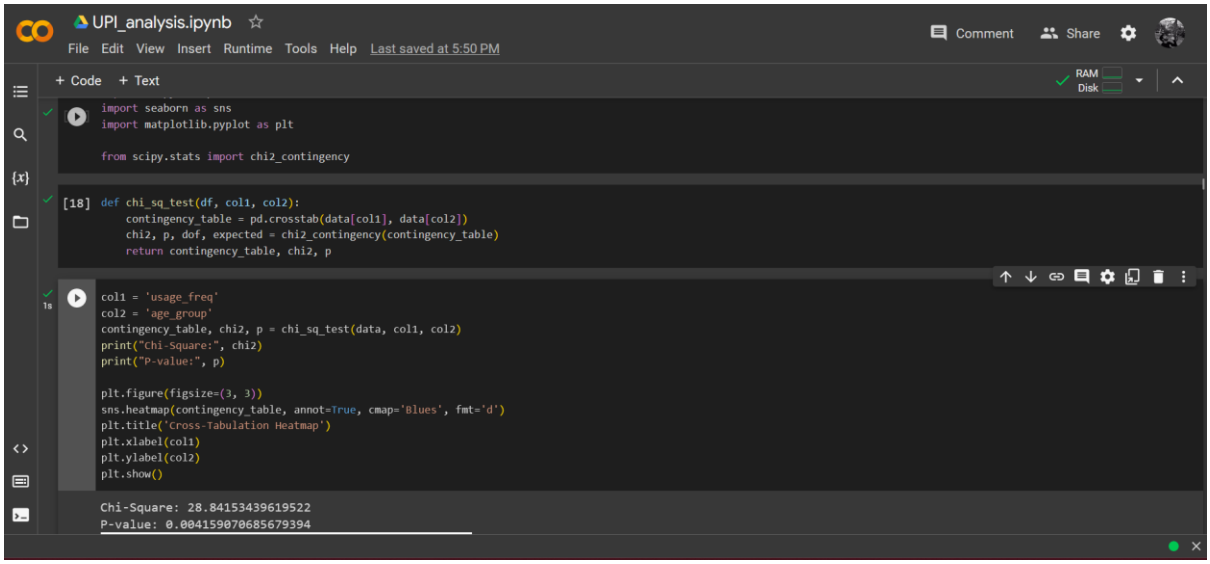
5 Test of Significance

Test of significance is carried out to find the chi square values and the p values which are plotted against every indeoendent variable and each of the identified dependent variable, viz. ‘usage_freq’ which gives the usage frequency and ‘recommend_upi’ for recommending upi to others. The values which thus are calculated shed light on thr correlation of every independent variabkle with the defined dependent variable.

For instance, the code performs a chi-square test on two categorical variables from a DataFrame, col1 which is usage_freq and col2 which is age_group, to determine their independence. It is then followed by the following steps:

1. Importing necessary libraries: numpy, seaborn, matplotlib, and scipy.stats.
2. Defining a function, chi_sq_test, to compute the chi-square test using a contingency table.
3. Applying the chi-square test to the data columns usage_freq and age_group.
4. Printing the chi-square statistic and the p-value.
5. Visualising the contingency table using a heatmap with Seaborn.

The main goal is to establish if there's a significant correlation between usage_freq , (which for our study purpose is the dependent variable) and age_group (which is the independent variable). The process is then repeated for every independent variable. Similarly, recommend_upi is our next dependent variable, which captures the sentiment of the upi user. It then follows the similar process against each independent variable.



6 Decision Trees Classifier

A decision tree classifier is used to classify the data on the given dependent variable's parameters and is classed against all the responses, which in our case are 155 responses from the Google form.

For the Decision Tress classifier to be implemented, the following steps are followed:

1. Necessary libraries such as pandas, scikit-learn, and numpy are imported.
2. A subset of the data, excluding the first column, is selected for further processing.
3. The target variable for the decision tree classification is identified, which in our case is 'usage_freq'.
4. The LabelEncoder from scikit-learn is used to convert categorical variables in the dataset to numerical form, ensuring the target column isn't encoded.
5. The data is then split into feature variables (X) and the target variable (y).
6. Using the train_test_split function, the dataset is divided into training and test subsets. 20% of the data is reserved for testing. We have deployed the 80-20 methodology.
7. A Decision Tree classifier object is initialized with a specified random state for reproducibility.
8. The classifier is trained using the training data subset.
9. Predictions are made on the test data using the trained classifier.
10. Finally, the classification_report function evaluates and prints the classifier's performance on the test set, detailing metrics like precision, recall, and F1-score, where the usually the F1-score is taken into consideration for understanding how good the model fits on the data. The higher the F1 Score, the better it is.
11. The same process is repeated for the other dependent variable 'recommend_upi' to get its own F1-score.

UPI_analysis.ipynb

File Edit View Insert Runtime Tools Help Last saved at 5:50 PM

Comment Share

+ Code + Text

RAM Disk

[45]
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import LabelEncoder

data_for_tree = data[data.columns[1:]].copy()
```

[46]
```
data_for_tree.head()
```

	gender	usage_freq	motivation	age_group	occupation	upi_usage_challenges	comfort_high_val_transaction	app_selection_factors
0	Male	Very Frequently	Speed of Transactions	26-35 years	Business person	transaction failures	5	Security features and authentication methods
1	Female	Regularly	Convenience	26-35 years	Working professional	technical issues	4	User interface and ease of use
2	Female	Very Frequently	Security,Speed of Transactions,Convenience,Gov...	26-35 years	Working professional	I have not faced any challenges	5	Security features and authentication methods
								Security features and

UPI_analysis.ipynb

File Edit View Insert Runtime Tools Help Last saved at 5:50 PM

Comment Share

+ Code + Text

RAM Disk

[47]
```
# Label encode categorical variables
target_column = 'usage_freq'

label_encoder = LabelEncoder()
for column in data_for_tree.columns:
    if column == target_column: continue
    data_for_tree[column] = label_encoder.fit_transform(data_for_tree[column])

# Split data into features (X) and target (y)
X = data_for_tree.drop(columns=[target_column])
y = data_for_tree[target_column]
```

[48]
```
X
```

	gender	motivation	age_group	occupation	upi_usage_challenges	comfort_high_val_transaction	app_selection_factors	upi_trust	recommend_upi	upi_
0	1	17	1	0	16	4	12	0	4	
1	0	0	1	4	10	3	19	2	4	
2	0	15	1	4	0	4	15	0	4	
3	1	13	1	4	10	4	15	0	4	

UPI_analysis.ipynb

File Edit View Insert Runtime Tools Help Last saved at 5:50 PM

Comment Share

+ Code + Text

RAM Disk

```
# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

[51]
```
# Decision Tree classifier
tree_classifier = DecisionTreeClassifier(random_state=42)

# Training the classifier on the training data
tree_classifier.fit(X_train, y_train)
```

```
DecisionTreeClassifier
DecisionTreeClassifier(random_state=42)
```

[52]
```
from sklearn.metrics import classification_report

print(classification_report(y_test, tree_classifier.predict(X_test)))
```

```
                    precision    recall  f1-score   support

      Occasionally       0.00      0.00      0.00         3
Rarely or Never          0.00      0.00      0.00         2
       Regularly         0.36      0.42      0.38        12
```

7 Conclusion:

The configuration manual was used to describe key technologies that were used for the purpose of this research. It discusses how the data was collected, cleaned, processed and then analysed. This manual contains necessary information that can help replicate this study.