

# Project Fake-bill Classification

# Fake Invoices:

- **Fake invoice:**

The “Invoices” that are usually treated as ‘fake’ are those where in the GST invoices are raised by an entity without actual supply of goods or services or payment of GST.

**Potential motives for using fake invoices:**

> Inflating turnover for the purpose of:

- a) Availing higher Credit Limit/ Overdraft from Banks
- b) Obtaining bank loans
- c) Improving valuations for IPO or sale of stake
- d) Obtaining contracts including Government contracts

> Booking fake purchases for getting Income-tax benefits by:

- a) Showing reduced profit margins and higher expenses
- b) Avoiding payment of Income-tax by reducing net profit

> Cash generation/ diversion of company funds

> Laundering of money

# Project Implementation Process:

## 1. Setting Up

- [1.1 Introduction](#)
- [1.2 Loading Libraries](#)
- [1.3 Loading Data](#)
- [1.4 Data Distribution](#)
- [1.5 Correlation Between the Data](#)

## 2. Missing Values

- [2.1 Separating Train and Test Splits](#)
- [2.2 Model Evaluation](#)
- [2.3 Validation Predictions Visualization](#)

## 3. Modeling

- [3.1 Naïve Bayes](#)
- [3.2 Bagging Classifier](#)
- [3.3 Logistic Regression](#)
- [3.4 Decision Trees](#)
- [3.5 Random Forest Classifier](#)
- [3.6 svm](#)
- [3.7 K-Nearest Neighbors](#)
- [3.8 Boosting Classifier](#)

## 4. Conclusion about models

## 5. Deployment using streamlit and spyder



-:Detailed Files:-

Data : Fakebill\_classification.csv

EDA, Modeling : Fakebill\_Classification\_project.ipynb

Deployment : fakebill\_deployment\_app.py

## Dataset: Fakebill\_classification.csv

- About the Dataset

This dataset has 1500 rows and 7 columns:

1. **is\_genuine**: Whether the bill is fake or not. (boolean)
2. **diagonal**: diagonal measurements in mm (float)
3. **height\_left**: the height of the left side in mm(float)
4. **height\_right**: the height of the right side in mm (float)
5. **margin\_low**: the lower margin in mm (float)
6. **margin\_up**: the upper margin in mm (float)
7. **length**: the total length in mm (float)

## Fakebill\_classification.csv

	is_genuine	diagonal	height_left	height_right	margin_low	margin_up	length
0	True	171.81	104.86	104.95	4.52	2.89	112.83
1	True	171.46	103.36	103.66	3.77	2.99	113.09
2	True	172.69	104.48	103.50	4.40	2.94	113.16
3	True	171.36	103.91	103.94	3.62	3.01	113.51
4	True	171.73	104.28	103.46	4.04	3.48	112.54
...	...	...	...	...	...	...	...
1495	False	171.75	104.38	104.17	4.42	3.09	111.28
1496	False	172.19	104.63	104.44	5.27	3.37	110.97
1497	False	171.80	104.01	104.12	5.51	3.36	111.95
1498	False	172.06	104.28	104.06	5.17	3.46	112.25
1499	False	171.47	104.15	103.82	4.63	3.37	112.07

1500 rows x 7 columns

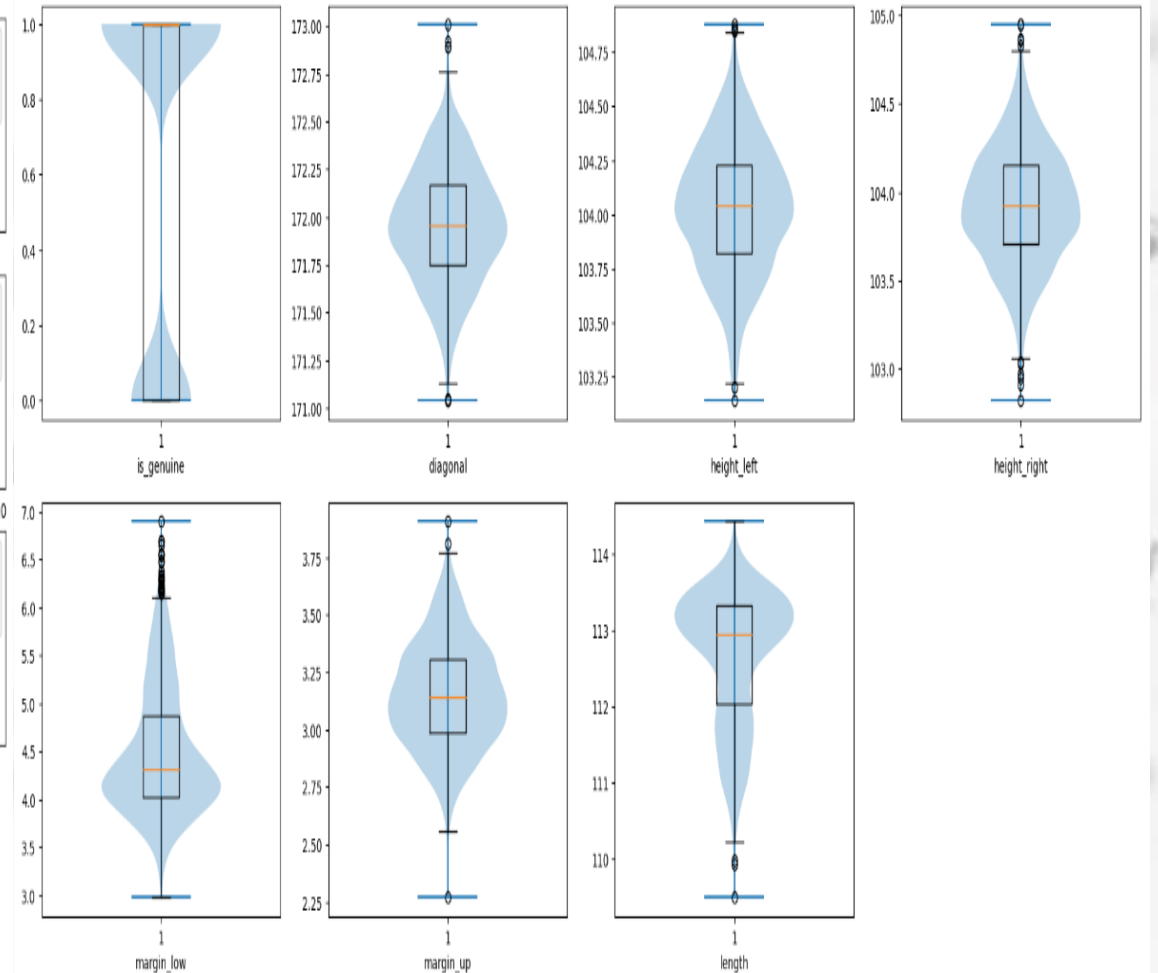
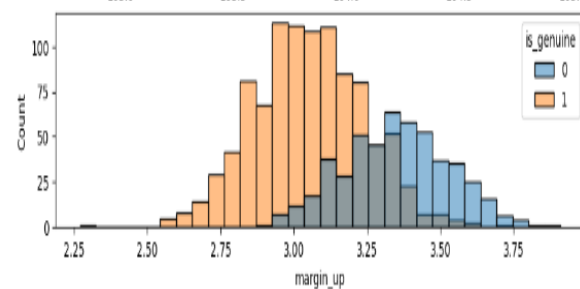
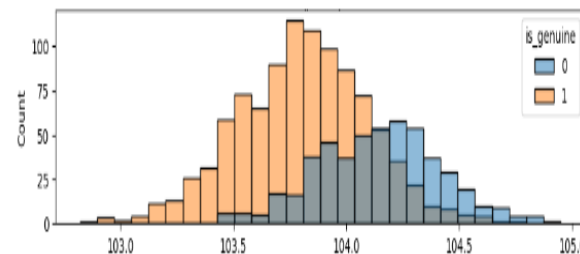
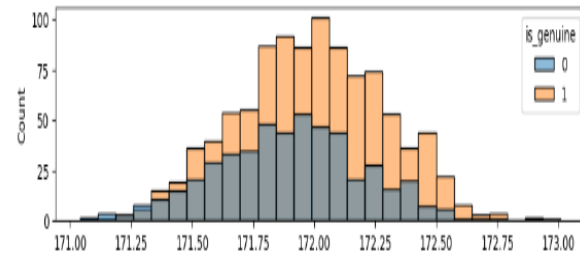
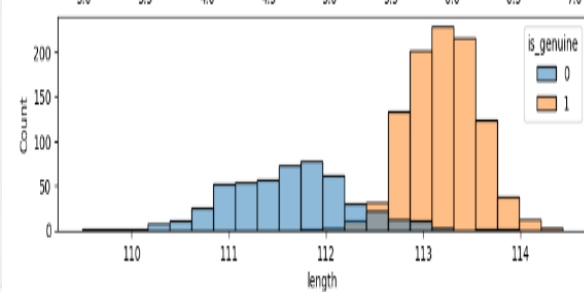
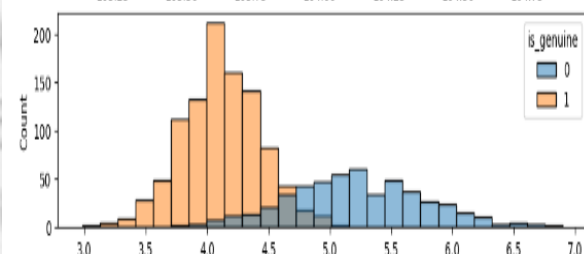
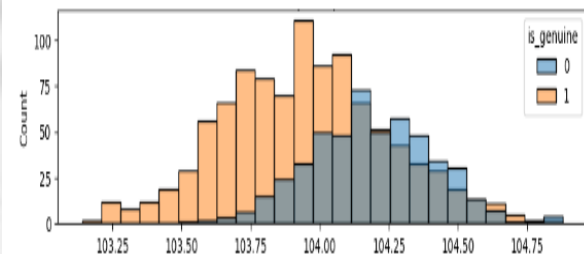
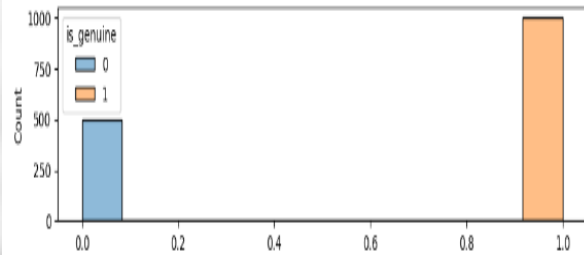
## EDA Observations:

- There are 1500 rows and 7 columns in the given dataset.
- All features have the right data type with the data values.
- There are no duplicate records in the given dataset.
- There are 37 null values in the margin\_low feature. Rest All doesn't have any null values.

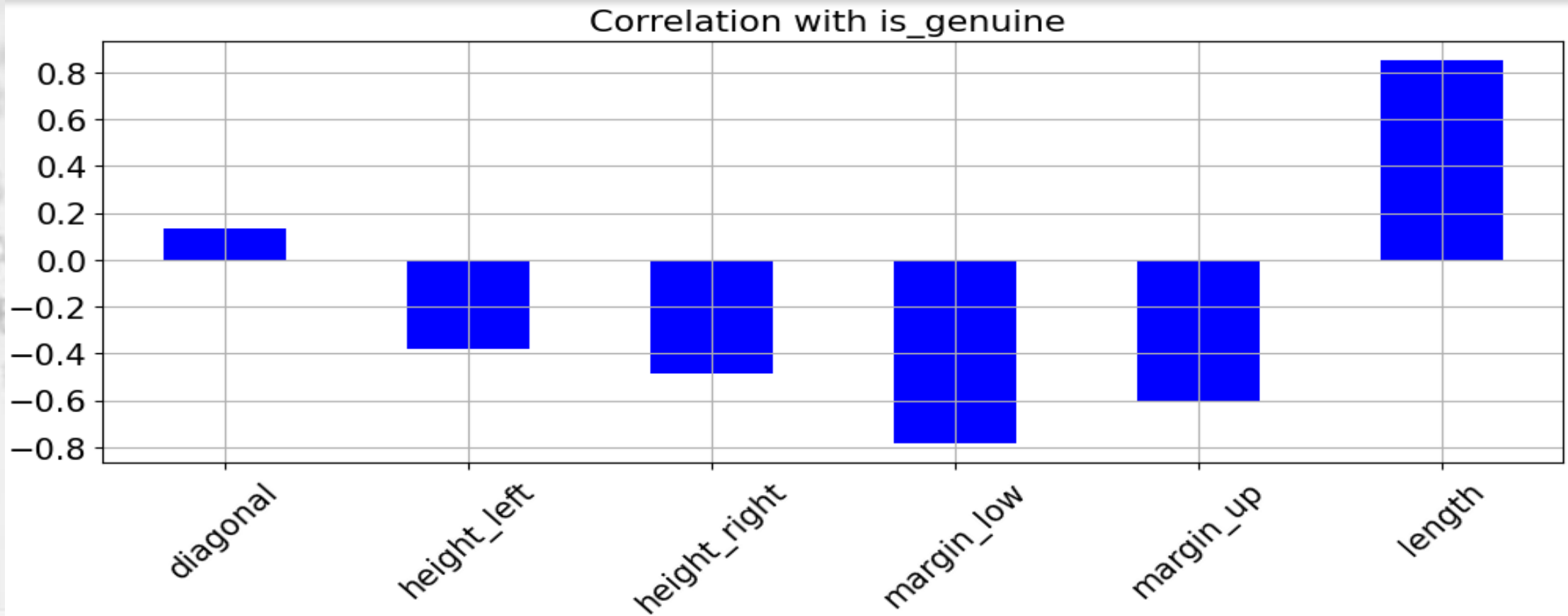
➤ There are

Distinct values in feature	is_genuine	are	2
Distinct values in feature	diagonal	are	159
Distinct values in feature	height_left	are	155
Distinct values in feature	height_right	are	170
Distinct values in feature	margin_low	are	285
Distinct values in feature	margin_up	are	123
Distinct values in feature	length	are	336

# Data Visuals:



- Among all features, length and is\_genuine are highly and positively correlated, Margin\_low and is\_genuine are highly and negatively correlated.





## ❖ Conclusion about models:

Except for Bernoulli naive Bayes and SVM without parameters, all models are giving us with around 99% accuracy.

	Model	Train score(in %) (without parameter tuning)	Test Score(in %) (without parameter tuning)	Train score(in %) (with parameter tuning)	Test Score(in %) (with parameter tuning)
0	Guassian Naive Bayes	99.33	99.24	99.33	99.24
1	Multinomial Naive Bayes	93.13	93.94	93.28	94.39
2	Bernoulli Naive Bayes	50.22	49.55	50.22	49.55
3	Complement Naive Bayes	93.28	94.39	93.28	94.39
4	Bagging Classifier	99.85	98.94	99.03	98.79
5	LogisticRegression	99.03	98.94	99.25	99.09
6	DecisionTreeClassifier	100.00	98.18	99.55	98.64
7	RandomForestClassifier	100.00	99.39	99.25	98.94
8	svm	50.22	49.55	99.40	99.24
9	KNeighborsClassifier	99.25	99.24	100.00	99.55
10	GradientBoostingClassifier	100.00	98.79	99.40	99.09
11	AdaBoostClassifier	100.00	98.94	99.70	98.79

❖ Selected Model For Deployment

❖ Logistic Regression

❑ Deployment Environment:

❑ Streamlit

❑ Spyder

- Setting up – Code in spyder for deployment

The image shows the Spyder Python IDE interface. The main window is titled "Spyder (Python 3.9)". The menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, and Help. The toolbar contains icons for file operations, running, and debugging. The file explorer on the left shows the current project files: temp.py, fake\_bills.csv, and fakebill\_deployment.py. The code editor displays the following Python code:

```
1  """
2
3  fakebills_classification Logistic model deployment
4
5  """
6
7  import pandas as pd
8  import streamlit as st
9  from sklearn.linear_model import LogisticRegression
10
11  st.title('Model Deployment: Logistic Regression')
12
13  st.sidebar.header('User Input Parameters')
14
15  def user_input_features():
16      diagonal = st.sidebar.number_input('enter diagonal value')
17      height_left = st.sidebar.number_input('enter height_left value')
18      height_right = st.sidebar.number_input('enter height_right value')
19      margin_low = st.sidebar.number_input("enter margin_low value")
20      margin_up = st.sidebar.number_input("enter margin_up value")
21      length = st.sidebar.number_input('enter length value')
22
23      data = {'diagonal':diagonal,
24             'height_left':height_left,
25             'height_right':height_right,
26             'margin_low':margin_low,
27             'margin_up':margin_up,
28             'length':length}
29      features = pd.DataFrame(data,index = [0])
30      return features
31
32  df = user_input_features()
33  st.subheader('User Input parameters')
34  st.write(df)
35
36  fake_bills = pd.read_csv("fake_bills.csv",sep=';')
```

The right sidebar contains a "Usage" panel with the following text:

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in **Preferences > Help**.

Below the Usage panel are tabs for Help, Variable Explorer, Plots, and Files. The Console panel at the bottom shows the following output:

```
Python 3.9.12 (main, Apr 4 2022, 05:22:27)
[MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for
more information.

IPython 8.2.0 -- An enhanced Interactive
Python.

In [1]:
```

The status bar at the bottom indicates "LSP Python: ready", "conda: base (Python 3.9.12)", "Line 1, Col 1", "ASCII LF RW", and "Mem 81%".

## Creating Deployment Page using Streamlit:

```
C:\WINDOWS\system32\cmd.exe

(base) C:\Users\HP>pip install streamlit

Requirement already satisfied: decorator>=3.4.0 in c:\users\hp\anaconda3\lib\site-packages (from validators>=0.2->streamlit) (5.1.1)

(base) C:\Users\HP>streamlit run    fakebill_deployment.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.29.10:8501
```

# Testing the Deployment Page: fake invoice

← ↻ ⓘ localhost:8501

User Input Parameters

enter diagonal value

172.06 - +

enter height\_left value

104.28 - +

enter height\_right value

104.06 - +

enter margin\_low value

4.63 - +

enter margin\_up value

3.37 - +

enter length value

112.07 - +

## Model Deployment: Logistic Regression

### User Input parameters

	diagonal	height_left	height_right	margin_low	margin_up	length
0	172.0600	104.2800	104.0600	4.6300	3.3700	112.0700

### Predicted Result

fake invoice

### Prediction Probability

	0	1
0	0.9853	0.0147

# Testing the Deployment Page: genuine invoice

← ↻ ⓘ localhost:8501

**User Input Parameters**

enter diagonal value

171.36 - +

enter height\_left value

103.91 - +

enter height\_right value

103.94 - +

enter margin\_low value

3.52 - +

enter margin\_up value

3.01 - +

enter length value

112.54 - +

## Model Deployment: Logistic Regression

### User Input parameters

	diagonal	height_left	height_right	margin_low	margin_up	length
0	171.3600	103.9100	103.9400	3.5200	3.0100	12.5400

### Predicted Result

genuine invoice

### Prediction Probability

	0	1
0	0.0003	0.9997

THANK YOU