SRI RAMACHANDRA

INSTITUTE OF HIGHER EDUCATION AND RESEARCH

(Category - I Deemed to be University) Porur, Chennai

SRI RAMACHANDRA FACULTY OF ENGINEERING AND TECHNOLOGY

ADDING ML LAYER TO FREIGHT AUDIT TOOL

INT 291 – INTERNSHIP - 1

PROJECT REPORT

Submitted by

A S SAI THEJASWINI – E5222035

In partial fulfilment for the award of the degree of

BACHELOR OF SCIENCE

in

COMPUTER SCIENCE

(Artificial Intelligence and Data Analytics)

Sri Ramachandra Faculty of Engineering and Technology

Sri Ramachandra Institute of Higher Education and Research, Porur, Chennai -600116

August, 2024

SRI RAMACHANDRA
INSTITUTE OF HIGHER EDUCATION AND RESEARCH
(Category - I Deemed to be University) Porur, Chennai
SRI RAMACHANDRA FACULTY OF ENGINEERING AND TECHNOLOGY

BONAFIDE CERTIFICATE

Certified that this project report "**Adding ML Layer to Freight Audit tool**" is the Bonafide record of work done by **"A S SAI THEJASWINI – E5222035"** who carried out the internship work under my supervision.

Signature of the Supervisor	**Signature of Program coordinator**
Dr. Surya S	Dr. A Christoper Tamilmathi
Assistant Professor	Assistant Professor,
Department of Cyber Security and IoT	Department of Computer Science
Sri Ramachandra Faculty of Engineering and technology	Sri Ramachandra Faculty of Engineering and Technology,
SRIHER, Porur, Chennai-600116	SRIHER, Porur, Chennai-600116

Evaluation Date: 01/08/24

SRI RAMACHANDRA
INSTITUTE OF HIGHER EDUCATION AND RESEARCH
(Category - I Deemed to be University) Porur, Chennai
SRI RAMACHANDRA FACULTY OF ENGINEERING AND TECHNOLOGY

ACKNOWLEDGEMENT

I express my sincere gratitude to our Programme coordinator Dr. A Christoper Tamilmathi for her support and for providing the required facilities for carrying out this study. I wish to thank Dr. Surya S, my faculty supervisor(s) Department of Cybersecurity and IoT, Sri Ramachandra faculty of Engineering and Technology for extending help and encouragement throughout the project. Without his/her continuous guidance and persistent help, this project would not have been a success for me. I am grateful to all the members of Sri Ramachandra Faculty of Engineering and Technology, my beloved parents and friends for extending the support, who helped us to overcome obstacles in the study.

3

TABLE OF CONTENTS

LIST OF IMAGES

ABSTRACT

Freight audit tools are crucial for ensuring accuracy in logistics and supply chain management by validating freight invoices and identifying discrepancies. Traditional freight audit systems, however, often rely heavily on manual processes and rule-based methods, which can be time-consuming and prone to human error. Adding a machine learning (ML) layer to freight audit tools can significantly enhance their efficiency and accuracy. This paper explores the integration of machine learning algorithms into freight audit systems to automate data analysis, detect anomalies, and predict potential discrepancies. By leveraging historical data, the ML layer can learn patterns and improve its decision-making capabilities over time, reducing the need for manual intervention and increasing the speed of audits. The implementation of such a system promises to reduce costs, improve operational efficiency, and provide deeper insights into freight operations, ultimately leading to more informed decision-making and strategic planning in logistics. Through a series of experiments and case studies, this paper demonstrates the potential benefits and challenges of incorporating machine learning into freight audit tools, providing a roadmap for future advancements in the field.

CHAPTER 1

INTRODUCTION

In the dynamic realm of global commerce, efficient management of transportation costs is crucial for businesses aiming to maintain competitiveness and profitability. A pivotal aspect of this management lies in the accurate verification and control of freight invoices. Freight audit tools are advanced software platforms designed to automate the validation of freight invoices against contracted rates, tariffs, and service agreements. Implementing and using a freight audit tool can be complex. It often requires significant training for employees to understand and utilize the software effectively. Implementing automation to handle data entry and analysis, reduces the likelihood of human error and speeding up the auditing process.

1.1. Problem Statement

- The companies like FedEx, claim for the refund from their associated courier service for the Late shipment, Delivery, void, address correlation and duplicate billings, etc.

- The Companies tend to find insights on the late delivery, shipment for their future use and also would like to predict the number of refunds that can be claimed from a set of courier charges provided.

- Outsourcing freight auditing to third-party providers can introduce risks related to trust and control. Companies may face challenges ensuring that the third party adheres to their specific compliance requirements and internal regulations.

1.2 Objective

- To solve the above problem statement, we develop a Machine Learning model that can get trained using the historical data to predict the number of refunds in the future approximately.

- The analysis and model is designed such that it predicts the number of refunds and provide insights based on the months, seasons, cities.

- The model takes all the features applicable and analyzes the future refunds and trends.

- This reduces the time and improves more accuracy in finding insights.

CHAPTER 2

LITERATURE SURVEY

YEAR	WEBSITE	AUTHOR	TITLE	REMARKS
2024	SAFETY CULTURE	LEON ALTOMONTE	A GUIDE TO FREIGHT AUDITS	THE ARTICLE DEFINES THE IMPORTANCE, TOOLS REQUIRED FOR THE CONDUCT OF THE FREIGHT AUDIT TOOL
2019	BARNESANDNOBLE	GERARDUS BLOKDYK	FREIGHT AUDIT A COMPLETE GUIDE	THIS SELF-ASSESSMENT EMPOWERS PEOPLE TO DO FREIGHT AUDIT INVESTMENTS WORK
2024	EBOOKS	ANDRIY	HE	THIS BOOK

		BURKOV	HUNDRED – PAGE MACHINE LEARNING BOOK	GIVES THE SOLID INTRODUCTION TO MACHINE LEARNING. THE BOOK COMBINES BOTH THEORY AND PRACTICE.
2024	RESEARCHPAPERS	SHANNON CORGAN	RATELINX	FOR MOST LOGISTICS PROFESSIONALS , GOING THROUGH STACKS OF FREIGHT INVOICES. THIS GUIDE WILL LOOK AT WHAT A FREIGHT AUDIT, AND PAYMENT SERVICES AVAILABLE.

CHAPTER 3

METHODOLOGY

1. REQUIREMENTS GATHERING:

- Identifying and documenting the needs and expectations of stake holders.
- Clearly define the primary objectives of the freight audit tool.
- Determine whether the organization has the necessary operational capacity and resources to implement and maintain the tool

2. UNDERSTANDING OBJECTIVE:

- Clearly understanding the problem statement and objective
- Understanding the data requirements

3. IMPLEMENTATION:

- Choose a programming language (Python)
- Break down the problem statement into smaller problems
- Develop the model

4. ANALYSIS:

- Train the model using the AI model
- Test the accuracy of the model and analyze the insights

5. REPORT AND DOCUMENTATION:

- Document the analysis and provide the report

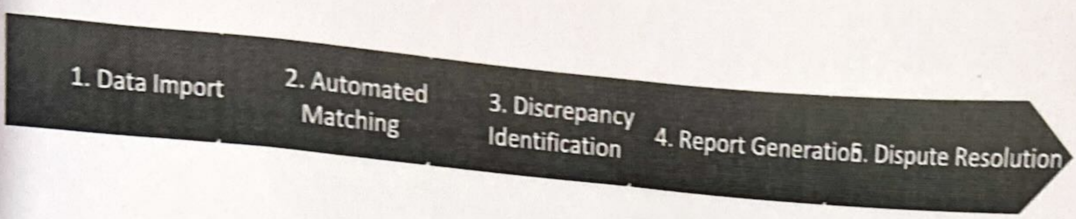
CHAPTER 4

IMPLEMENTATION

FREIGHT AUDITING FOR FINDING REFUNDS CRITERIA:

1. Tracking Orders
2. Shipment Date
3. Estimated Delivery Date
4. Actual Delivery Date
5. Sender Cities
6. Receiver Cities

IMPLEMENTATION STEPS



1. **Data Import:**
   - Import the dataset and prepare the data for the model building.
2. **Automated Matching:**
   - Once the data is well prepared, start with matching the data's
3. **Discrepancy Identification**
   - Identify the discrepancies in dimensions like date, tracking number, cities, Address
4. **Model Building:**
   - Build the model and train using the cleansed dataset
   - Test the accuracy of the model
5. **Report Generation:**
   - Generate the report based on the accuracy, uses and benefits of the model
6. **Dispute Resolution:**
   - Resolve the disputes if any.

CHAPTER 5

CONCLUSION

In conclusion, machine learning technology is reshaping the landscape of freight auditing, offering unprecedented accuracy, efficiency, and strategic insights.

A well-designed freight audit tool offers significant advantages for businesses engaged in logistics and supply chain management. By automating the audit process, it ensures accuracy and efficiency in verifying freight invoices, thereby reducing the risk of overpayments and financial discrepancies. The tool provides comprehensive insights through detailed reporting and analytics, enabling businesses to identify cost-saving opportunities and optimize their shipping strategies. Furthermore, it enhances transparency and accountability by facilitating effective dispute management with carriers. Overall, the adoption of a freight audit tool leads to improved financial control, operational efficiency, and strategic decision-making, ultimately contributing to a more streamlined and cost-effective logistics operation.

.

APPENDICES

1. ML MODEL FOR FINDING THE NO OF REFUNDS

```
import pandas as pd

import numpy as np

import seaborn as sns

mport matplotlib.pyplot as plt

rom sklearn.preprocessing

mport LabelEncoderfrom sklearn.model_selection

nport train_test_splitfrom sklearn.linear_model

nport LinearRegressionfrom sklearn.metrics
```

```
import mean_squared_error, r2_scoreimport joblib

df = pd.read_excel('shipment2024.xlsx')

df.head()

df.tail()

df.info()

df.describe()

sns.histplot(df['Invoice Amount’])

plt.show()

# Convert categorical variables to numerical

label_encoder = LabelEncoder()df['Sender City'] =
label_encoder.fit_transform(df['Sender City’])

df['Sender State'] = label_encoder.fit_transform(df['Sender State’])

df['Receiver City'] = label_encoder.fit_transform(df['Receiver City’]

)df['Receiver State'] = label_encoder.fit_transform(df['Receiver State’])

df['Carrier service type'] = label_encoder.fit_transform(df['Carrier service type’])

df['status'] = label_encoder.fit_transform(df['status’])

df['transportation mode'] = label_encoder.fit_transform(df['transportation mode'])

# Extract features and target variable

X = df[['Sender City', 'Sender State', 'Receiver City', 'Receiver State', 'Carrier
service type', 'status', 'transportation mode’]]
```

```
y = df['Invoice Amount']

# Split data into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the model

model = LinearRegression()model.fit(X_train, y_train)

sety_pred = model.predict(X_test)

# Evaluate the model

mse = mean_squared_error(y_test, y_pred)

r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error: {mse}')

print(f'R-squared: {r2}')
```

SCREENSHOTS OF THE OUTPUT

1

	Recipient Number	Account Number	Sender City	Sender State	Receiver City	Receiver State	Carrier service type	Invoice Amount	status	ship date	delivery date	transportation mode
0	0000447989	0000447989	AUBURN	IN	COCHRAN	GA	Next Day Air Commercial	696.06	Delivered	2024-04-05	2024-04-09 09:30:00	ship
1	0000447989	0000447989	AUBURN	IN	COCHRAN	GA	Delivery Area Surcharge - Extended	696.06	Delivered	2024-04-03	2024-04-11 10:56:00	flight
2	0000447989	0000447989	AUBURN	IN	COCHRAN	GA	Fuel Surcharge	696.06	Delivered	2024-04-03	2024-04-11 10:56:00	flight
3	0000447989	0000447989	AUBURN	IN	COCHRAN	GA	Next Day Air Commercial	696.06	Delivered	2024-04-03	2024-04-11 10:56:00	flight
4	0000447989	0000447989	Newburgh	NY	SAN JUAN	PR	Ground Commercial	696.06	Delivered	2024-04-03	2024-04-11 10:56:00	flight

2

	Recipient Number	Account Number	Sender City	Sender State	Receiver City	Receiver State	Carrier service type	Invoice Amount	status	ship date	delivery date	transportation mode
295	0000A88V64	0000A88V64	3	3	29	15	13	9199.53	1	2024-04-03	2024-04-09 11:44:00	2
296	0000A88V64	0000A88V64	3	3	15	7	13	9199.53	1	2024-04-04	2024-04-09 11:11:00	2
297	0000A88V64	0000A88V64	3	3	15	7	21	9199.53	1	2024-04-03	2024-04-12 12:02:00	2
298	0000A88V64	0000A88V64	3	3	15	7	9	9199.53	1	2024-04-09	2024-04-12 12:21:00	0
299	0000A88V64	0000A88V64	3	3	15	7	13	9199.53	1	2024-04-03	2024-04-12 10:07:00	0

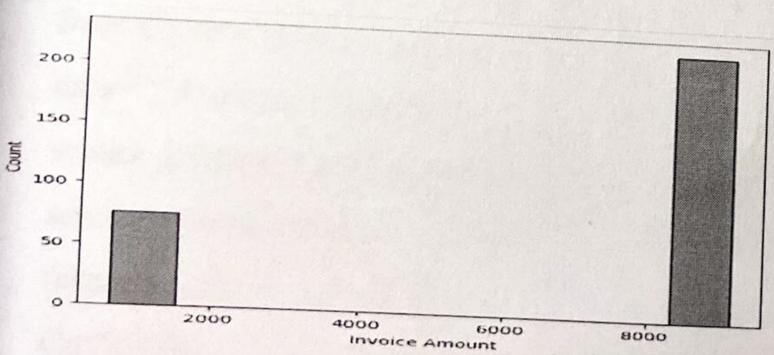
3

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 12 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Recipient Number        300 non-null    object
 1   Account Number          300 non-null    object
 2   Sender City             300 non-null    object
 3   Sender State            300 non-null    object
 4   Receiver City           300 non-null    object
 5   Receiver State          300 non-null    object
 6   Carrier service type    300 non-null    object
 7   Invoice Amount          300 non-null    float64
 8   status                  300 non-null    object
 9   ship date               300 non-null    datetime64[ns]
 10  delivery date           300 non-null    datetime64[ns]
 11  transportation mode     300 non-null    object
dtypes: datetime64[ns](2), float64(1), object(9)
memory usage: 28.3+ KB
```

4

	Invoice Amount	ship date	delivery date
count	300.000000	300	300
mean	7045.317600	2024-04-06 12:14:24	2024-04-10 05:04:55.200000
min	696.060000	2024-03-04 00:00:00	2024-03-05 11:16:00
25%	696.060000	2024-04-04 00:00:00	2024-04-09 10:45:00
50%	9199.530000	2024-04-08 00:00:00	2024-04-10 11:14:00
75%	9199.530000	2024-04-09 00:00:00	2024-04-11 12:03:45
max	9199.530000	2024-04-12 00:00:00	2024-04-13 18:25:00
std	3704.509968	NaN	NaN

5



6

```
▾ LinearRegression
LinearRegression()
```

```
7
```

```
Mean Squared Error: 5093800.815030558
R-squared: 0.6242938373731587
```

2. ML MODEL FOR FINDING THE REPUTED CITIES FOR REFUND AND ADJUSTING THE PROBABILITY OF EACH CITY WITH NEW DATASET

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import mean_squared_error
data = pd.read_csv('internvisuals.csv' ,encoding='latin')
sender_refunds = data['sender City'].value_counts().head(10).reset_index()
sender_refunds.columns = ['city', 'sender_refund_count']
receiver_refunds = data['Receiver
City'].value_counts().head(10).reset_index()
receiver_refunds.columns = ['city', 'receiver_refund_count']
city_refunds = pd.merge(sender_refunds, receiver_refunds, on='city',
how='outer').fillna(0)
```

```
city_refunds['total_refund_count'] =city_refunds['sender_refund_count'] +
city_refunds['receiver_refund_count']

X = city_refunds[['sender_refund_count', 'receiver_refund_count']]
y = city_refunds['total_refund_count']
```

```
sender_cities = city_refunds.sort_values(by='sender_refund_count',
ascending=False).head(10)['city'].tolist()

receiver_cities = city_refunds.sort_values(by='receiver_refund_count',
ascending=False).head(10)['city'].tolist()

print(f'Top 10 sender cities: {sender_cities}')

print(f'Top 10 receiver cities: {receiver_cities}')

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

model = RandomForestClassifier()

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

y_pred

mse = mean_squared_error(y_test, y_pred)

print(f'Mean Squared Error: {mse}')

def update_with_new_data(new_data_path, model):

    new_data = pd.read_csv(new_data_path)

    new_sender_refunds =
new_data['sender_city'].value_counts().head(10).reset_index()

    new_sender_refunds.columns = ['city', 'sender_refund_count']
```

```
new_receiver_refunds =
new_data['receiver_city'].value_counts().head(10).reset_index()
new_receiver_refunds.columns = ['city', 'receiver_refund_count']
new_city_refunds = pd.merge(new_sender_refunds, new_receiver_refunds,
on='city', how='outer').fillna(0)
new_city_refunds['total_refund_count'] =
new_city_refunds['sender_refund_count'] +
new_city_refunds['receiver_refund_count']
new_X = new_city_refunds[['sender_refund_count', 'receiver_refund_count']]
new_city_refunds['predicted_total_refund_count'] = model.predict(new_X)
top_sender_cities = new_city_refunds.sort_values(by='sender_refund_count',
ascending=False).head(10)['city'].tolist()
top_receiver_cities = new_city_refunds.sort_values(by='receiver_refund_count',
ascending=False).head(10)['city'].tolist()
return top_sender_cities, top_receiver_cities
new_data_path = 'new_shipment_data.csv'
top_sender_city, top_receiver_city = update_with_new_data(new_data_path,
model)
print(f' top_sender_city:{top_sender_city}')
print(f' top_receiver_city:{top_receiver_city}')
def plot_top_cities(train, test, column, title):
```

```
train_top_cities = train[column].value_counts().head(10).reset_index()
train_top_cities.columns = ['city', 'refund_count']

test_top_cities = test[column].value_counts().head(10).reset_index()
test_top_cities.columns = ['city', 'refund_count']

# Merge train and test refund counts on city
combined = pd.merge(train_top_cities, test_top_cities, on='city', how='outer',
suffixes=('_train', '_test')).fillna(0)
combined.set_index('city', inplace=True)

# Plotting the combined data
combined.plot(kind='bar', figsize=(12, 8))
plt.title(title)
plt.xlabel('City')
plt.ylabel('Refund Count')
plt.legend(['Training Data', 'Testing Data'])
plt.show()
train_data, test_data = train_test_split(data, test_size=0.2, random_state=42)
```

plot_top_cities(train_data, test_data, 'sender City', 'Top 10 Sender Cities in Training vs Testing Data')

plot_top_cities(train_data, test_data, 'Receiver City', 'Top 10 Receiver Cities in Training vs Testing Data')

SCREENSHOTS OF THE OUTPUT

8

```
Top 10 sender cities: ['GREENSBORO', 'SOUTH HACKENSACK', 'NEW ALBANY', 'Irvine', 'YORK', 'ENGLEWOOD', 'IRVINE', 'PHOENIX', 'PLAINVIEW', 'DALLAS']
Top 10 receiver cities: ['GREENSBORO', 'SOUTH HACKENSACK', 'NEW ALBANY', 'DALLAS', 'INDIANAPOLIS', 'PHOENIX', 'ENGLEWOOD', 'YORK', 'PLAINVIEW', 'ASTORIA']
```

9

```
RandomForestClassifier
RandomForestClassifier()
```

10

```
array([118.,  52.,  30.])
```

11

```
Mean Squared Error: 860.0
```

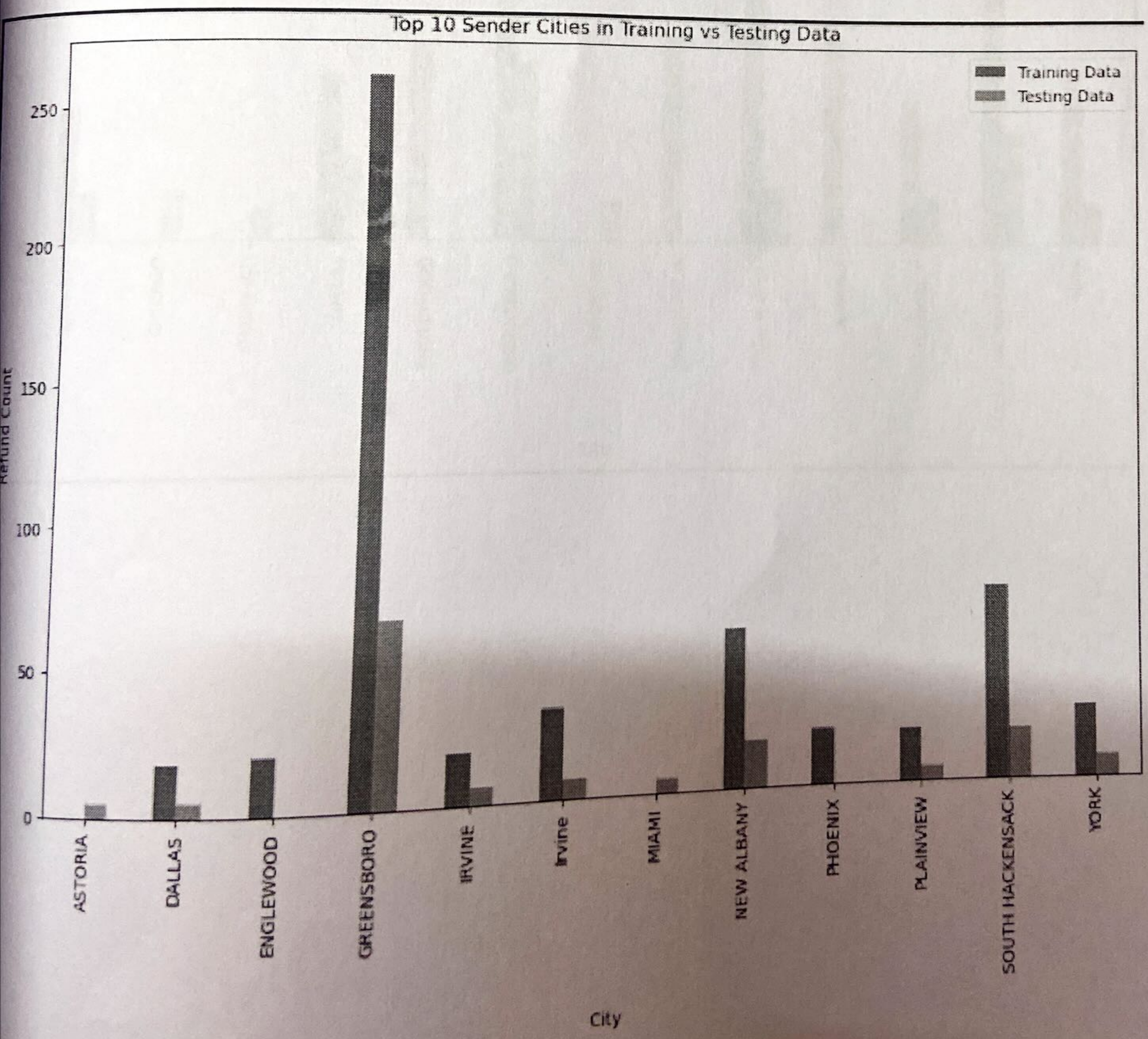
12

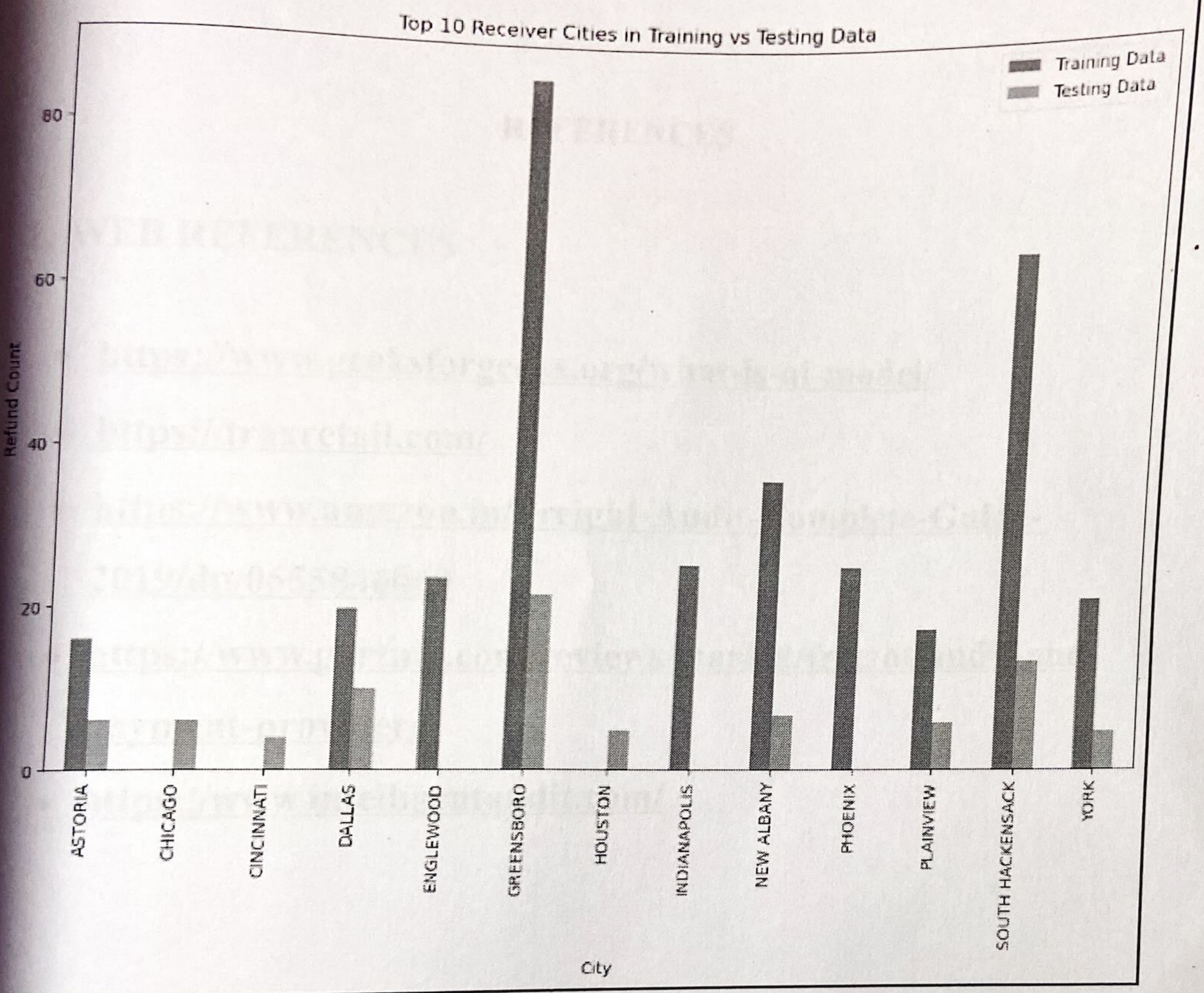
top_sender_city:['Albuquerque', 'Charlotte', 'Minneapolis', 'New York', 'Omaha', 'Louisville', 'Dallas', 'San Jose', 'Las Vegas', 'Raleigh']

13

top_receiver_city:['San Francisco', 'Detroit', 'San Antonio', 'Chicago', 'Colorado Springs', 'Dallas', 'San Jose', 'Long Beach', 'Boston', 'Phoenix']

14




Top 10 Receiver Cities in Training vs Testing Data
Training Data
Testing Data
Refund Count
0
20
40
60
80
ASTORIA
CHICAGO
CINCINNATI
DALLAS
ENGLEWOOD
GREENSBORO
HOUSTON
INDIANAPOLIS
NEW ALBANY
PHOENIX
PLAINVIEW
SOUTH HACKENSACK
YORK
City

REFERENCES

1. WEB REFERENCES

- **https://www.geeksforgeeks.org/what-is-ai-model/**
- **https://traxretail.com/**
- **https://www.amazon.in/Freight-Audit-Complete-Guide-2019/dp/0655846069**
- **https://www.gartner.com/reviews/market/freight-audit-and-payment-providers**
- **https://www.intelligentaudit.com/**

2. JOURNAL REFERENCES

- SAFETY CULTURE - A GUIDE TO FREIGHT AUDITS
- BARNESANDNOBLE - FREIGHT AUDIT A COMPLETE GUIDE
- EBOOKS - HE HUNDRED – PAGE MACHINE LEARNING BOOK
- RESEARCHPAPERS - RATELINX