

Data Cleaning and Predictive Analysis of eSIM Adoption Survey Data

Author: Collins Mello

1. Introduction

In today's evolving mobile telecommunications landscape, eSIM (embedded SIM) technology represents a significant shift from traditional physical SIM cards. My original thesis explored user perceptions, adoption readiness, and potential challenges associated with transitioning to eSIM technology, with a particular focus on security, convenience, and fraud concerns.

This report documents the post-thesis data science work I conducted on the survey data collected. Using both SQL and Python (Jupyter Notebook), I cleaned, prepared, and analyzed the data, culminating in a predictive model to estimate a user's likelihood of switching to eSIM technology.

2. Dataset Description

The dataset was derived from a structured questionnaire distributed to participants across various countries, primarily in Southern Africa, Cyprus, and Turkey. In total, 64 valid responses were processed for this analysis. The key variables in the dataset include:

- Country, Area, City
- Familiarity with eSIM technology (q1_familiarity)
- Importance of Security (q2_security_importance)
- Likelihood to Switch to eSIM (q3_likelihood_to_switch)
- Cost Advantage Perception (q5_lower_cost)
- Fraud Experience (q7_fraud_affected)
- Perceived Convenience (q6_more_convenient)

Most variables were collected on categorical or binary scales and later transformed into numeric format for analysis.

3. 1 Raw data.

3.2 Results of clean data

[illegible]

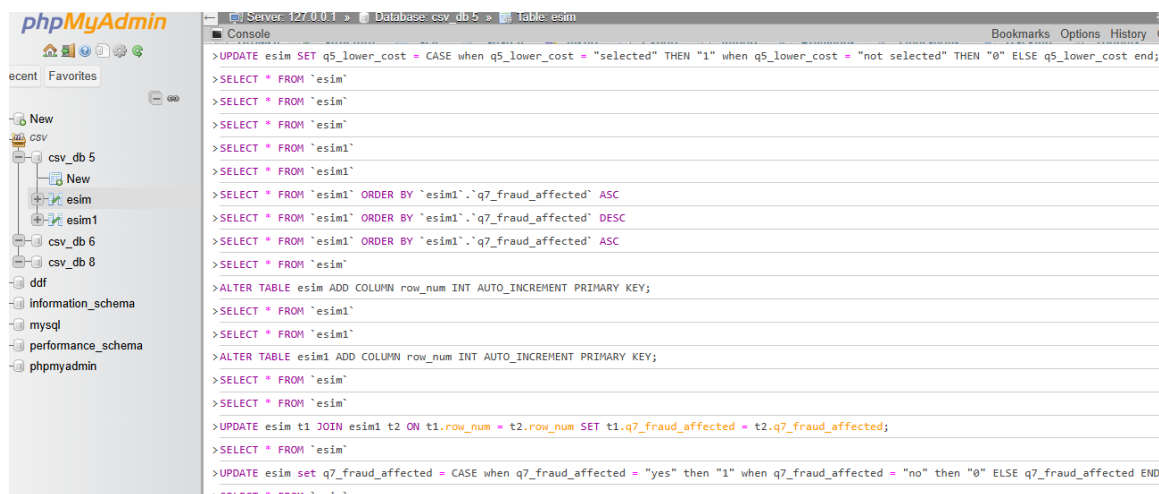
3.1 SQL Cleaning

The raw survey data initially contained various inconsistencies, such as mixed data types (text, numeric), variations in categorical responses, and missing binary encodings. To address these issues, I imported the data into a MySQL database and applied a series of SQL transformation queries. Key cleaning steps included:

- Converting binary fields from text ("selected", "not selected") into numeric values (1 and 0).
- Standardizing responses like "yes"/"no" into 1/0 for the q7_fraud_affected column.
- Handling non-numeric categorical answers in q1_familiarity and q3_likelihood_to_switch, converting phrases such as "somewhat familiar" and "not sure" into numeric 0/1 representations.
- Synchronizing multiple data tables (e.g., esim and esim1) by introducing an auto-incremented primary key (row_num) for reliable joining and merging of records.

Example SQL statement used:

```
UPDATE esim SET q5_lower_cost = CASE  
  WHEN q5_lower_cost = "selected" THEN "1"  
  WHEN q5_lower_cost = "not selected" THEN "0"  
  ELSE q5_lower_cost END;
```



3.2 Excel Cleanup

Prior to SQL import, I visually inspected and manually cleaned the dataset in Excel, addressing issues such as completely empty rows, duplicate entries, and obvious data entry mistakes. This step ensured a smooth SQL import and minimized data loss during transformation

4. Data Analysis in Python (Jupyter Notebook)

4.1 Exploratory Data Analysis (EDA)

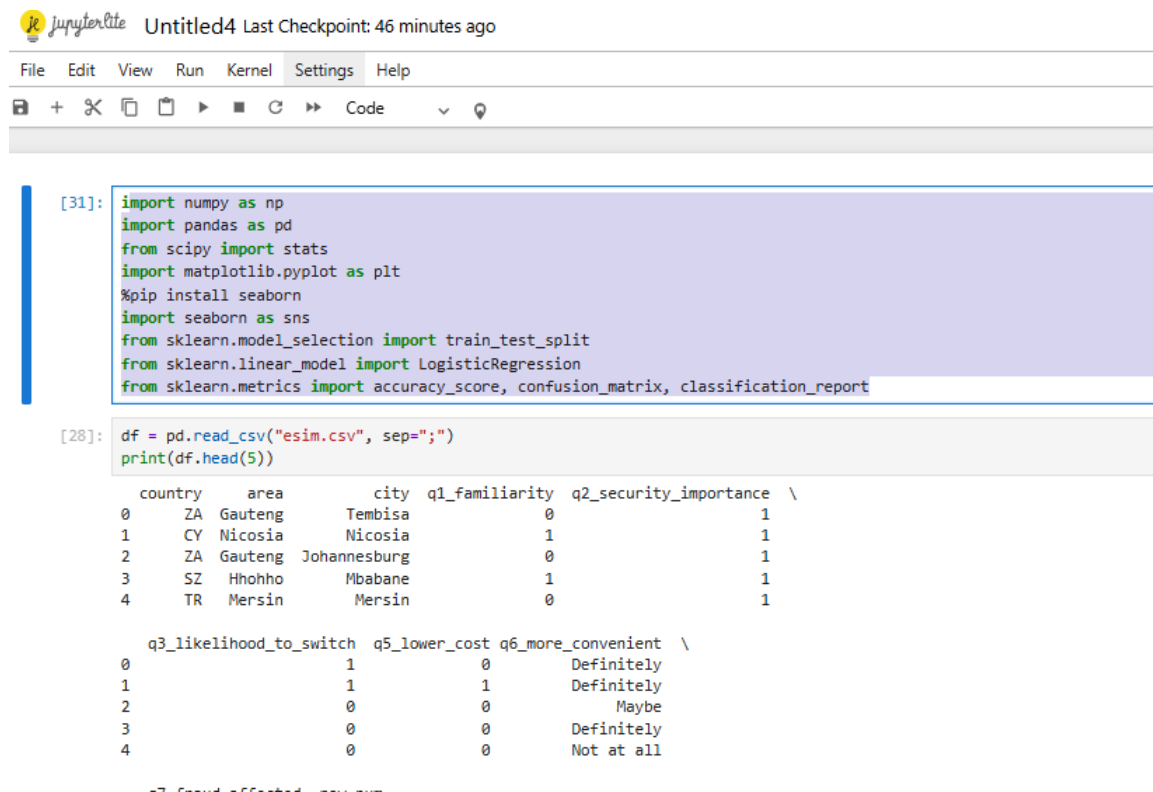
Using pandas and seaborn, I performed initial data exploration to visualize patterns across countries and user responses. Key findings included:

- Significant variation in fraud experiences across different countries.
- High perceived convenience and security importance among most participants.
- Majority of respondents indicated willingness to switch to eSIM technology.

Example visualization code:

```
fraud_counts = df.groupby(['country',  
'q7_fraud_affected']).size().reset_index(name='count')  
sns.barplot(data=fraud_counts, x='country', y='count',  
hue='q7_fraud_affected')
```

Firstly I imported libraries and extracted data



The screenshot shows a Jupyter Notebook window titled 'Untitled4' with a last checkpoint of 46 minutes ago. The interface includes a menu bar (File, Edit, View, Run, Kernel, Settings, Help) and a toolbar with icons for file operations and execution. The code area displays two cells. The first cell, labeled [31], imports various libraries: numpy, pandas, scipy, matplotlib, seaborn, and sklearn. The second cell, labeled [28], reads a CSV file named 'esim.csv' and prints the first five rows of the resulting DataFrame. The output shows columns for country, area, city, familiarity, security importance, likelihood to switch, lower cost, and convenience.

```
[31]: import numpy as np  
import pandas as pd  
from scipy import stats  
import matplotlib.pyplot as plt  
%pip install seaborn  
import seaborn as sns  
from sklearn.model_selection import train_test_split  
from sklearn.linear_model import LogisticRegression  
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

```
[28]: df = pd.read_csv("esim.csv", sep=";")  
print(df.head(5))
```

	country	area	city	q1_familiarity	q2_security_importance	\
0	ZA	Gauteng	Tembisa	0	1	
1	CY	Nicosia	Nicosia	1	1	
2	ZA	Gauteng	Johannesburg	0	1	
3	SZ	Hhohho	Mbabane	1	1	
4	TR	Mersin	Mersin	0	1	

	q3_likelihood_to_switch	q5_lower_cost	q6_more_convenient	\
0	1	0	Definitely	
1	1	1	Definitely	
2	0	0	Maybe	
3	0	0	Definitely	
4	0	0	Not at all	

q7_fraud_affected

I used mode to the most participant country, I used mean to check cheap cost, I used seaborn to show the graph of frauds

```
: #which country appear the most.
country = df["country"]
mode = country.mode()
print("the most participant country is:",mode)

the most participant country is: 0    ZA
Name: country, dtype: object

: #average of people thinking its cheaper
cheap = df["q5_lower_cost"]
mean = np.mean(cheap)
print(" the average of people thinking its cheap:",mean*100)

the average of people thinking its cheap: 23.4375

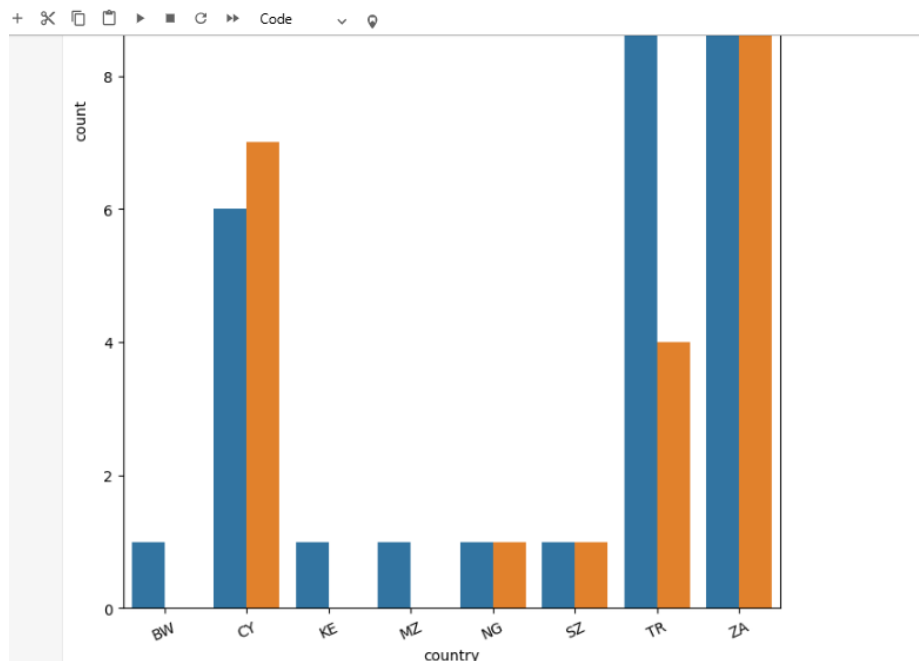
: #graph of frauds based on countries
fraud_count = df.groupby(["country","q7_fraud_affected"]).size().reset_index(name="count")

plt.figure(figsize=(8,12))
sns.barplot(data = fraud_count,x = "country", y = "count",hue="q7_fraud_affected")

plt.title("frauds of countries")
plt.xlabel("country")
plt.ylabel("count")
plt.xticks(rotation=25)
plt.tight_layout
plt.show

: (function matplotlib.pyplot.show(force=None, block=None))
```

I used seaborn to show the graph of frauds



4.2 Predictive Modeling

```
✂  📄  ▶  ■  ↺  ▶▶  Code  ▼  🔍

In [9]: #the prediction of people thinking of switching in the future
#PREPARE DATA

x= df[["q1_familiarity","q2_security_importance"]]
y = df["q3_likelihood_to_switch"]

#SPLIT DATA INTO TRAINING AND TEST SET
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

#Build and train the regression
model=LogisticRegression()
model.fit(x_train,y_train)

#MAKE PREDICTION
y_pred = model.predict(x_test)
#EVALUATE MODEL
accuracy = accuracy_score(y_test, y_pred)

print("accuracy:",accuracy)
print("confusion matrix:",confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))

accuracy: 0.7692307692307693
confusion matrix: [[1 3]
 [0 9]]
Classification Report:
              precision    recall  f1-score   support

     0       1.00      0.25      0.40         4
     1       0.75      1.00      0.86         9
```

The primary modeling goal was to predict whether a user would switch to eSIM (q3_likelihood_to_switch) based on their familiarity (q1_familiarity) and perceived security importance (q2_security_importance).

Using scikit-learn's Logistic Regression model, I trained the model on 80% of the data and tested on 20%:

- Accuracy achieved: ~77%

- Confusion Matrix:

	Actual	Predicted No	Predicted Yes
No	1	3	
Yes	0	9	

Full evaluation report:

Precision (Class 0): 1.00
Recall (Class 0): 0.25
F1-score (Class 0): 0.40
Precision (Class 1): 0.75
Recall (Class 1): 1.00
F1-score (Class 1): 0.86

5. Conclusion

Through a structured SQL cleaning process followed by Python-based data science modeling, I successfully prepared and analyzed my thesis dataset. The predictive model built demonstrated that familiarity with eSIM technology and perceived security importance are strong predictors of users' willingness to switch.

Despite the limited dataset size and class imbalance, the model showed promising results, correctly predicting 77% of cases overall and achieving 100% recall on the switching group.

Acknowledgements

This analysis builds on the foundation of my thesis work completed in 2024 and demonstrates the power of combining data science techniques with traditional academic research.