```
# Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, precision_recall_curve
from sklearn.preprocessing import StandardScaler, PolynomialFeatures
from sklearn.feature_selection import SelectKBest, mutual_info_classif
from imblearn.over_sampling import SMOTE
import matplotlib.pyplot as plt
from google.colab import drive
from google.colab import files

# Mount Google Drive
drive.mount('/content/drive')

# Provide the path to your dataset in Google Drive
file_path = '/content/drive/MyDrive/booking.csv'

# Load the dataset
hotel_data = pd.read_csv(file_path)
print("Data Preview:\n", hotel_data.head())

# Data Preprocessing
# Convert 'date of reservation' to datetime format and extract useful features
hotel_data['date of reservation'] = pd.to_datetime(hotel_data['date of reservation'], errors='coerce')
hotel_data['reservation_month'] = hotel_data['date of reservation'].dt.month
hotel_data['reservation_year'] = hotel_data['date of reservation'].dt.year

# Convert 'booking status' to a binary target variable
hotel_data['cancellation_status'] = hotel_data['booking status'].apply(lambda x: 1 if x == 'Canceled' else 0)

# Encode categorical variables using one-hot encoding
hotel_data_encoded = pd.get_dummies(hotel_data, columns=['type of meal', 'room type', 'market segment type'], drop_first=True)

# Drop unnecessary columns and handle missing values
hotel_data_encoded.drop(columns=['date of reservation', 'booking status', 'Booking_ID'], inplace=True)
hotel_data_encoded = hotel_data_encoded.dropna()

# Separate features (X) and target (Y)
X = hotel_data_encoded.drop(['cancellation_status'], axis=1)
Y = hotel_data_encoded['cancellation_status']

# Handle class imbalance using SMOTE
sm = SMOTE(random_state=42)
X, Y = sm.fit_resample(X, Y)

# Split the 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)

# Feature scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Feature selection
selector = SelectKBest(mutual_info_classif, k=10)
X_train_selected = selector.fit_transform(X_train_scaled, y_train)
X_test_selected = selector.transform(X_test_scaled)

# Hyperparameter tuning
param_grid = {
    'C': [0.01, 0.1, 1, 10, 100],
    'penalty': ['l1', 'l2'],
    'solver': ['liblinear', 'saga']
}
grid_search = GridSearchCV(LogisticRegression(max_iter=500, random_state=42, class_weight='balanced'), param_grid, cv=5, scoring='f1')
grid_search.fit(X_train_selected, y_train)

# Best Logistic Regression model
best_model = grid_search.best_estimator_
print("Best Parameters:", grid_search.best_params_)

# Fit the model
best_model.fit(X_train_selected, y_train)

# Predict on the test set
y_pred = best_model.predict(X_test_selected)

# Evaluate the model
metrics = {
```

```
        "Accuracy": accuracy_score(y_test, y_pred),
        "Precision": precision_score(y_test, y_pred),
        "Recall": recall_score(y_test, y_pred),
        "F1 Score": f1_score(y_test, y_pred),
    }
    print("Logistic Regression Metrics:", metrics)

    # Precision-Recall Curve
    probs = best_model.predict_proba(X_test_selected)[:, 1]
    precisions, recalls, thresholds = precision_recall_curve(y_test, probs)

    plt.plot(thresholds, precisions[:-1], label="Precision")
    plt.plot(thresholds, recalls[:-1], label="Recall")
    plt.xlabel("Threshold")
    plt.ylabel("Score")
    plt.title("Precision-Recall Curve")
    plt.legend()
    plt.show()
```

```
Mounted at /content/drive
Data Preview:
   Booking_ID  number of adults  number of children  number of weekend nights  \
0   INN00001                 1                   1                         2
1   INN00002                 1                   0                         1
2   INN00003                 2                   1                         1
3   INN00004                 1                   0                         0
4   INN00005                 1                   0                         1

   number of week nights  type of meal  car parking space    room type  \
0                      5   Meal Plan 1                  0  Room_Type 1
1                      3  Not Selected                  0  Room_Type 1
2                      3   Meal Plan 1                  0  Room_Type 1
3                      2   Meal Plan 1                  0  Room_Type 1
4                      2  Not Selected                  0  Room_Type 1

   lead time market segment type  repeated  P-C  P-not-C  average price  \
0        224             Offline         0    0        0          88.00
1          5              Online         0    0        0         106.68
2          1              Online         0    0        0          50.00
3        211              Online         0    0        0         100.00
4         48              Online         0    0        0          77.00

   special requests date of reservation booking status
0                 0           10/2/2015   Not Canceled
```