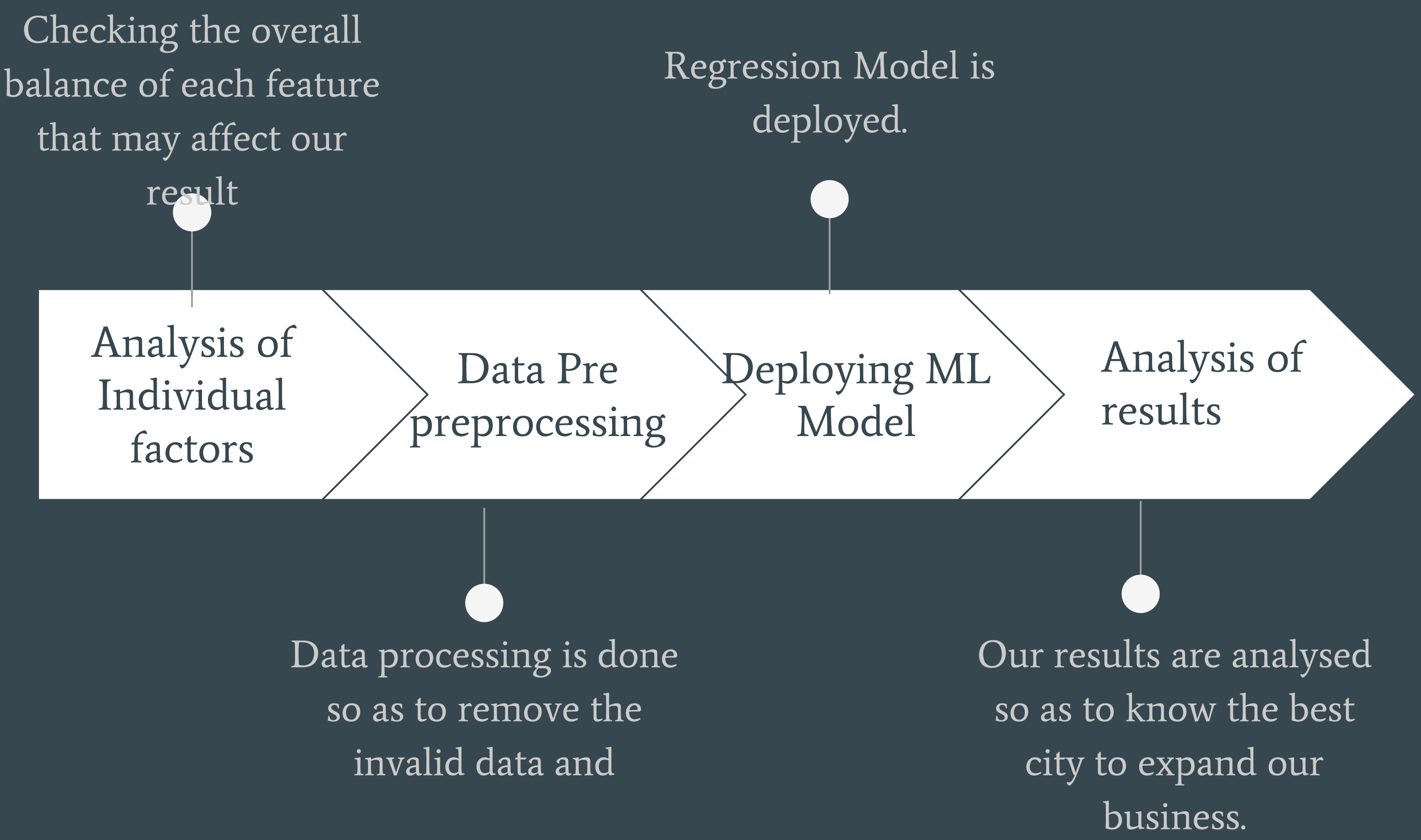
Processing of Telemarketing Data

May 25, 2020



Analysis of Individual factors

Checking the overall balance of each feature that may affect our result

Data Pre preprocessing

Data processing is done so as to remove the invalid data and

Deploying ML Model

Regression Model is deployed.

Analysis of results

Our results are analysed so as to know the best city to expand our business.

Project objective:

To find the best case scenario in which the customer purchases a solar panel.

Dataset Description

The data is collected by the Telemarketing Team of Peacock Solar and shared to the interns so as to analyse it. The dataset contains around 1500 rows of data points and feature variables. My main objective is to use various modules provided by python to deploy a regression model so as to know the best case scenario in which we can get a customer to purchase a solar panel.

Analysis Of Individual Feature variables.

Our Feature Variables include the following

Lead Validation

Valid Contact or an Invalid Contact

Calling Time & Date

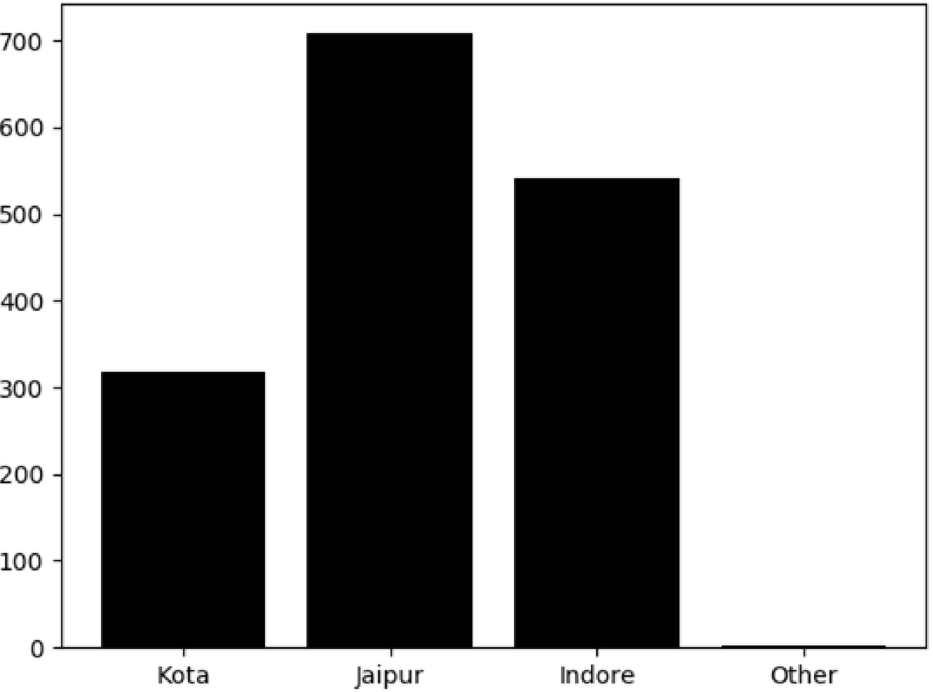
The time and date at which the call was made.

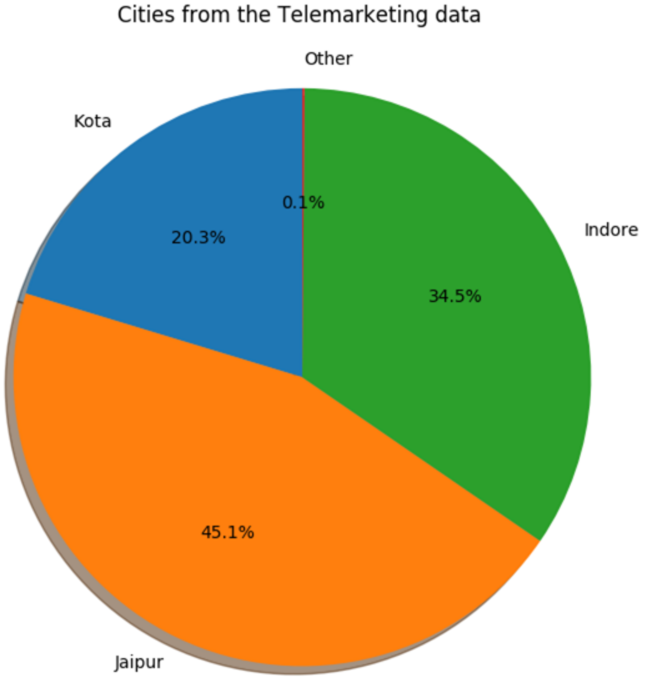
Picked the Phone

Whether the person picked up the call or not. If he did did he ask to call later.

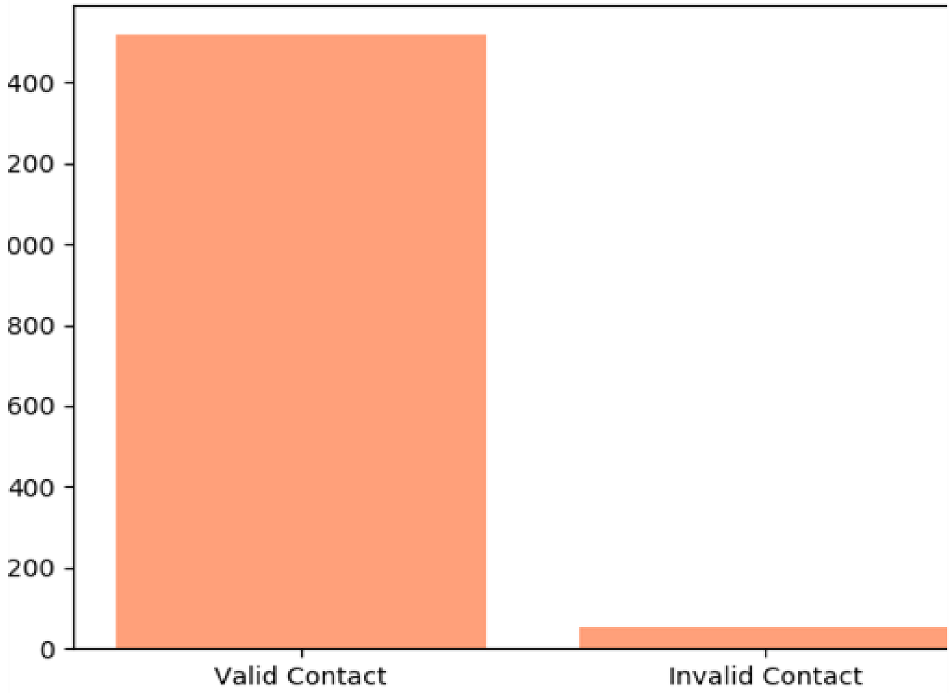
City

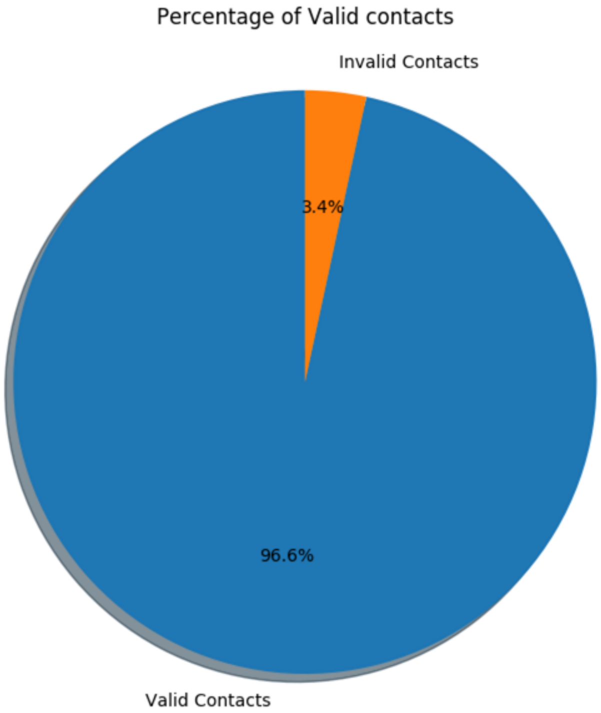
The customer is from which city.





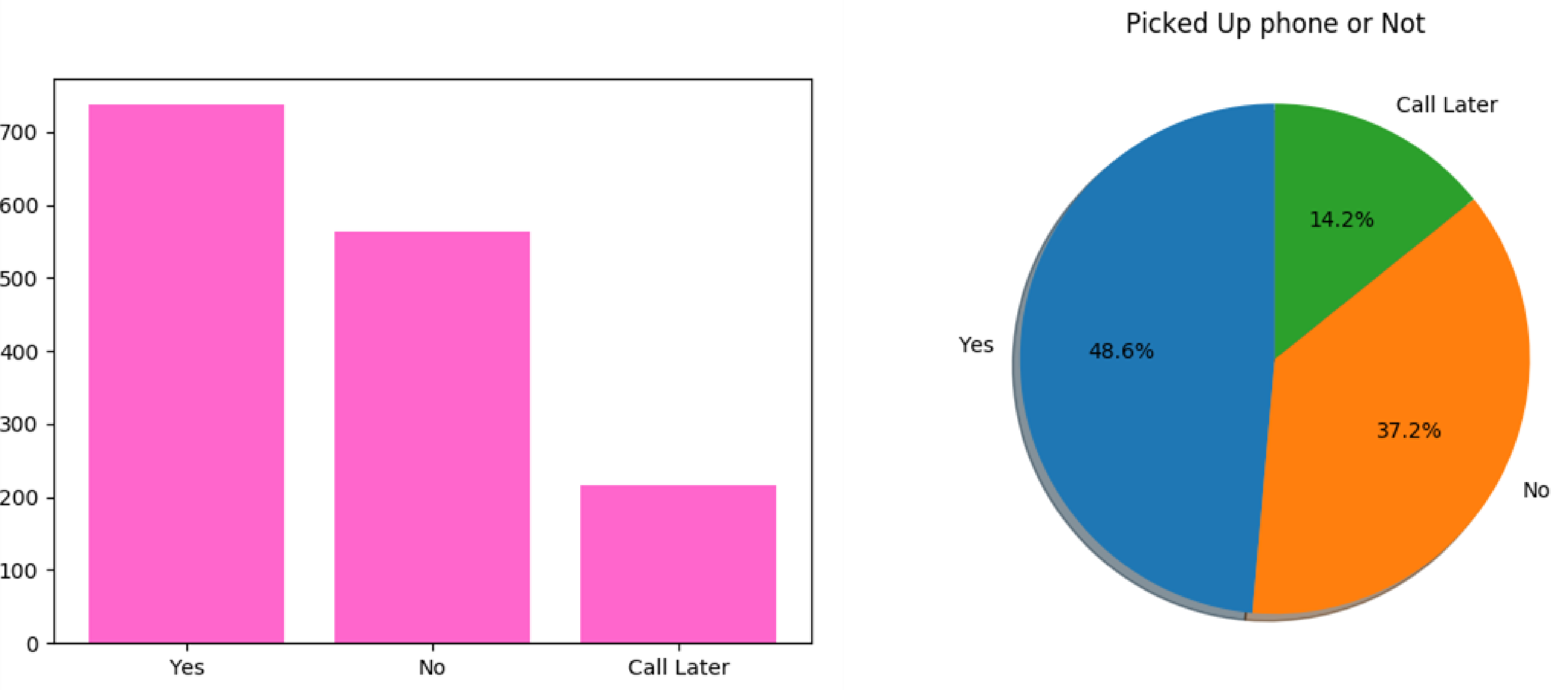
City Feature Variable





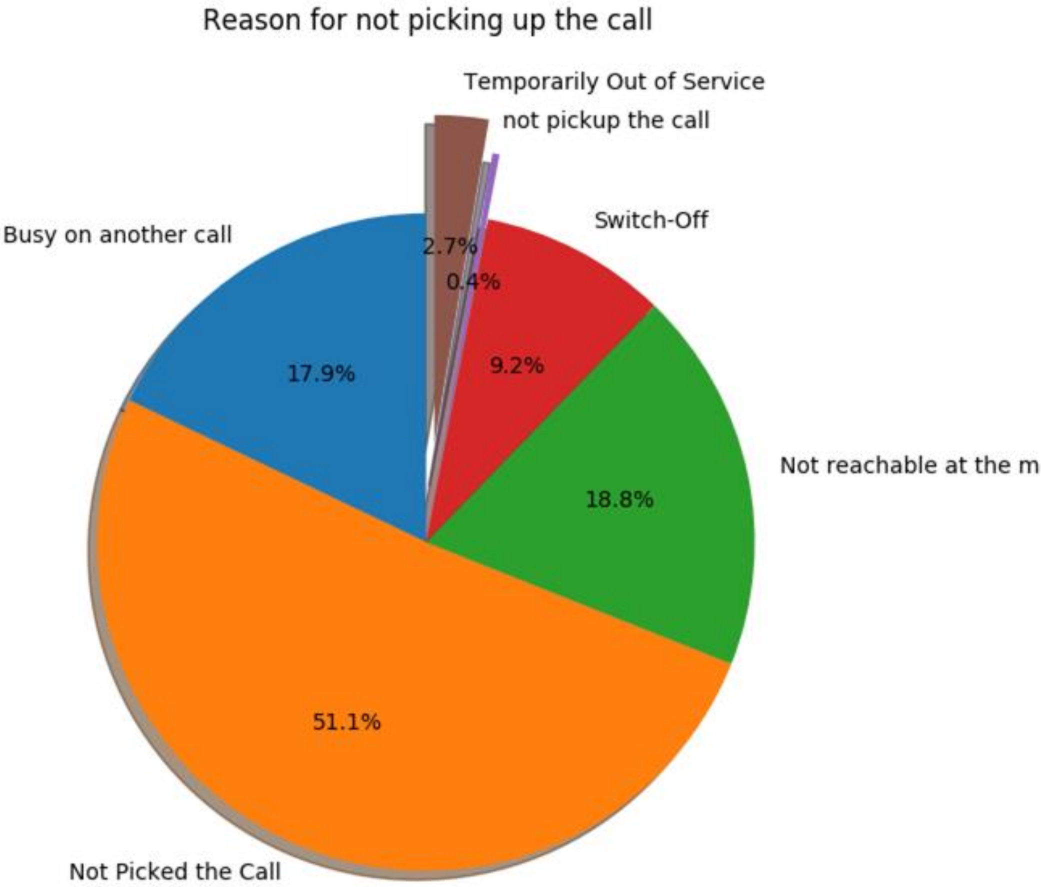
Lead Validation Feature Variable

Calling Time Variable

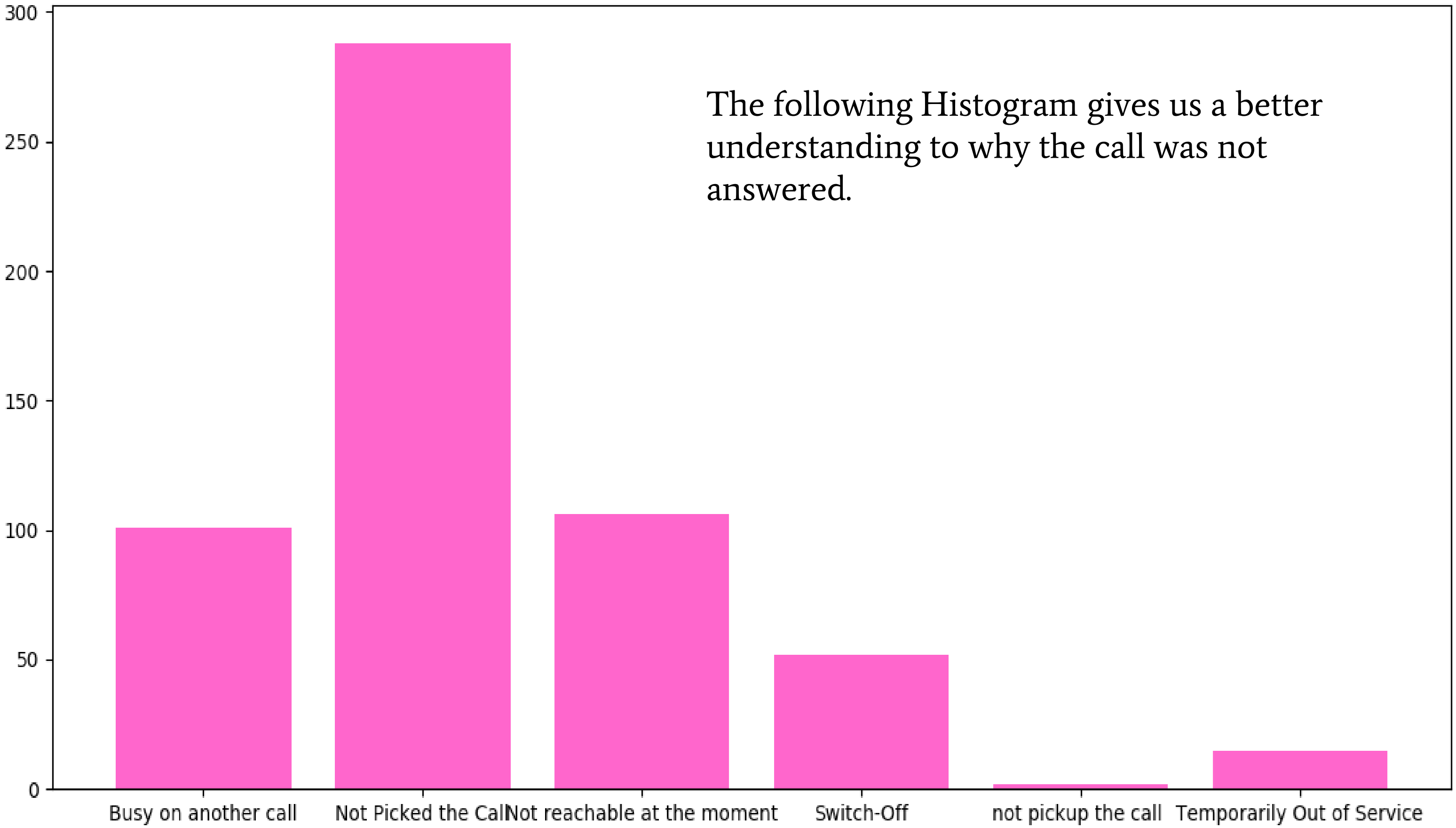


Picked up the Phone Variable

As we can see more than 35% of the calls were not picked up. So let us analyse this in greater depth.



The following Histogram gives us a better understanding to why the call was not answered.



Feature parameters used in Machine Learning Modelling

- Lead Validation : {Valid Contact : 1 , Invalid Contact : 0 },
- Picked the phone : { Yes : 1 , No : 0 , Call Later : 2} ,
- City : {Kota :1 , Jaipur : 2 , Indore : 3 , Other : 0 },

Class parameter

- Lead Interested or Not : {Yes : 1 ,No:0}

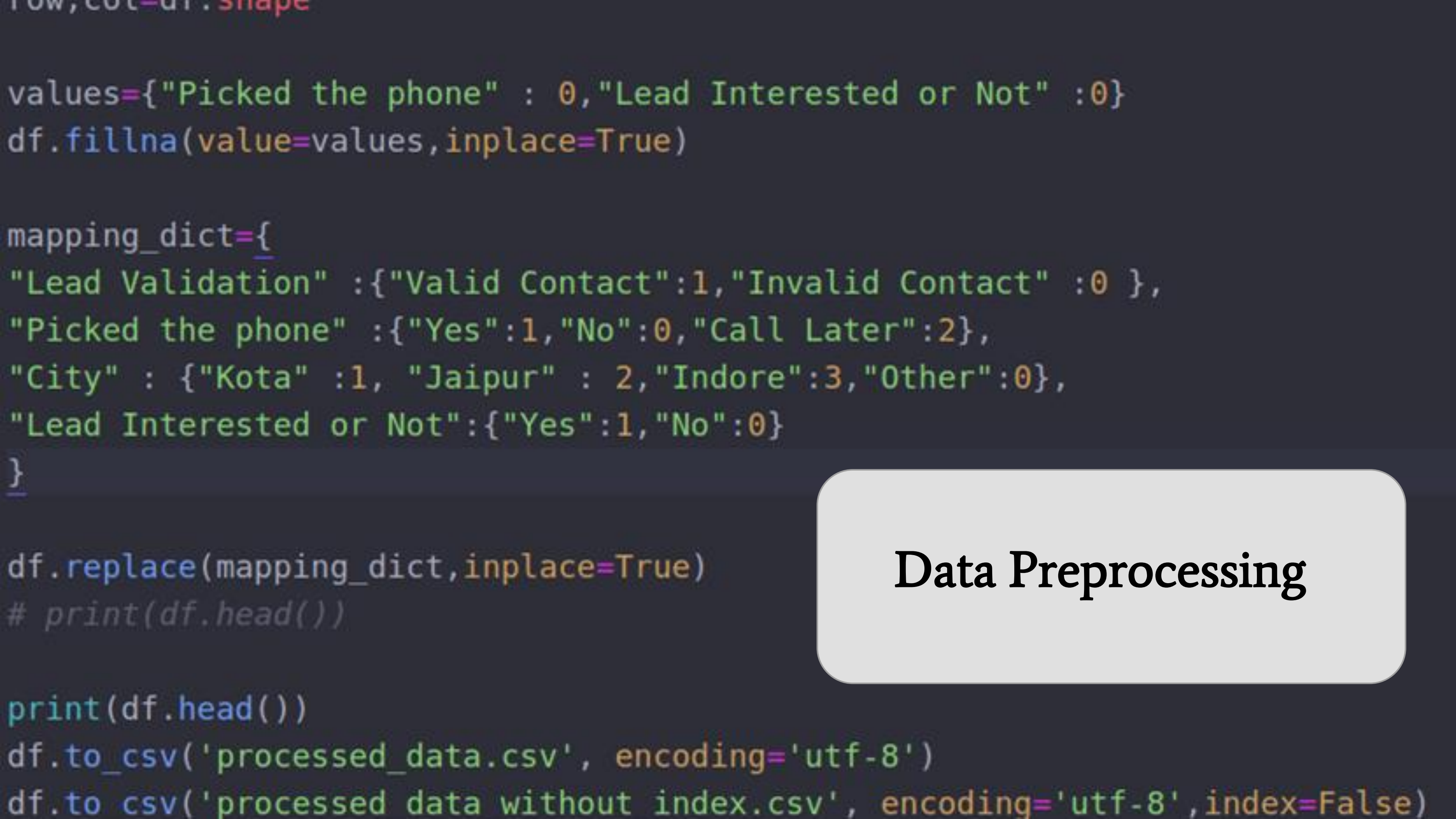
The calling time and date feature variable is omitted as the data set contains only 1500 data points and the feature takes a total of around 15 values which makes the model rather unreliable.

Deploying the regression model with the above feature variables.

Tools Used : Python and Advanced Excel

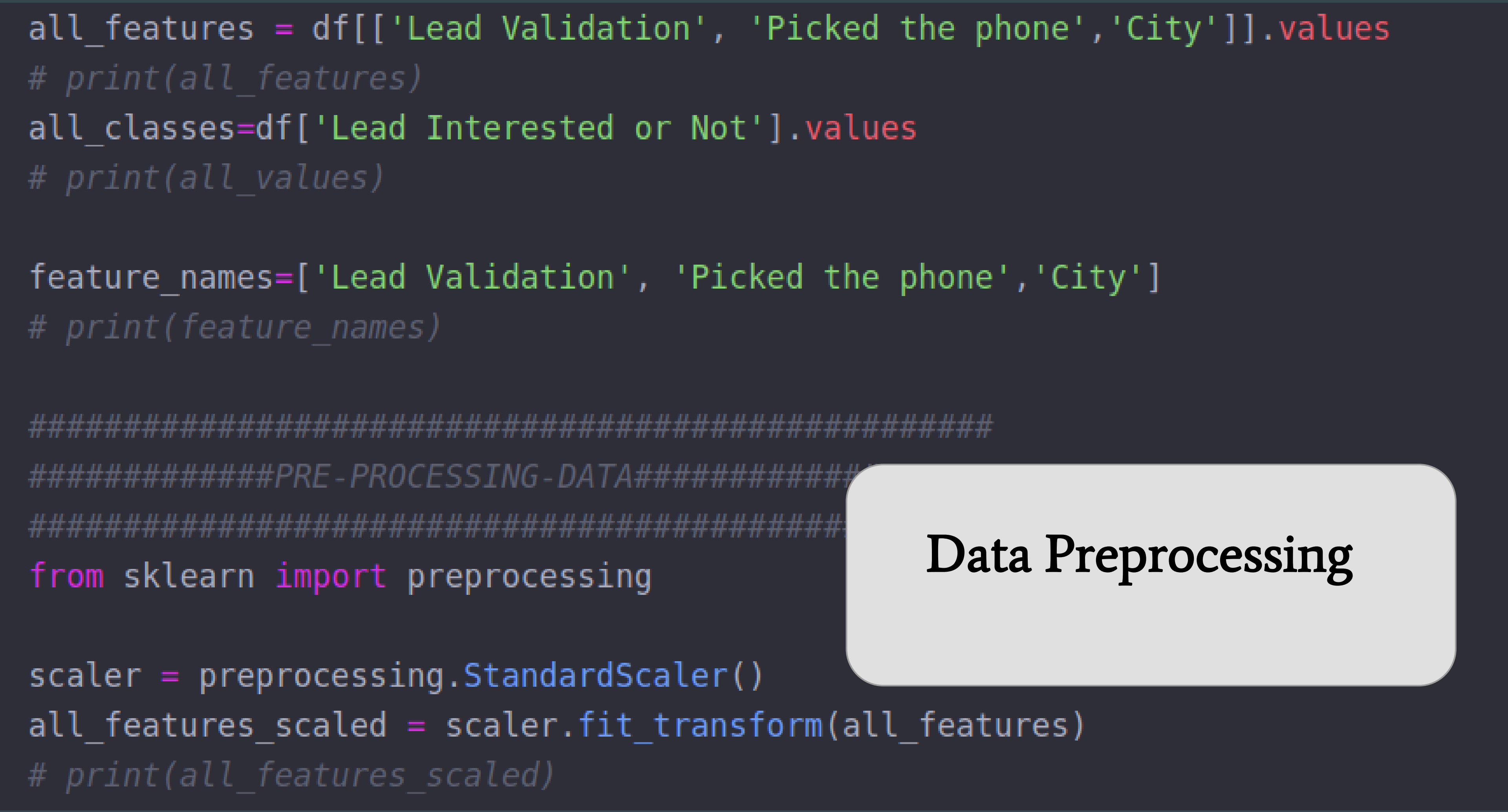
Necessary python libraries and packages used

```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
from sklearn import preprocessing
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_score
```



```
row,col=df.shape

values={"Picked the phone" : 0,"Lead Interested or Not" :0}
df.fillna(value=values,inplace=True)

mapping_dict={
"Lead Validation" :{"Valid Contact":1,"Invalid Contact" :0 },
"Picked the phone" :{"Yes":1,"No":0,"Call Later":2},
"City" : {"Kota" :1, "Jaipur" : 2,"Indore":3,"Other":0},
"Lead Interested or Not":{"Yes":1,"No":0}
}

df.replace(mapping_dict,inplace=True)
# print(df.head())

print(df.head())
df.to_csv('processed_data.csv', encoding='utf-8')
df.to_csv('processed_data_without_index.csv', encoding='utf-8',index=False)
```

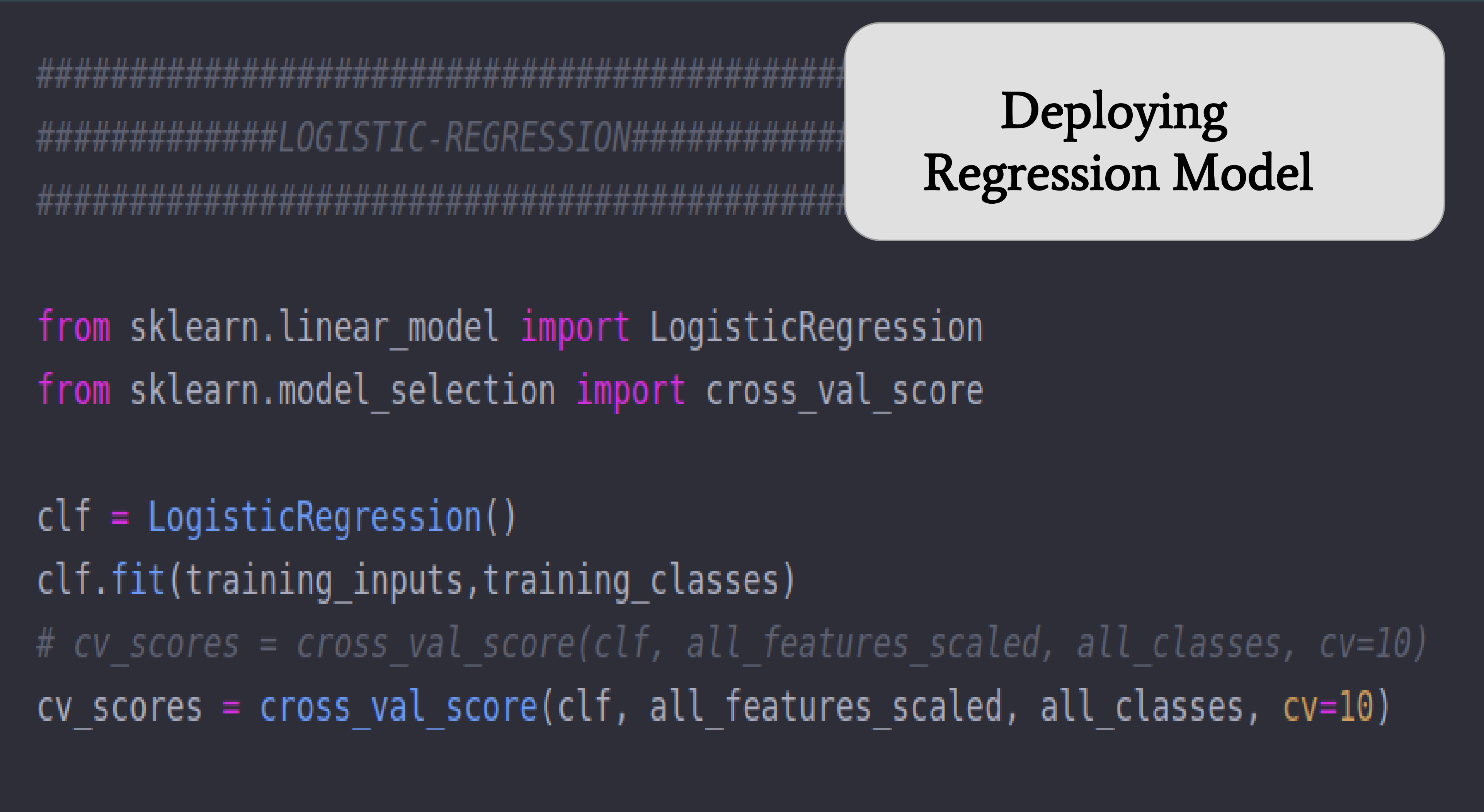
Data Preprocessing



```
all_features = df[['Lead Validation', 'Picked the phone','City']].values
# print(all_features)
all_classes=df['Lead Interested or Not'].values
# print(all_values)

feature_names=['Lead Validation', 'Picked the phone','City']
# print(feature_names)

###################################################
#############PRE-PROCESSING-DATA###########
###########################################
from sklearn import preprocessing

scaler = preprocessing.StandardScaler()
all_features_scaled = scaler.fit_transform(all_features)
# print(all_features_scaled)
```

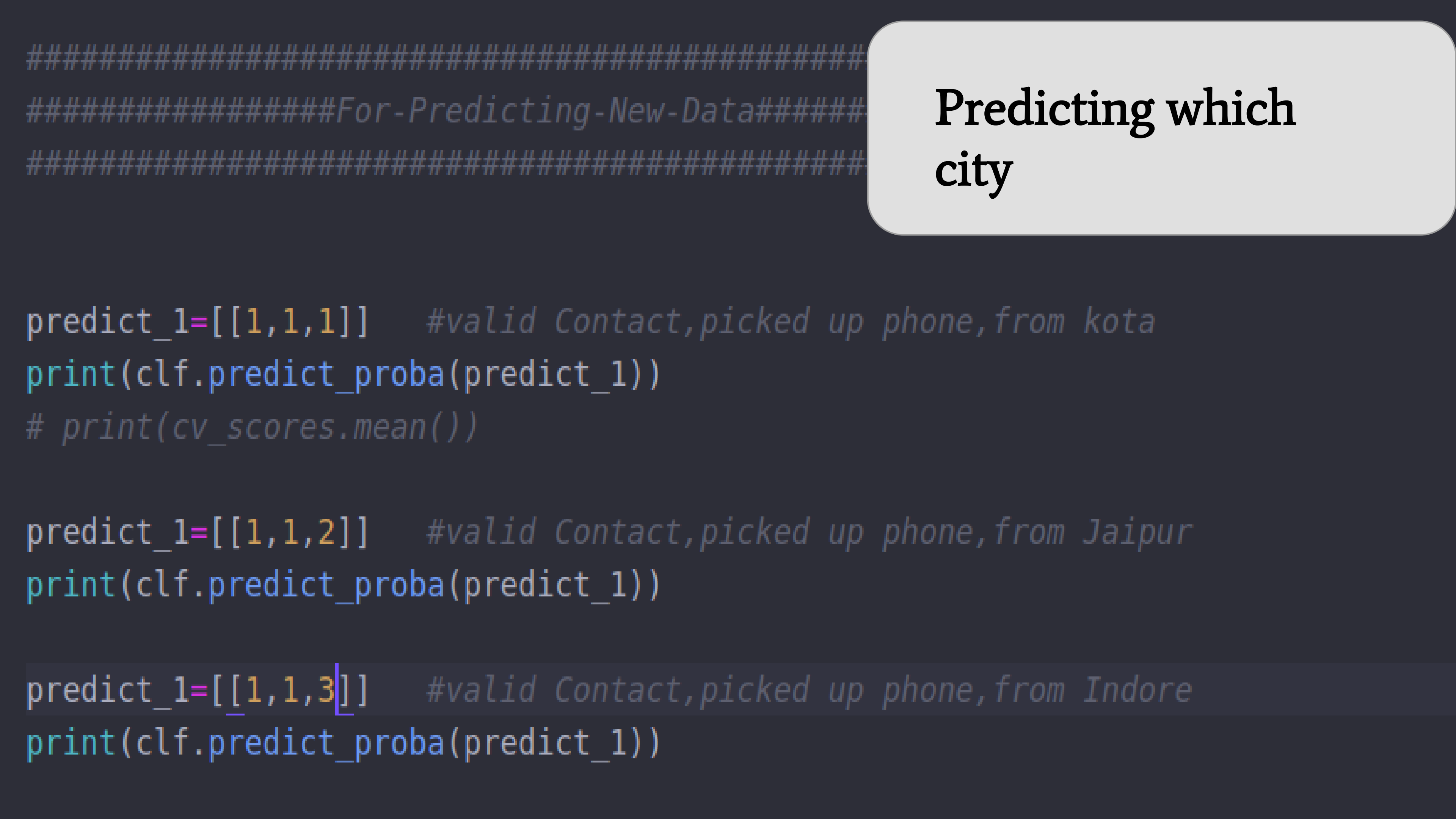
Data Preprocessing

Deploying Regression Model



```
############################################
#############LOGISTIC-REGRESSION###########
############################################

from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_score

clf = LogisticRegression()
clf.fit(training_inputs,training_classes)
# cv_scores = cross_val_score(clf, all_features_scaled, all_classes, cv=10)
cv_scores = cross_val_score(clf, all_features_scaled, all_classes, cv=10)
```

Predicting which city



```
##############################################
#################For-Predicting-New-Data######
##############################################


predict_1=[[1,1,1]]   #valid Contact,picked up phone,from kota
print(clf.predict_proba(predict_1))
# print(cv_scores.mean())

predict_1=[[1,1,2]]   #valid Contact,picked up phone,from Jaipur
print(clf.predict_proba(predict_1))

predict_1=[[1,1,3]]   #valid Contact,picked up phone,from Indore
print(clf.predict_proba(predict_1))
```

Target audience

By applying the regression model we obtain the following data for each individual state :

Probability of not going ahead with the lead given that it’s a valid contact and they picked up the phone

Kota : 0.88195628

Jaipur : 0.8541652

Indore : 0.82115801

So the chances of a customer buying a solar panel is more in Indore when compared to the other states provided in the dataset.