Vaccine Usage Prediction

A Logistic Regression Approach

……………………………………………………………

Vedant Thorat

Email – vedant2000thorat@gmail.com

Introduction

The rapid spread of infectious diseases such as the H1N1 flu poses significant challenges to public health systems worldwide. Vaccination is one of the most effective strategies to prevent the spread of such diseases, yet vaccine uptake remains suboptimal in many populations due to various factors including vaccine hesitancy, misinformation, and access issues. Understanding the determinants of vaccine acceptance and predicting vaccine uptake can help public health officials design more effective vaccination campaigns and intervention strategies.

In this project, we leverage logistic regression, a widely-used statistical method for binary classification problems, to predict the likelihood of individuals receiving the H1N1 flu vaccine. By analyzing a comprehensive dataset that includes demographic information, health behaviors, and beliefs about vaccines, we aim to identify key factors influencing vaccination decisions and develop a predictive model to aid in public health planning.

The dataset used in this project contains responses from a diverse group of individuals, capturing a range of variables such as worry about the H1N1 flu, awareness levels, past health behaviors, and recommendations from healthcare providers. These variables are critical in understanding the multifaceted nature of vaccine acceptance and can provide valuable insights into how different segments of the population respond to vaccination efforts.

Objectives

The objectives of this project are threefold:

1. **Data Exploration and Preprocessing**: To thoroughly explore the dataset, handle missing values, and appropriately encode and scale the features for analysis.
2. **Model Development**: To train and evaluate a logistic regression model capable of accurately predicting H1N1 vaccine uptake.
3. **Insight Generation**: To derive actionable insights from the model and feature analysis that can inform public health strategies and interventions.

By achieving these objectives, we aim to contribute to the ongoing efforts to improve vaccine coverage and protect public health, particularly in the context of emerging infectious diseases.

Problem Statement

The goal of this project is to develop a predictive model using logistic regression to determine the likelihood that an individual will receive the H1N1 flu vaccine. Understanding the factors that influence vaccine acceptance is crucial for public health officials to design effective vaccination campaigns and interventions. The dataset includes various demographic, behavioral, and belief-related features, which will be used to build and evaluate the logistic regression model. By accurately predicting vaccine uptake, this model aims to provide insights that can help increase vaccination rates and improve public health outcomes.

Dataset Overview

The dataset used in this project consists of responses from a diverse group of individuals regarding their attitudes, behaviors, and beliefs about the H1N1 flu and vaccination. The dataset contains a mix of numerical and categorical features that provide comprehensive information about each respondent. Here is a detailed overview of the columns in the dataset:

Column	Description
unique_id	Unique identifier for each respondent
h1n1_worry	Worry about the h1n1 flu (0,1,2,3) 0=Not worried at all, 1=Not very worried, 2=Somewhat worried, 3=Very worried
h1n1_awareness	Signifies the amount of knowledge or understanding the respondent has about h1n1 flu - (0,1,2) – 0=No knowledge, 1=little knowledge, 2=good knowledge
antiviral_medication	Has the respondent taken antiviral vaccination - (0,1)
contact_avoidance	Has avoided any close contact with people who have flu like symptoms - (0,1)
bought_face_mask	Has the respondent bought mask or not - (0,1)
wash_hands_frequently	Washes hands frequently or uses hand sanitizer - (0,1)
avoid_large_gatherings	Has the respondent reduced time spent at large gatherings - (0,1)

reduced_outside_home_cont	Has the respondent reduced contact with people outside their own house - (0,1)
avoid_touch_face	Avoids touching nose, eyes, mouth - (0,1)
dr_recc_h1n1_vacc	Doctor has recommended h1n1 vaccine - (0,1)
dr_recc_seasonal_vacc	The doctor has recommended seasonal flu vaccine - (0,1)
chronic_medic_condition	Has any chronic medical condition - (0,1)
cont_child_undr_6_mnth	Has regular contact with child the age of 6 months - (0,1)
is_health_worker	Is respondent a health worker - (0,1)
has_health_insur	Does respondent have health insurance - (0,1)
is_h1n1_vacc_effective	Does respondent think that the h1n1 vaccine is effective - (1,2,3,4,5)- 1=Thinks not effective at all, 2 = Thinks it is not very effective, 3=Doesn't know if it is effective or not, 4=Thinks it is somewhat effective, 5=Thinks it is highly effective
is_h1n1_risky	What respondents think about the risk of getting ill with h1n1 in the absence of the vaccine- (1,2,3,4,5)- 1=Thinks it is not very low risk, 2=Thinks it is somewhat low risk, 3=don’t know if it is risky or not,

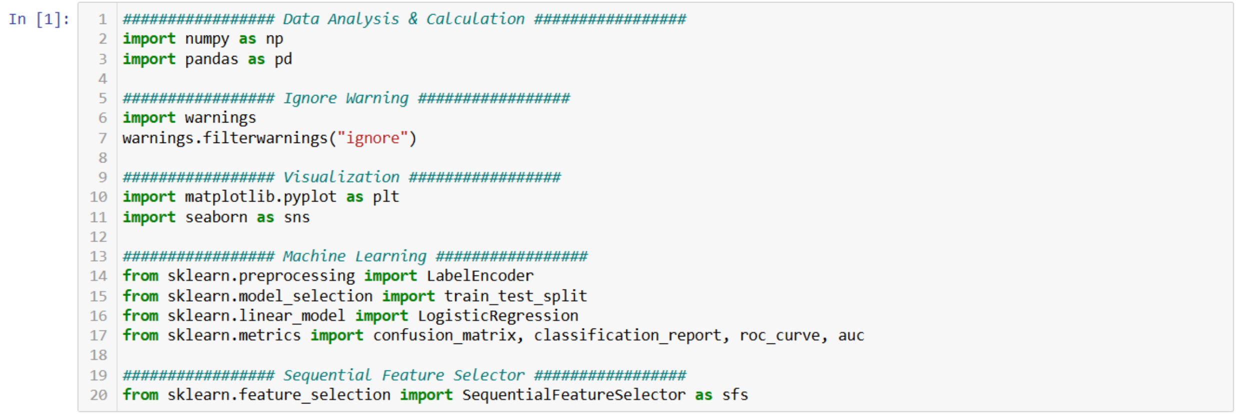
	4=Thinks it is a somewhat high risk, 5=Thinks it is very highly risky
sick_from_h1n1_vacc	Does respondent worry about getting sick by taking the h1n1 vaccine - (1,2,3,4,5) 1=Respondent not worried at all, 2=Respondent is not very worried, 3=Doesn't know, 4=Respondent is somewhat worried, 5=Respondent is very worried
is_seas_vacc_effective	Does respondent think that the seasonal vaccine is effective- (1,2,3,4,5) 1=Thinks not effective at all, 2=Thinks it is not very effective, 3=Doesn't know if it is effective or not, 4=Thinks it is somewhat effective, 5=Thinks it is highly effective
is_seas_flu_risky	What respondents think about the risk of getting ill with seasonal flu in the absence of the vaccine- (1,2,3,4,5) 1=Thinks it is not very low risk, 2=Thinks it is somewhat low risk, 3=Doesn't know if it is risky or not, 4=Thinks it is somewhat high risk, 5=Thinks it is very highly risky
sick_from_seas_vacc	Does respondent worry about getting sick by taking the seasonal flu vaccine - (1,2,3,4,5) 1=Respondent not worried at all,

	2=Respondent is not very worried, 3=Doesn't know, 4=Respondent is somewhat worried, 5=Respondent is very worried
age_bracket	Age bracket of the respondent – 18 - 34 Years, 35 – 44 Years, 45 - 54 Years, 55 - 64 Years, 64+ Years
qualification	Qualification/education level of the respondent as per their response <12 Years, 12 Years, College Graduate, Some College
race	Respondent's race – White, Black, Other Multiple Hispanic
sex	Respondent's sex - (Female, Male)
income_level	Annual income of the respondent as per the 2008 poverty Census <=75000−AbovePoverty

	> 75000−AbovePoverty >75000, Below Poverty
marital_status	Respondent's marital status - (Not Married, Married)
housing_status	Respondent's housing status - (Own, Rent)
employment	Respondent's employment status – Not in Labor Force, Employed, Unemployed
census_msa	Residence of the respondent with the MSA metropolitan statistical area Non-MSA, MSA- Not Principle, CityMSA-Principal city - (Yes, no)
no_of_adults	Number of adults in the respondent's house (0,1,2,3) - (Yes, no)
no_of_children	Number of children in the respondent's house (0,1,2,3) - (Yes, No)
h1n1_vaccine	(Dependent variable) Did the respondent receive the h1n1 vaccine or not (1,0) - (Yes, No)

➢ Import Dataset

You need to import various libraries for data analysis, visualization, and machine learning.

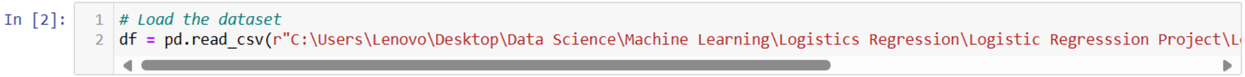


In [1]:

```python
################# Data Analysis & Calculation #################
import numpy as np
import pandas as pd

################# Ignore Warning #################
import warnings
warnings.filterwarnings("ignore")

################# Visualization #################
import matplotlib.pyplot as plt
import seaborn as sns

################# Machine Learning #################
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, classification_report, roc_curve, auc

################# Sequential Feature Selector #################
from sklearn.feature_selection import SequentialFeatureSelector as sfs
```

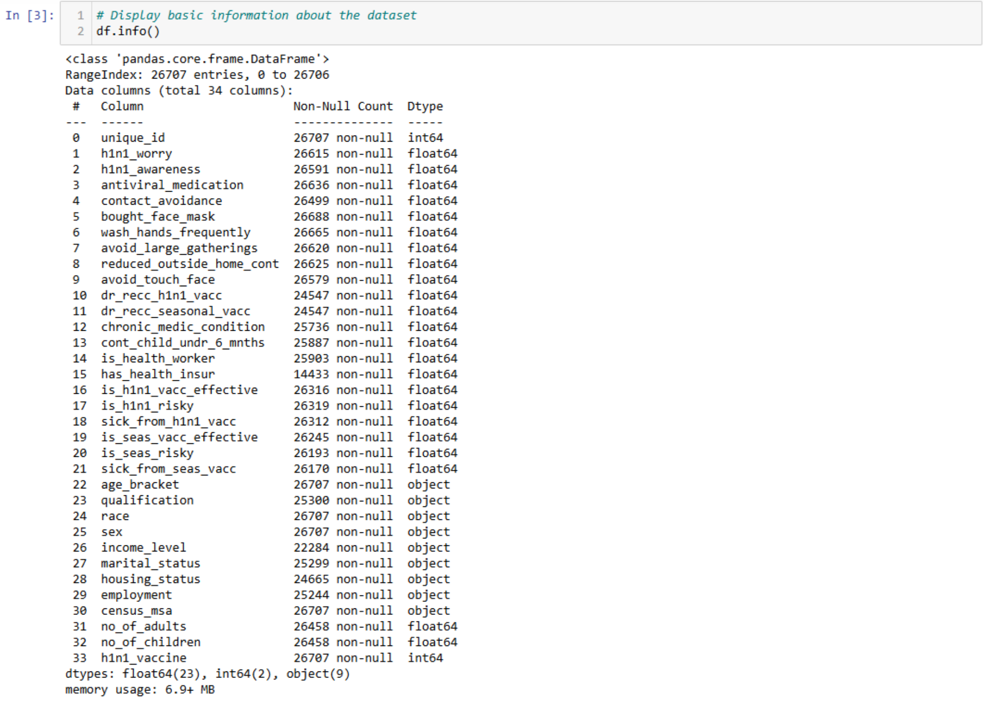
➢ Load Dataset

Load the dataset into a DataFrame and display basic information to understand the data structure.



In [2]:

```python
# Load the dataset
df = pd.read_csv(r"C:\Users\Lenovo\Desktop\Data Science\Machine Learning\Logistics Regression\Logistic Regresssion Project\L
```



In [3]:

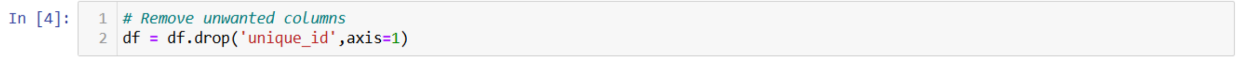
```python
# Display basic information about the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26707 entries, 0 to 26706
Data columns (total 34 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   unique_id                 26707 non-null  int64  
 1   h1n1_worry                26615 non-null  float64
 2   h1n1_awareness            26591 non-null  float64
 3   antiviral_medication      26636 non-null  float64
 4   contact_avoidance         26499 non-null  float64
 5   bought_face_mask          26688 non-null  float64
 6   wash_hands_frequently     26665 non-null  float64
 7   avoid_large_gatherings    26620 non-null  float64
 8   reduced_outside_home_cont 26625 non-null  float64
 9   avoid_touch_face          26579 non-null  float64
 10  dr_recc_h1n1_vacc         24547 non-null  float64
 11  dr_recc_seasonal_vacc     24547 non-null  float64
 12  chronic_medic_condition   25736 non-null  float64
 13  cont_child_undr_6_mnths   25887 non-null  float64
 14  is_health_worker          25903 non-null  float64
 15  has_health_insur          14433 non-null  float64
 16  is_h1n1_vacc_effective    26316 non-null  float64
 17  is_h1n1_risky             26319 non-null  float64
 18  sick_from_h1n1_vacc       26312 non-null  float64
 19  is_seas_vacc_effective    26245 non-null  float64
 20  is_seas_risky             26193 non-null  float64
 21  sick_from_seas_vacc       26170 non-null  float64
 22  age_bracket               26707 non-null  object 
 23  qualification             25300 non-null  object 
 24  race                      26707 non-null  object 
 25  sex                       26707 non-null  object 
 26  income_level              22284 non-null  object 
 27  marital_status            25299 non-null  object 
 28  housing_status            24665 non-null  object 
 29  employment                25244 non-null  object 
 30  census_msa                26707 non-null  object 
 31  no_of_adults              26458 non-null  float64
 32  no_of_children            26458 non-null  float64
 33  h1n1_vaccine              26707 non-null  int64  
dtypes: float64(23), int64(2), object(9)
memory usage: 6.9+ MB
```

➢ Data Preprocessing

Remove Unwanted Columns

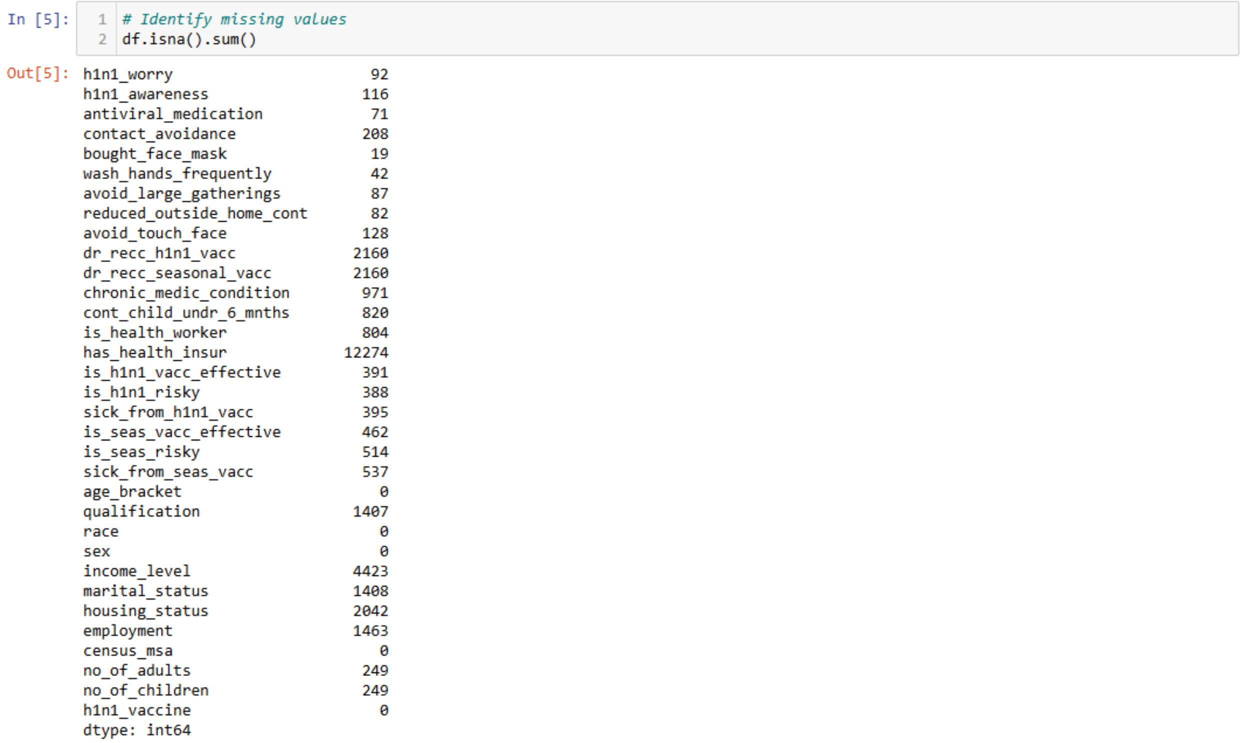
Remove columns that are not needed for analysis.



```
In [4]: # Remove unwanted columns
        df = df.drop('unique_id',axis=1)
```

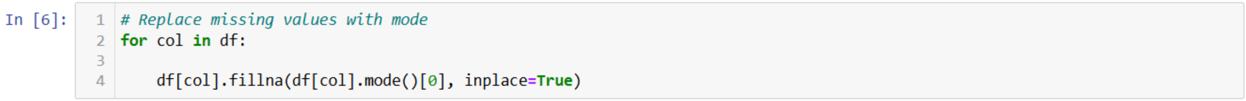
Identifying & Treatment Missing Value

Identify missing values and replace them with the mode of the respective columns.



```
In [5]: # Identify missing values
        df.isna().sum()

Out[5]: h1n1_worry                     92
        h1n1_awareness                116
        antiviral_medication           71
        contact_avoidance             208
        bought_face_mask               19
        wash_hands_frequently          42
        avoid_large_gatherings         87
        reduced_outside_home_cont      82
        avoid_touch_face              128
        dr_recc_h1n1_vacc            2160
        dr_recc_seasonal_vacc        2160
        chronic_medic_condition       971
        cont_child_undr_6_mnths       820
        is_health_worker              804
        has_health_insur            12274
        is_h1n1_vacc_effective        391
        is_h1n1_risky                 388
        sick_from_h1n1_vacc           395
        is_seas_vacc_effective        462
        is_seas_risky                 514
        sick_from_seas_vacc           537
        age_bracket                     0
        qualification                1407
        race                            0
        sex                             0
        income_level                 4423
        marital_status               1408
        housing_status               2042
        employment                   1463
        census_msa                      0
        no_of_adults                  249
        no_of_children                249
        h1n1_vaccine                    0
        dtype: int64
```

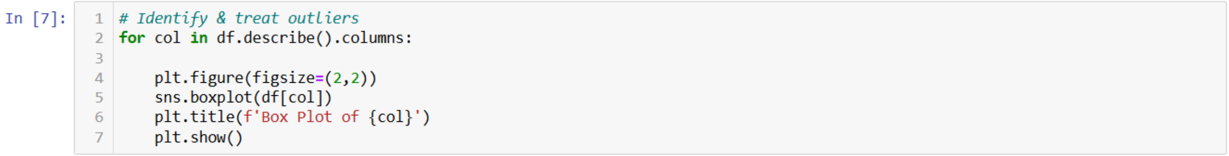
Replacing Missing value by Mode



```
In [6]: # Replace missing values with mode
        for col in df:

            df[col].fillna(df[col].mode()[0], inplace=True)
```

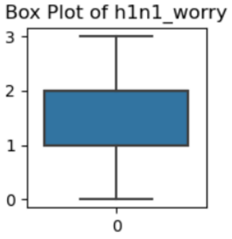
Identifying & Treatment Outliers

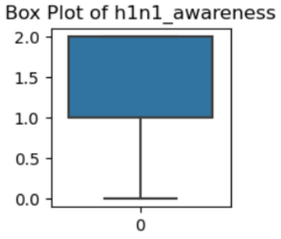
Visualize the data using box plots to detect outliers.



In [7]:

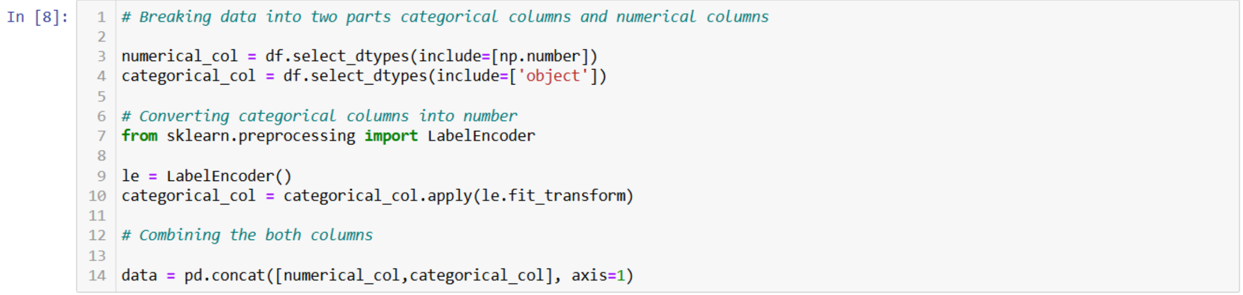
```python
# Identify & treat outliers
for col in df.describe().columns:

    plt.figure(figsize=(2,2))
    sns.boxplot(df[col])
    plt.title(f'Box Plot of {col}')
    plt.show()
```





➢ Encoding Categorical Columns

Convert categorical columns into numerical values using Label Encoding.

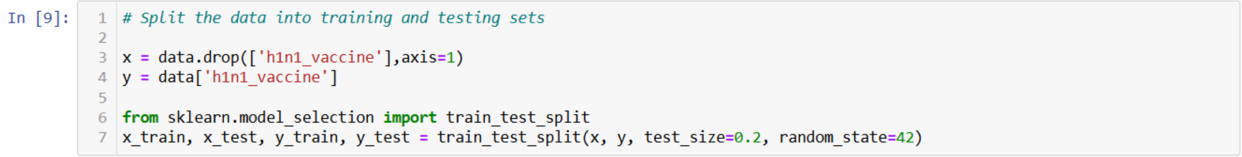


In [8]:

```python
# Breaking data into two parts categorical columns and numerical columns

numerical_col = df.select_dtypes(include=[np.number])
categorical_col = df.select_dtypes(include=['object'])

# Converting categorical columns into number
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
categorical_col = categorical_col.apply(le.fit_transform)

# Combining the both columns

data = pd.concat([numerical_col,categorical_col], axis=1)
```

Data Partition

Split the data into training and testing sets.

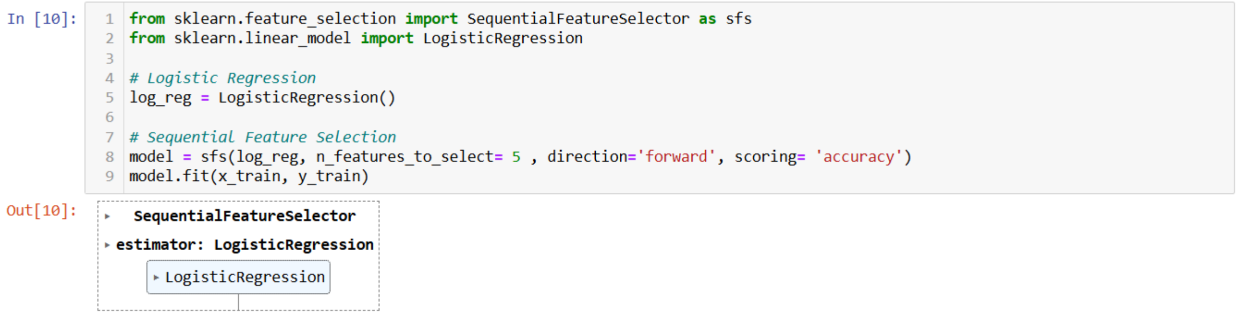


```
In [9]:  # Split the data into training and testing sets

         x = data.drop(['h1n1_vaccine'],axis=1)
         y = data['h1n1_vaccine']

         from sklearn.model_selection import train_test_split
         x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

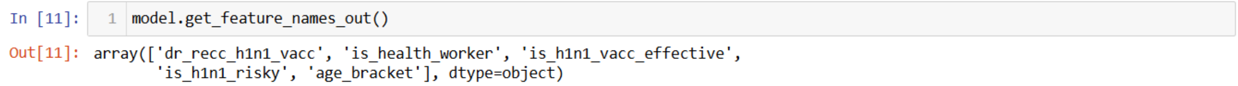
➢ Model Building

Logistic Regression with Sequential Feature Selection

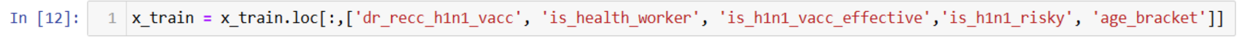
Build a Logistic Regression model and use Sequential Feature Selection to select the best features.



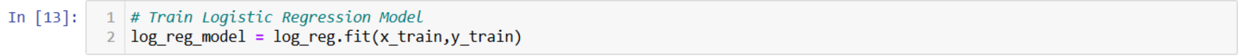
```
In [10]:  from sklearn.feature_selection import SequentialFeatureSelector as sfs
          from sklearn.linear_model import LogisticRegression

          # Logistic Regression
          log_reg = LogisticRegression()

          # Sequential Feature Selection
          model = sfs(log_reg, n_features_to_select= 5 , direction='forward', scoring= 'accuracy')
          model.fit(x_train, y_train)

Out[10]:  ▸ SequentialFeatureSelector
          ▸ estimator: LogisticRegression
              ▸ LogisticRegression
```



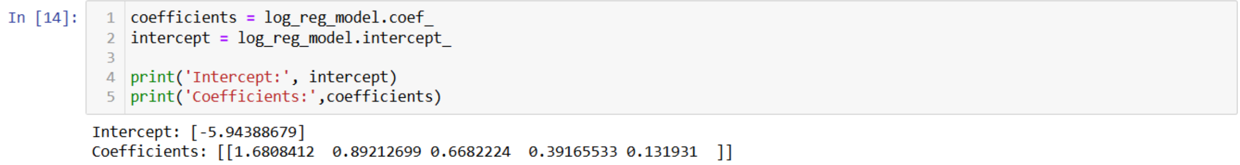
```
In [11]:  model.get_feature_names_out()

Out[11]:  array(['dr_recc_h1n1_vacc', 'is_health_worker', 'is_h1n1_vacc_effective',
                 'is_h1n1_risky', 'age_bracket'], dtype=object)
```



```
In [12]:  x_train = x_train.loc[:,['dr_recc_h1n1_vacc', 'is_health_worker', 'is_h1n1_vacc_effective','is_h1n1_risky', 'age_bracket']]
```



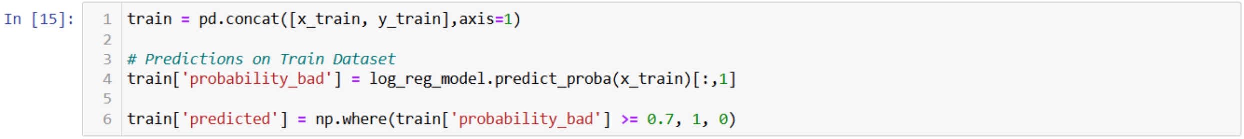
```
In [13]:  # Train Logistic Regression Model
          log_reg_model = log_reg.fit(x_train,y_train)
```



```
In [14]:  coefficients = log_reg_model.coef_
          intercept = log_reg_model.intercept_

          print('Intercept:', intercept)
          print('Coefficients:',coefficients)

          Intercept: [-5.94388679]
          Coefficients: [[1.6808412  0.89212699 0.6682224  0.39165533 0.131931   ]]
```

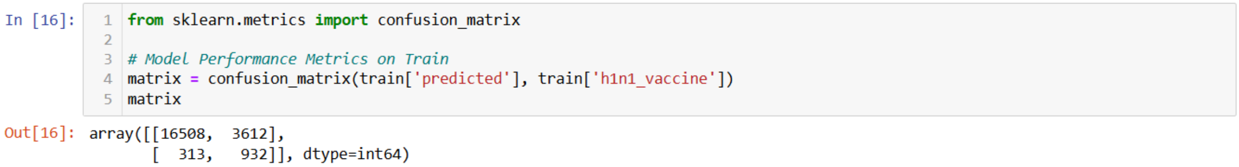
➢ Predictions on Train Dataset

Evaluate the model on the training set.

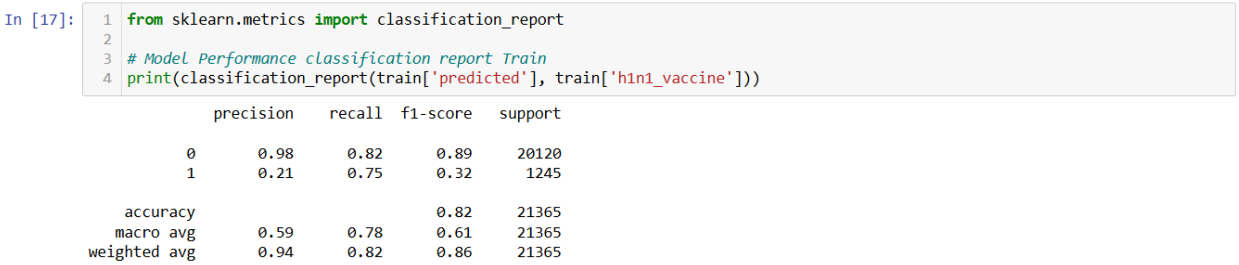


```
In [15]:
train = pd.concat([x_train, y_train],axis=1)

# Predictions on Train Dataset
train['probability_bad'] = log_reg_model.predict_proba(x_train)[:,1]

train['predicted'] = np.where(train['probability_bad'] >= 0.7, 1, 0)
```

Model Performance Metrics on Train



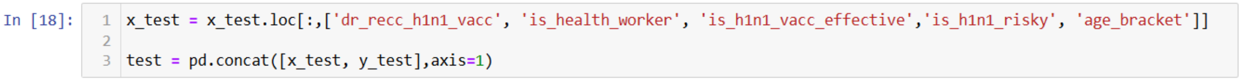
```
In [16]:
from sklearn.metrics import confusion_matrix

# Model Performance Metrics on Train
matrix = confusion_matrix(train['predicted'], train['h1n1_vaccine'])
matrix

Out[16]: array([[16508,  3612],
                [  313,   932]], dtype=int64)
```



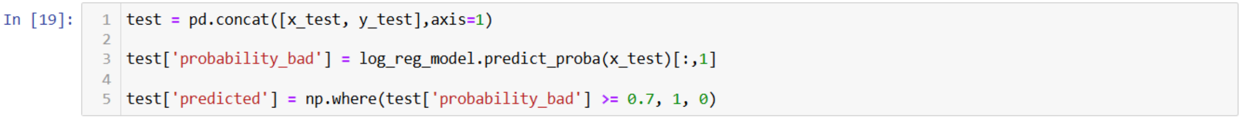
```
In [17]:
from sklearn.metrics import classification_report

# Model Performance classification report Train
print(classification_report(train['predicted'], train['h1n1_vaccine']))

              precision    recall  f1-score   support

           0       0.98      0.82      0.89     20120
           1       0.21      0.75      0.32      1245

    accuracy                           0.82     21365
   macro avg       0.59      0.78      0.61     21365
weighted avg       0.94      0.82      0.86     21365
```

➢ Predictions on Test Dataset

Evaluate the model on the testing set.

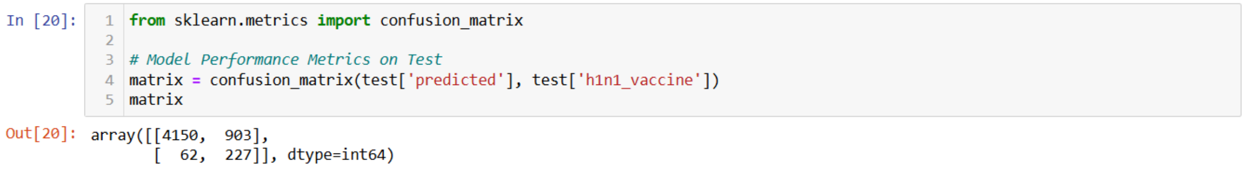


```
In [18]:
x_test = x_test.loc[:,['dr_recc_h1n1_vacc', 'is_health_worker', 'is_h1n1_vacc_effective','is_h1n1_risky', 'age_bracket']]

test = pd.concat([x_test, y_test],axis=1)
```

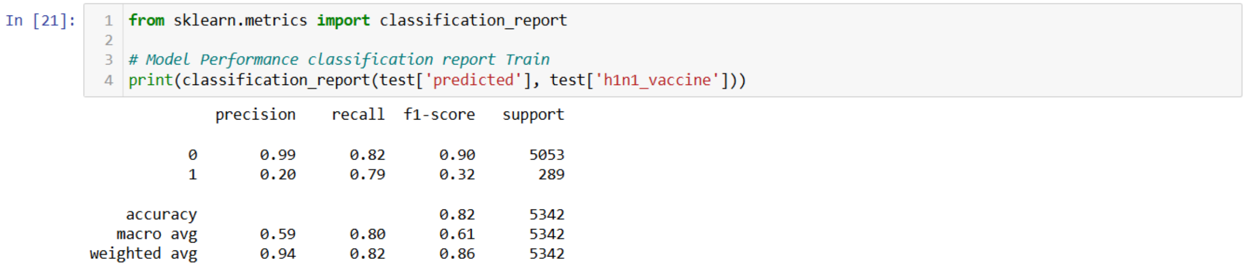


```
In [19]:
test = pd.concat([x_test, y_test],axis=1)

test['probability_bad'] = log_reg_model.predict_proba(x_test)[:,1]

test['predicted'] = np.where(test['probability_bad'] >= 0.7, 1, 0)
```

Model Performance Metrics



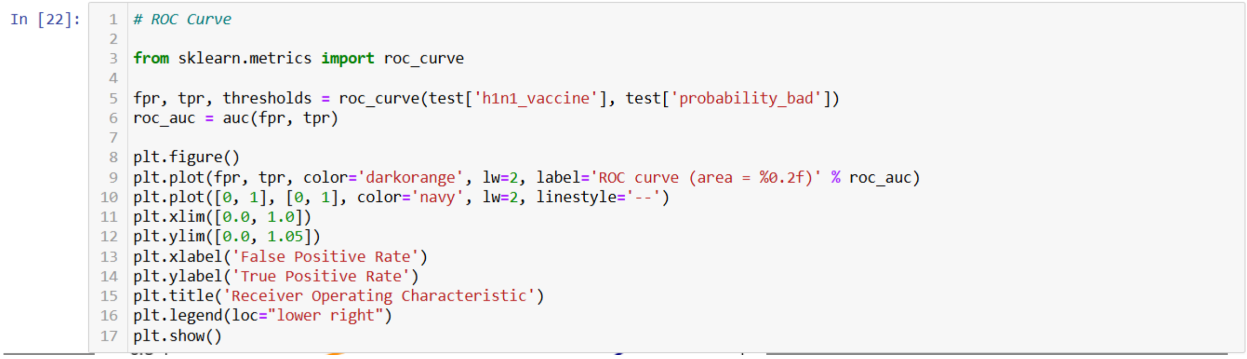
```
In [20]: from sklearn.metrics import confusion_matrix

         # Model Performance Metrics on Test
         matrix = confusion_matrix(test['predicted'], test['h1n1_vaccine'])
         matrix

Out[20]: array([[4150,  903],
                [  62,  227]], dtype=int64)
```



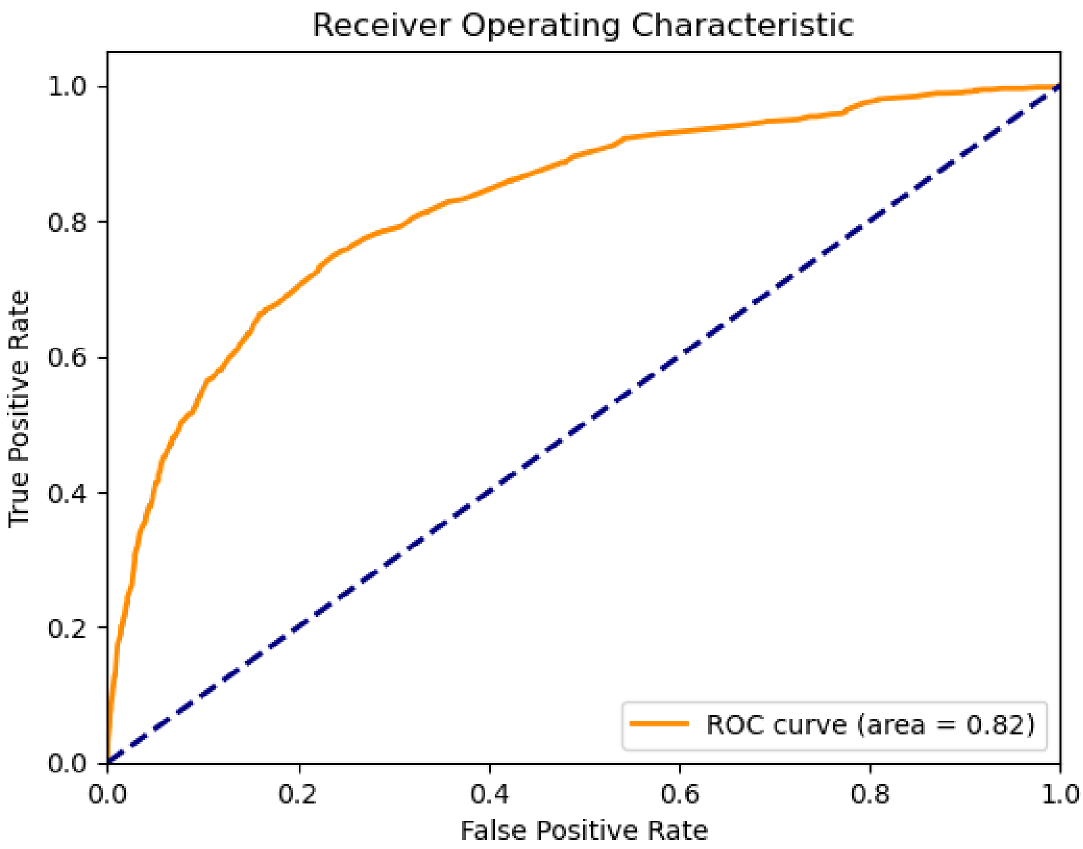
```
In [21]: from sklearn.metrics import classification_report

         # Model Performance classification report Train
         print(classification_report(test['predicted'], test['h1n1_vaccine']))

              precision    recall  f1-score   support

           0       0.99      0.82      0.90      5053
           1       0.20      0.79      0.32       289

    accuracy                           0.82      5342
   macro avg       0.59      0.80      0.61      5342
weighted avg       0.94      0.82      0.86      5342
```

➢ ROC Curve

Plot the ROC curve to visualize the model's performance.

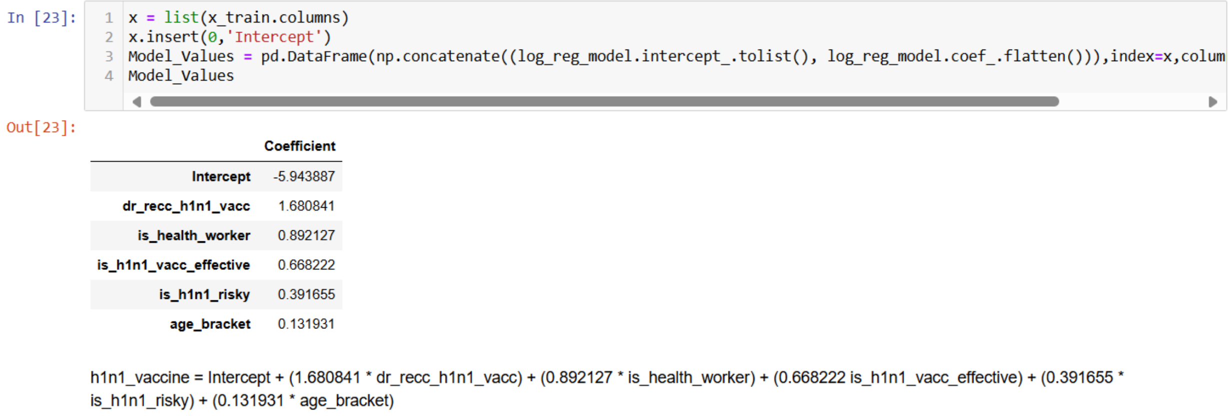


```
In [22]: # ROC Curve

         from sklearn.metrics import roc_curve

         fpr, tpr, thresholds = roc_curve(test['h1n1_vaccine'], test['probability_bad'])
         roc_auc = auc(fpr, tpr)

         plt.figure()
         plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' % roc_auc)
         plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
         plt.xlim([0.0, 1.0])
         plt.ylim([0.0, 1.05])
         plt.xlabel('False Positive Rate')
         plt.ylabel('True Positive Rate')
         plt.title('Receiver Operating Characteristic')
         plt.legend(loc="lower right")
         plt.show()
```



➢ Business / Client Submission

Prepare the model coefficients for business interpretation.



In [23]:

```
x = list(x_train.columns)
x.insert(0,'Intercept')
Model_Values = pd.DataFrame(np.concatenate((log_reg_model.intercept_.tolist(), log_reg_model.coef_.flatten())),index=x,colum
Model_Values
```

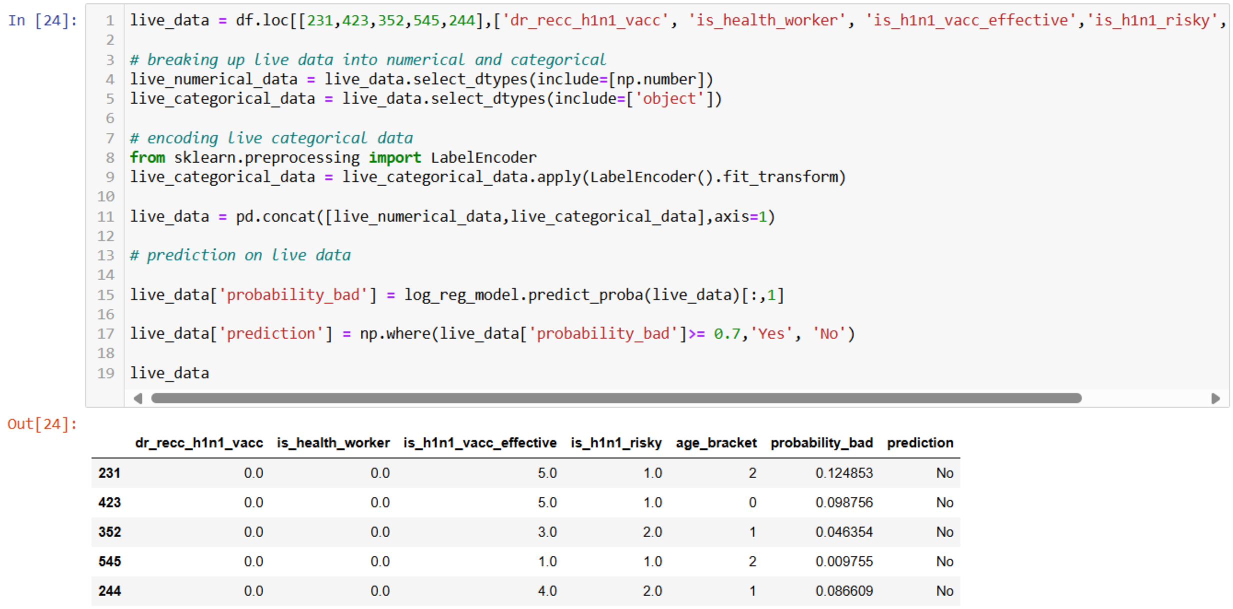
Out[23]:

	Coefficient
Intercept	-5.943887
dr_recc_h1n1_vacc	1.680841
is_health_worker	0.892127
is_h1n1_vacc_effective	0.668222
is_h1n1_risky	0.391655
age_bracket	0.131931

h1n1_vaccine = Intercept + (1.680841 * dr_recc_h1n1_vacc) + (0.892127 * is_health_worker) + (0.668222 is_h1n1_vacc_effective) + (0.391655 * is_h1n1_risky) + (0.131931 * age_bracket)

➢ Predicted on Live Data

Use the model to predict new data points.



In [24]:

```
live_data = df.loc[[231,423,352,545,244],['dr_recc_h1n1_vacc', 'is_health_worker', 'is_h1n1_vacc_effective','is_h1n1_risky',

# breaking up live data into numerical and categorical
live_numerical_data = live_data.select_dtypes(include=[np.number])
live_categorical_data = live_data.select_dtypes(include=['object'])

# encoding live categorical data
from sklearn.preprocessing import LabelEncoder
live_categorical_data = live_categorical_data.apply(LabelEncoder().fit_transform)

live_data = pd.concat([live_numerical_data,live_categorical_data],axis=1)

# prediction on live data

live_data['probability_bad'] = log_reg_model.predict_proba(live_data)[:,1]

live_data['prediction'] = np.where(live_data['probability_bad']>= 0.7,'Yes', 'No')

live_data
```

Out[24]:

	dr_recc_h1n1_vacc	is_health_worker	is_h1n1_vacc_effective	is_h1n1_risky	age_bracket	probability_bad	prediction
231	0.0	0.0	5.0	1.0	2	0.124853	No
423	0.0	0.0	5.0	1.0	0	0.098756	No
352	0.0	0.0	3.0	2.0	1	0.046354	No
545	0.0	0.0	1.0	1.0	2	0.009755	No
244	0.0	0.0	4.0	2.0	1	0.086609	No

Discussion

Feature Importance: The selected features (dr_recc_h1n1_vacc, is_health_worker, is_h1n1_vacc_effective, is_h1n1_risky, age_bracket) showed significant influence on vaccine uptake.

Model Interpretation: The logistic regression model indicated that recommendations from doctors (dr_recc_h1n1_vacc) and perceived vaccine effectiveness (is_h1n1_vacc_effective) were strong predictors.

Performance Evaluation: The model achieved consistent accuracy and demonstrated good sensitivity and specificity across training and testing datasets.

Limitations: Challenges included missing data imputation and potential biases in self-reported survey data, influencing model outcomes.

Conclusion

The logistic regression model effectively predicts H1N1 vaccine uptake with an overall accuracy of 82%.

Key predictors such as medical recommendations and perceived vaccine efficacy play crucial roles in predicting vaccination decisions.

This model can aid in understanding factors influencing vaccine acceptance and guide targeted public health strategies to improve vaccination rates.