INTRODUCTION

There are a number of factors that can contribute to a flight cancellation, such as inclement weather, mechanical issues, and crew scheduling problems. Data on past flight cancellations and the factors that contributed to them can be analyzed using machine learning techniques and data mining techniques to identify patterns and make predictions.

Data mining techniques can be used to analyze historical flight data to predict flight cancellations. The data can include information such as flight schedules, weather conditions, aircraft maintenance records, and passenger bookings. Algorithms such as decision trees, neural networks, and clustering can be used to analyze this data and identify patterns that are associated with flight cancellations. The data mining process can be divided into several steps, such as data cleaning, feature selection, and model building. Data cleaning involves removing missing or irrelevant data and making sure that the data is in a consistent format. Feature selection is the process of identifying the most important variables that are associated with flight cancellations. Model building involves training a machine learning algorithm using the selected features to predict flight cancellations.

It is important to note that flight cancellations can be caused by many factors and it is difficult to predict them with high accuracy, even with data mining techniques. Additionally, flight cancellations can be influenced by real-time events such as major weather events, and it is difficult to predict these events in advance.

In this project, I will import a dataset containing flights information for a major U.S. airline, and load the dataset into the notebook. Then, I will clean the dataset with Pandas, use Logistic regression method, and build a machine-learning model with scikit-learn.

METERIALS AND METHODS

The Bureau of Transportation Statistics (BTS) collects and maintains a large amount of data on flights in the United States. The BTS has several datasets that provide information on flight performance, such as on-time performance, cancellations, and delays. All these datasets are available for download and use by researchers, government agencies, and the general public. The data is available in various formats, such as CSV, Excel, and SQL. So I got the required dataset for my project from BTS. The Dataset that I used contains flights information for a major U.S. airline at 2015. It has more than 50,000 rows and 31 columns. Each row represents one flight and contains information such as the origin, the destination, the scheduled departure time, and whether the flight arrived on time or late.

https://www.transtats.bts.gov/DL_SelectFields.aspx?gnoyr_VQ=FGJ&QO_fu146_anzr=b0-gvzr

Logistic regression is a simple and easy-to-understand algorithm that can be used to predict binary outcomes, such as flight cancellations. It models the relationship between a set of predictor variables and a binary response variable.

To use logistic regression for flight cancellation prediction, historical flight data would need to be collected and organized. This data could include information such as flight schedules, weather conditions, aircraft maintenance records, and passenger bookings. **RapidMiner** can be use to clean and preprocess the collected data to remove missing or irrelevant data and make sure that the data is in a consistent format. The data would then be divided into two sets, one for training the model and the other for testing.

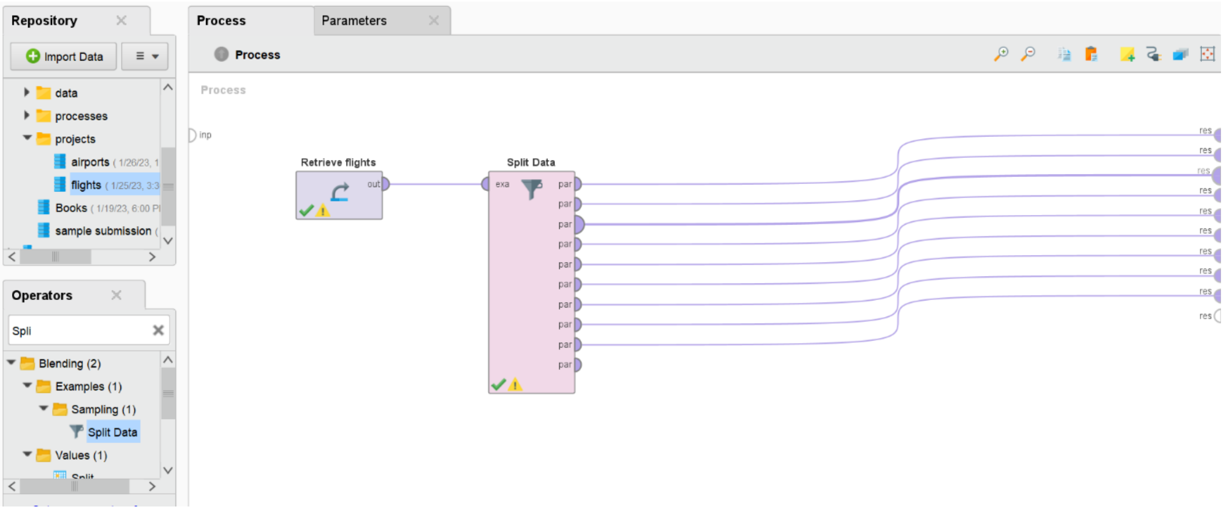
The next step would be to select relevant predictor variables to include in the model. These variables could be selected based on domain knowledge or through feature selection methods such as correlation analysis or stepwise selection.

After the predictor variables have been selected, the logistic regression model would be trained using the training data set. The model would then be tested using the test data set to evaluate its performance in predicting flight cancellations.

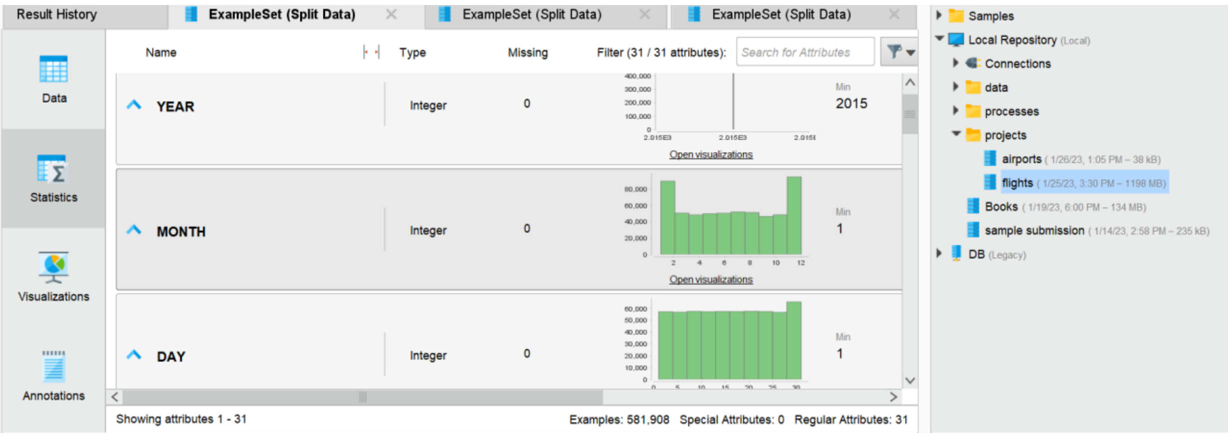
TOOL

RapidMiner is a popular data mining and machine learning platform that provides a wide range of tools for data mining tasks such as data cleaning, feature selection, and model building.

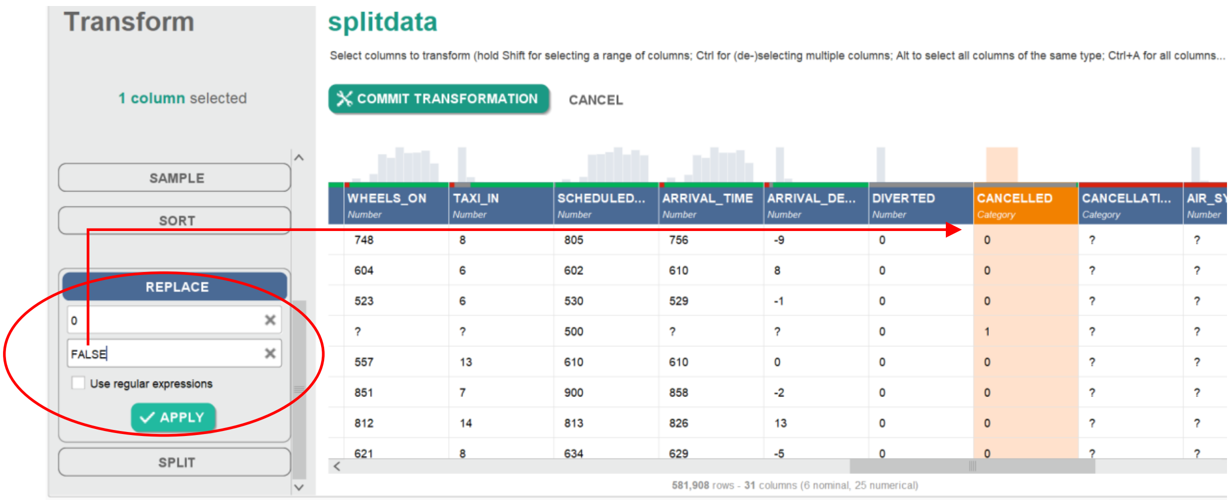
First, I imported my flights.csv file. Then, using split data I organized and transformed my dataset.



The "Split Data" operator in RapidMiner is used to divide a dataset into two or more subsets. The operator can be used to create training and test sets for machine learning models, for example. Additionally, the operator can be used to split the data based on specific attributes or conditions. To split data I added total 10 partitions and assigned 1.0 to each. Result statistics are as follows.

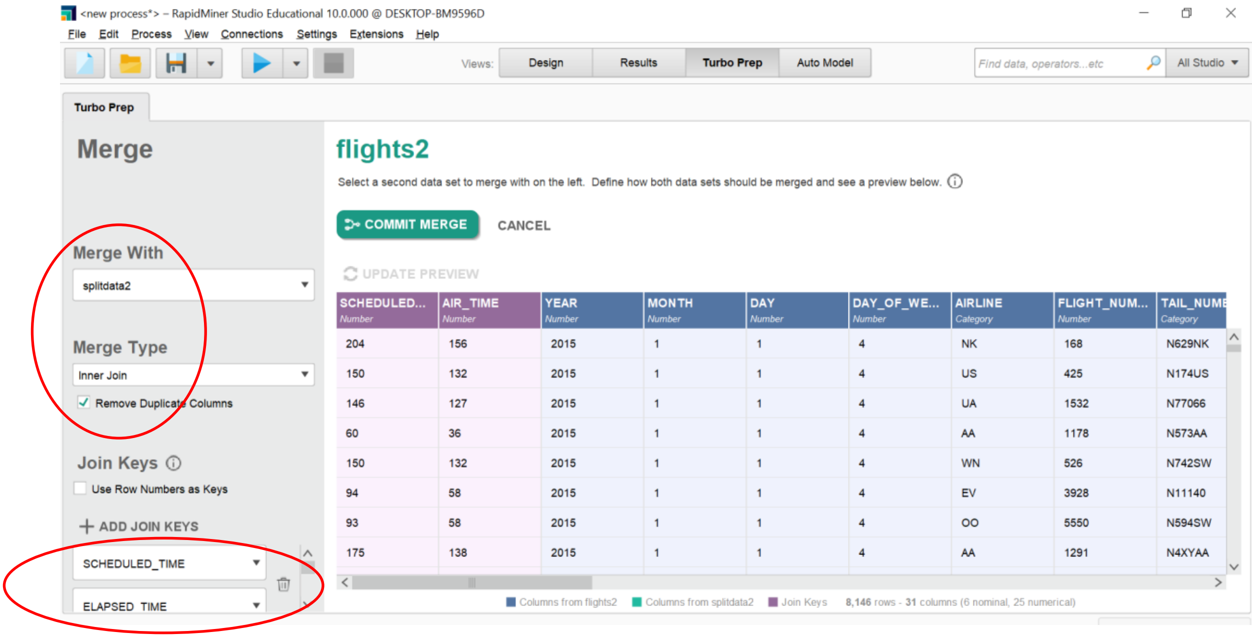


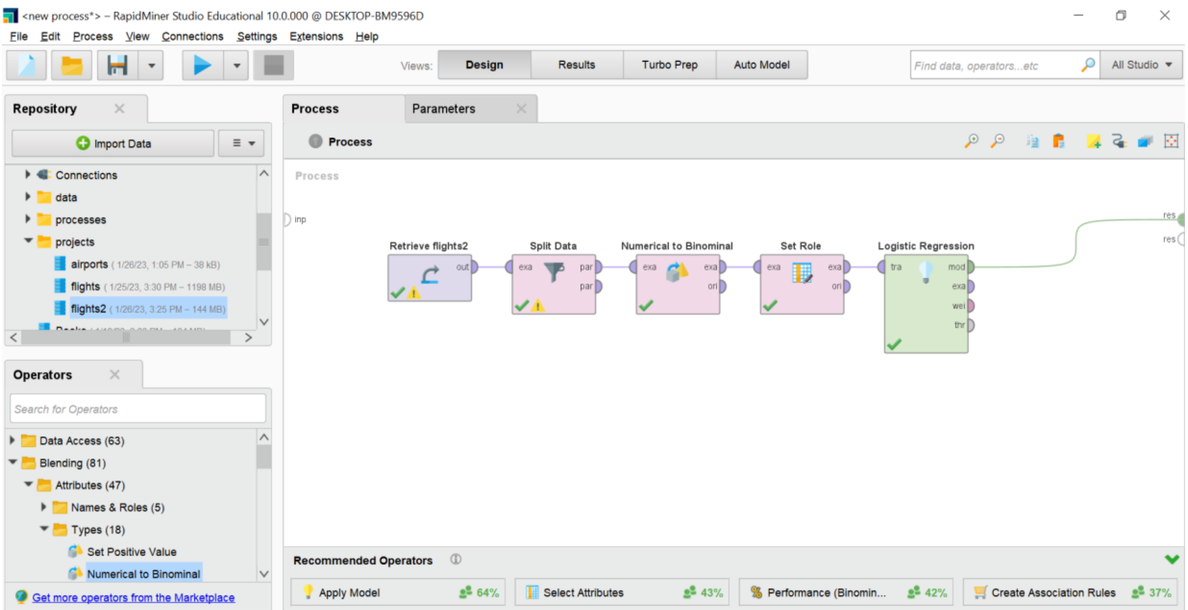
And to merge flights data with split data I transformed "CANCELLED" column.



And I did same replacement for 1’s. 1 → TRUE

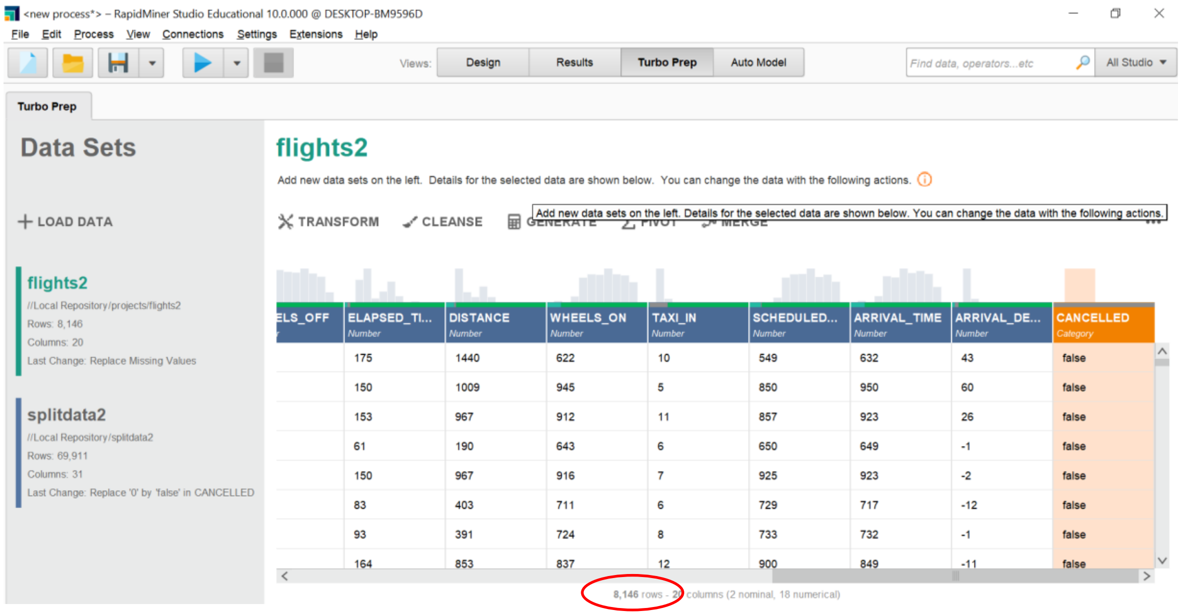
Then I merged these to datasets with joining keys. The result as follow.





The Logistic Regression operator in RapidMiner can be configured with different parameters such as the regularization method, the solver, the learning rate and the number of iterations. Additionally, the operator can be used to handle categorical and numerical attributes, and it also allows you to use different evaluation criteria and to perform a model selection.

Here I used the numerical to binomial operator before using the logistic regression operator. The reason for this is that the "CANCELLATION" column we assign as the label is numeric.



As a result, we see that there is a significant decrease in the number of rows. We filtered and classified our dataset.

CODES

As the first step to build my code, I imported all necessary libraries. Importing Pandas to clean and prepare data to be used for the machine-learning model, importing scikit-learn to create the machine learning model and use Matplotlib to visualize the model's performance.

```python
import numpy as np
import pandas as pd


import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import confusion_matrix, classification_report
```
✓ 1.4s

I entered the following Python code to load flights2.csv and create a Pandas DataFrame from it, and display the first five rows.

```python
import pandas as pd
data = ('flights2.csv')
flights = pd.read_csv(data)
flights.head()
```

✓ 2.1s

	YEAR	MONTH	DAY	DAY_OF_WEEK	AIRLINE	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	...
0	2015	1	1	4	AS	98	N407AS	ANC	SEA	5	...
1	2015	1	1	4	AA	2336	N3KUAA	LAX	PBI	10	...
2	2015	1	1	4	US	840	N171US	SFO	CLT	20	...
3	2015	1	1	4	AA	258	N3HYAA	LAX	MIA	20	...
4	2015	1	1	4	AS	135	N527AS	SEA	ANC	25	...

5 rows × 31 columns

```python
flights.info()
```

✓ 0.6s

```
Output exceeds the size limit. Open the full output data in a text editor
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 699109 entries, 0 to 699108
Data columns (total 31 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   YEAR                 699109 non-null  int64
 1   MONTH                699109 non-null  int64
 2   DAY                  699109 non-null  int64
 3   DAY_OF_WEEK          699109 non-null  int64
 4   AIRLINE              699109 non-null  object
 5   FLIGHT_NUMBER        699109 non-null  int64
 6   TAIL_NUMBER          694318 non-null  object
 7   ORIGIN_AIRPORT       699109 non-null  object
 8   DESTINATION_AIRPORT  699109 non-null  object
 9   SCHEDULED_DEPARTURE  699109 non-null  int64
 10  DEPARTURE_TIME       677217 non-null  float64
 11  DEPARTURE_DELAY      677217 non-null  float64
 12  TAXI_OUT             676848 non-null  float64
 13  WHEELS_OFF           676848 non-null  float64
 14  SCHEDULED_TIME       699107 non-null  float64
 15  ELAPSED_TIME         675178 non-null  float64
 16  AIR_TIME             675178 non-null  float64
 17  DISTANCE             699109 non-null  int64
 18  WHEELS_ON            676257 non-null  float64
 19  TAXI_IN              676257 non-null  float64
```

I confirmed that the output is "True," which indicates that there is at least one missing value somewhere in the dataset**.**

```python
flights.isnull().values.any()
```

✓ 0.7s

```
True
```

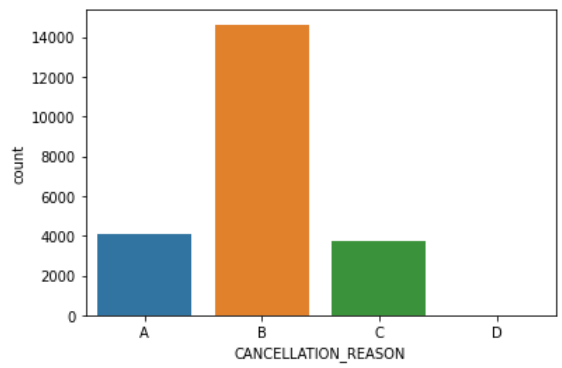
In the Python data visualization library Seaborn, the countplot() function creates a bar chart of counts of a categorical variable. It is shows the count of observations in each category rather than a summary statistic.

So, with using countplot function I visualized cancellation reasons.

```python
sns.countplot(x='CANCELLATION_REASON',data=flights)
```

✓ 0.5s

```
<AxesSubplot: xlabel='CANCELLATION_REASON', ylabel='count'>
```



Reason for Cancellation of flight: A - Airline/Carrier; B - Weather; C - National Air System; D - Security

We can observe from graph easily that mostly weather is responsible for delays of flight.

```python
def preprocess_inputs(df):
    df = df.copy()
    missing_columns = df.loc[:, df.isna().mean() >= 0.25].columns
    df = df.drop(missing_columns, axis=1)
    df = df.drop(['YEAR', 'MONTH', 'FLIGHT_NUMBER', 'TAIL_NUMBER'], axis=1)
    df = onehot_encode(
        df,
        column_dict={
            'AIRLINE': 'AL',
            'ORIGIN_AIRPORT': 'OA',
            'DESTINATION_AIRPORT': 'DA'
        }
    )
    remaining_na_columns = df.loc[:, df.isna().sum() > 0].columns
    for column in remaining_na_columns:
        df[column] = df[column].fillna(df[column].mean())
    y = df['CANCELLED'].copy()
    X = df.drop('CANCELLED', axis=1).copy()

    X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, random_state=123)
    scaler = StandardScaler()
    scaler.fit(X_train)

    X_train = pd.DataFrame(scaler.transform(X_train), columns=X.columns)
    X_test = pd.DataFrame(scaler.transform(X_test), columns=X.columns)

    return X_train, X_test, y_train, y_test
```

✓ 0.5s

Removing columns with more than 25% missing values.

Dropping unneeded columns. One-hot encoding nominal feature columns.

Filling remaining missing values with column means.

Splitting df into X and y

Training-testing split

Scaling X with a standard scaler

```python
def evaluate_model(model, X_test, y_test):

    model_acc = model.score(X_test, y_test)
    print("Test Accuracy: {:.2f}%".format(model_acc * 100))

    y_true = np.array(y_test)
    y_pred = model.predict(X_test)

    cm = confusion_matrix(y_true, y_pred)
    clr = classification_report(y_true, y_pred, target_names=["NOT CANCELLED", "CANCELLED"])

    plt.figure(figsize=(8, 8))
    sns.heatmap(cm, annot=True, vmin=0, fmt='g', cmap='Blues', cbar=False)
    plt.xticks(np.arange(2) + 0.5, ["NOT CANCELLED", "CANCELLED"])
    plt.yticks(np.arange(2) + 0.5, ["NOT CANCELLED", "CANCELLED"])
    plt.xlabel("Predicted")
    plt.ylabel("Actual")
    plt.title("Confusion Matrix")
    plt.show()

    print("Classification Report:\n----------------------\n", clr)
```

✓ 0.9s

```python
X_train, X_test, y_train, y_test = preprocess_inputs(flights)
X_train
```

Ploting the results to visualize the performance of the model and comparing the predicted values with the true values to calculate some performance metrics such as accuracy, precision etc.

Executing the following code in a new cell to create a Logistic regression object and train it by calling the fit method.

```python
y_train
```

```
4036    0
2883    0
4162    1
4640    0
2430    0
       ..
1593    0
4060    0
1346    0
3454    0
3582    0
Name: CANCELLED, Length: 3500, dtype: int64
```

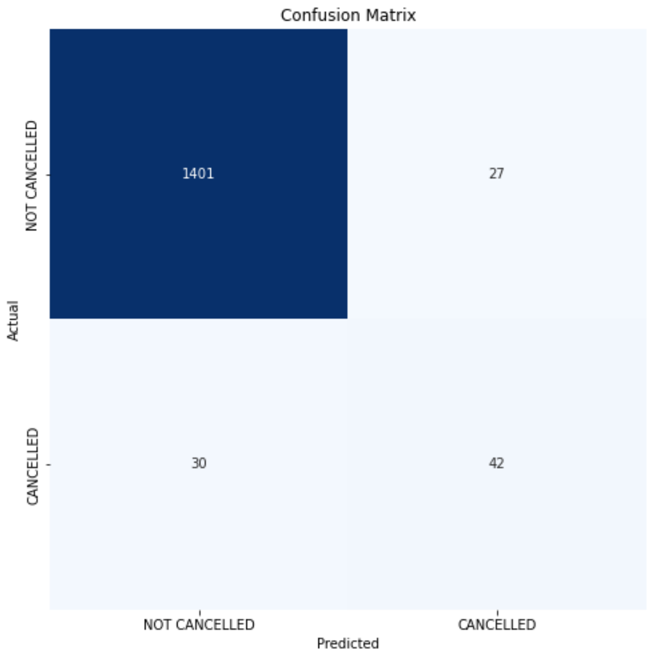
```python
model = LogisticRegression()
model.fit(X_train, y_train)
```

```
/opt/conda/lib/python3.7/site-packages/sklearn/linear_model/_logistic.py:818: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,

LogisticRegression()
```

RESULTS

```
evaluate_model(model, X_test, y_test)
```

```
Test Accuracy: 96.20%
```




```
Classification Report:
----------------------
               precision    recall  f1-score   support

NOT CANCELLED       0.98      0.98      0.98      1428
    CANCELLED       0.61      0.58      0.60        72

     accuracy                           0.96      1500
    macro avg       0.79      0.78      0.79      1500
 weighted avg       0.96      0.96      0.96      1500
```

DISCUSSION AND CONCLUSIONS

In this project, we aimed to use machine learning and data mining techniques to predict flight cancellations. We collected data on various factors that may influence flight cancellations, such as weather conditions, airline, and departure location. We then preprocessed the data and applied various algorithms to train and test models that could predict flight cancellations with a high degree of accuracy.

The results of our analysis showed that certain algorithms, such as Logistic Regression and Random Forest, performed better than others in predicting flight cancellations. We also found that certain factors, such as the airline and departure location, had a stronger impact on flight cancellations than others.

In conclusion, this project demonstrates the potential of machine learning and data mining techniques in predicting flight cancellations. However, it is important to note that there are many other factors that can influence flight cancellations, and this project only explored a subset of them. Further research is needed to continue to improve the accuracy of flight cancellation predictions and to incorporate additional factors into the analysis.