Machine learning project-Pegah Karimi_GH1019718

March 16, 2022

1 Using Python In Costumer Satisfaction Analysis:

This Dataset was provided by an airline company. Due to numerous reasons, the company's true name is not released , which is why the name "Invistico" airlines was chosen. Customers who have already flown with them are included in the dataset. Customers' feedback on a variety of topics, as well as their flight information, has been compiled.

The primary goal of this dataset is to forecast if a future customer will be satisfied with their service based on the values of the other criteria.

1.1 Business Problem

As Airlines should pay attention to the factores which affect on their costumer satisfaction,in this project i am going to estimate Customer Satisfaction .

1.2 Objective

My goal is to figure out what figures have the greatest impact on costumer (passengers) satisfaction and to develop a model that can predict whether or not a certain passenger will be satisfied with the airline services in the future or not .

1.2.1 Pipeline Structure

My structure is based on a typical data science pipeline which contains : 1-Obtaining the data. 2-cleaning the data is the next step. 3-Exploring the data. 4-Modeling the data. 5-INterpreting the data is last.

**Note: The data was found from the "Human Resources Analytics" dataset provided by Kaggle's website. https://www.kaggle.com/ludobenistant/hr-analytics

*Note: THIS DATASET IS SIMULATED.

1.2.2 Step 1: Importing the Library

[1]:
```
# Importing Necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as matplot
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report,␣
↪precision_score, recall_score, confusion_matrix, precision_recall_curve
from sklearn.preprocessing import RobustScaler
from sklearn.metrics import accuracy_score
from sklearn.naive_bayes import GaussianNB
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix
import sklearn.linear_model
from sklearn.datasets import load_iris
from sklearn.linear_model import LogisticRegression
import sklearn.metrics
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import ShuffleSplit
from sklearn.model_selection import cross_val_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
```

1.2.3 Step2-Loading the Dataset

[2]:
```
df = pd.read_csv("../../datasets/Invistico_Airline.csv")
df.head()
```

[2]:

	satisfaction	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Seat comfort	Departure/Arrival time convenient
0	satisfied	Female	Loyal Customer	65	Personal Travel	Eco	265	0	0
1	satisfied	Male	Loyal Customer	47	Personal Travel	Business	2464	0	0
2	satisfied	Female	Loyal Customer	15	Personal Travel	Eco	2138	0	0
3	satisfied	Female	Loyal Customer	60	Personal Travel	Eco	623	0	0
4	satisfied	Female	Loyal Customer	70	Personal Travel	Eco			

```
4              354             0                                    0

   Food and drink  …  Online support  Ease of Online booking  \
0               0  …               2                       3
1               0  …               2                       3
2               0  …               2                       2
3               0  …               3                       1
4               0  …               4                       2

   On-board service  Leg room service  Baggage handling  Checkin service  \
0                 3                 0                 3                5
1                 4                 4                 4                2
2                 3                 3                 4                4
3                 1                 0                 1                4
4                 2                 0                 2                4

   Cleanliness  Online boarding  Departure Delay in Minutes  \
0            3                2                           0
1            3                2                         310
2            4                2                           0
3            1                3                           0
4            2                5                           0

   Arrival Delay in Minutes
0                       0.0
1                     305.0
2                       0.0
3                       0.0
4                       0.0

[5 rows x 23 columns]
```

1.2.4 Step3-Cleaning the Dataset

Although the data set that I took from kaggle is clean but , I will need to review the dataset to ensure that everything else is understandable and that the observation values correspond to the feature names.So Ichecked it in two ways.

[3]:
```
df.isnull().any()
```

[3]:
```
satisfaction                     False
Gender                           False
Customer Type                    False
Age                              False
Type of Travel                   False
Class                            False
```

```
Flight Distance                      False
Seat comfort                         False
Departure/Arrival time convenient    False
Food and drink                       False
Gate location                        False
Inflight wifi service                False
Inflight entertainment               False
Online support                       False
Ease of Online booking               False
On-board service                     False
Leg room service                     False
Baggage handling                     False
Checkin service                      False
Cleanliness                          False
Online boarding                      False
Departure Delay in Minutes           False
Arrival Delay in Minutes              True
dtype: bool
```

[4]:
```
df = df.dropna()
```

[5]:
```
df.isnull().sum()
```

[5]:
```
satisfaction                         0
Gender                               0
Customer Type                        0
Age                                  0
Type of Travel                       0
Class                                0
Flight Distance                      0
Seat comfort                         0
Departure/Arrival time convenient    0
Food and drink                       0
Gate location                        0
Inflight wifi service                0
Inflight entertainment               0
Online support                       0
Ease of Online booking               0
On-board service                     0
Leg room service                     0
Baggage handling                     0
Checkin service                      0
Cleanliness                          0
Online boarding                      0
Departure Delay in Minutes           0
Arrival Delay in Minutes             0
dtype: int64
```

1.2.5 step4-Get a quick overview of what we are dealing with in our dataset

[6]:
```
df.head()
```

[6]:

	satisfaction	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Seat comfort	Departure/Arrival time convenient	Food and drink	…	Online support	Ease of Online booking	On-board service	Leg room service	Baggage handling	Checkin service	Cleanliness	Online boarding	Departure Delay in Minutes	Arrival Delay in Minutes
0	satisfied	Female	Loyal Customer	65	Personal Travel	Eco	265	0	0	0	…	2	3	3	0	3	5	3	2	0	0.0
1	satisfied	Male	Loyal Customer	47	Personal Travel	Business	2464	0	0	0	…	2	3	4	4	4	2	3	2	310	305.0
2	satisfied	Female	Loyal Customer	15	Personal Travel	Eco	2138	0	0	0	…	2	2	3	3	4	4	4	2	0	0.0
3	satisfied	Female	Loyal Customer	60	Personal Travel	Eco	623	0	0	0	…	3	1	1	0	1	4	1	3	0	0.0
4	satisfied	Female	Loyal Customer	70	Personal Travel	Eco	354	0	0	0	…	4	2	2	0	2	4	2	5	0	0.0

[5 rows x 23 columns]

[7]:
```
#As I wanted to check the affect of features on satisfaction i bring it to the␣
 ↪first column
front = df['satisfaction']
df.drop(labels=['satisfaction'], axis=1,inplace = True)
df.insert(0, 'satisfaction', front)
df.head()
```

[7]:

	satisfaction	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Seat comfort	Departure/Arrival time convenient	Food and drink	…	Online support	Ease of Online booking	On-board service	Leg room service	Baggage handling	Checkin service	Cleanliness	Online boarding	Departure Delay in Minutes	Arrival Delay in Minutes
0	satisfied	Female	Loyal Customer	65	Personal Travel	Eco	265	0	0	0	…	2	3	3	0	3	5	3	2	0	0.0
1	satisfied	Male	Loyal Customer	47	Personal Travel	Business	2464	0	0	0	…	2	3	4	4	4	2	3	2	310	305.0
2	satisfied	Female	Loyal Customer	15	Personal Travel	Eco	2138	0	0	0	…	2	2	3	3	4	4	4	2	0	0.0
3	satisfied	Female	Loyal Customer	60	Personal Travel	Eco	623	0	0	0	…	3	1	1	0	1	4	1	3	0	0.0
4	satisfied	Female	Loyal Customer	70	Personal Travel	Eco	354	0	0	0	…	4	2	2	0	2	4	2	5	0	

```
4                       0.0

[5 rows x 23 columns]
```

1.2.6 Step5:Exploring the Data

5-1-The statistical overview showes that in this Dataset we have 23 columns and around 130,000 Obsewrvations.

[8]:
```
df.shape
```

[8]:
```
(129487, 23)
```

5-2-Type of our features must be checked.

[9]:
```
df.dtypes
```

[9]:
```
satisfaction                        object
Gender                              object
Customer Type                       object
Age                                  int64
Type of Travel                      object
Class                               object
Flight Distance                      int64
Seat comfort                         int64
Departure/Arrival time convenient    int64
Food and drink                       int64
Gate location                        int64
Inflight wifi service                int64
Inflight entertainment               int64
Online support                       int64
Ease of Online booking               int64
On-board service                     int64
Leg room service                     int64
Baggage handling                     int64
Checkin service                      int64
Cleanliness                          int64
Online boarding                      int64
Departure Delay in Minutes           int64
Arrival Delay in Minutes           float64
dtype: object
```

5-3-Count the satisfied and unsatisfied passengers

5-4-The statistical review of the Dataset

[10]:
```
df.describe()
```

[10]:

	Age	Flight Distance	Seat comfort
count	129487.000000	129487.000000	129487.000000
mean	39.428761	1981.008974	2.838586
std	15.117597	1026.884131	1.392873
min	7.000000	50.000000	0.000000
25%	27.000000	1359.000000	2.000000
50%	40.000000	1924.000000	3.000000
75%	51.000000	2543.000000	4.000000
max	85.000000	6951.000000	5.000000

	Departure/Arrival time convenient	Food and drink	Gate location
count	129487.000000	129487.000000	129487.000000
mean	2.990277	2.852024	2.990377
std	1.527183	1.443587	1.305917
min	0.000000	0.000000	0.000000
25%	2.000000	2.000000	2.000000
50%	3.000000	3.000000	3.000000
75%	4.000000	4.000000	4.000000
max	5.000000	5.000000	5.000000

	Inflight wifi service	Inflight entertainment	Online support
count	129487.000000	129487.000000	129487.000000
mean	3.249160	3.383745	3.519967
std	1.318765	1.345959	1.306326
min	0.000000	0.000000	0.000000
25%	2.000000	2.000000	3.000000
50%	3.000000	4.000000	4.000000
75%	4.000000	4.000000	5.000000
max	5.000000	5.000000	5.000000

	Ease of Online booking	On-board service	Leg room service
count	129487.000000	129487.000000	129487.000000
mean	3.472171	3.465143	3.486118
std	1.305573	1.270755	1.292079
min	0.000000	0.000000	0.000000
25%	2.000000	3.000000	2.000000
50%	4.000000	4.000000	4.000000
75%	5.000000	4.000000	5.000000
max	5.000000	5.000000	5.000000

	Baggage handling	Checkin service	Cleanliness	Online boarding
count	129487.000000	129487.000000	129487.000000	129487.000000
mean	3.695460	3.340729	3.705886	3.352545
std	1.156487	1.260561	1.151683	1.298624
min	1.000000	0.000000	0.000000	0.000000

```
25%            3.000000          3.000000          3.000000          2.000000
50%            4.000000          3.000000          4.000000          4.000000
75%            5.000000          4.000000          5.000000          4.000000
max            5.000000          5.000000          5.000000          5.000000
```

```
       Departure Delay in Minutes  Arrival Delay in Minutes
count                129487.000000             129487.000000
mean                     14.643385                 15.091129
std                      37.932867                 38.465650
min                       0.000000                  0.000000
25%                       0.000000                  0.000000
50%                       0.000000                  0.000000
75%                      12.000000                 13.000000
max                    1592.000000               1584.000000
```

5-5-Changing the name of variable into easier names

[11]:
```
from sklearn.preprocessing import LabelEncoder     #Converting categorical data
LE = LabelEncoder()
df["Gender_cat"] = LE.fit_transform(df["Gender"])
df["Customer Type_cat"] = LE.fit_transform(df["Customer Type"])
df["Type of Travel_cat"] = LE.fit_transform(df["Type of Travel"])
df["Class_cat"] = LE.fit_transform(df["Class"])
```

[12]:
```
df.drop(["Gender", "Customer Type", "Type of Travel", "Class"], axis=1,␣
↪inplace=True)
```

[13]:
```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy="most_frequent")
imputer.fit(df)
df[:] = imputer.transform(df)
```

1.2.7 Step6- Splitting the dataset into 70% train, 30% test

[14]:
```
y=df.satisfaction
X=df.drop(["satisfaction"],axis="columns")
```

[15]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30,␣
↪random_state = 42)
```

1.2.8 step7-Predicting and Testing with diffrent Algorithms:

1-Training the model with the Decision Tree Algorithm

[16]:
```
from sklearn.tree import DecisionTreeClassifier
dt_model = DecisionTreeClassifier()
dt_model.fit(X_train, y_train)
```

[16]:
```
DecisionTreeClassifier()
```

[17]:
```
#Classification of the model with test data
y_test_dt = dt_model.predict(X_test)
```

[18]:
```
#Demonstrating the classification prediction success of the model
from sklearn.metrics import accuracy_score
acc_dt = accuracy_score(y_test, y_test_dt)
acc_dt
```

[18]:
```
0.9382449095168224
```

2-Training the model with the KNN Algorithm

[19]:
```
from sklearn.neighbors import KNeighborsClassifier
knn_model = KNeighborsClassifier(n_neighbors=5, metric="minkowski", p=2)
knn_model.fit(X_train, y_train)
```

[19]:
```
KNeighborsClassifier()
```

[20]:
```
#Classification of the model with test data
y_test_knn = knn_model.predict(X_test)
```

[21]:
```
#Demonstrating the classification prediction success of the model
acc_knn = accuracy_score(y_test, y_test_knn)
acc_knn
```

[21]:
```
0.6999768321878137
```

3-Training the model with Naive Bayes Algorithm

[22]:
```
from sklearn.naive_bayes import GaussianNB
nb_model = GaussianNB()
nb_model.fit(X_train, y_train)
```

[22]:
```
GaussianNB()
```

[23]:
```
#Classification of the model with test data
y_test_nb = nb_model.predict(X_test)
```

[24]:
```
#Demonstrating the classification prediction success of the model
from sklearn.metrics import accuracy_score
acc_nb = accuracy_score(y_test, y_test_nb)
acc_nb
```

[24]: 0.8184930625273509

[25]:
```
#Loading the Logistic Regression and showing the prediction accuracy
model = sklearn.linear_model.LogisticRegression()
model.fit(X_train, y_train)
```

```
/usr/local/lib/python3.8/dist-packages/sklearn/linear_model/_logistic.py:763:
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
  n_iter_i = _check_optimize_result(
```

[25]: LogisticRegression()

[26]:
```
y_predicted = model.predict(X_test)
accuracy = sklearn.metrics.accuracy_score(y_test, y_predicted)
accuracy
```

[26]: 0.7906916879038278

The results showes that we get the best result through the Decision Tree model:

dont forget to write the results in text?-hyper promiter

[32]:
```
def find_best_model_using_gridsearchcv(X,y):
        algos={
         'NaiveBais': { 'model': GaussianNB(),
                       'params': {'var_smoothing': np.logspace(0,-9, num=100)␣
 ↪}},



        'decision_tree': {
        'model': DecisionTreeClassifier(),
        'params': {
            #'min_samples_leaf': ['int','float'],
```

```
            'max_features' : ['auto','sqrt','log2' ]

            }
        }}
        scores = []
        cv = ShuffleSplit(n_splits=5, test_size=0.2, random_state=0)
        for algo_name, config in algos.items():
            gs = GridSearchCV(config['model'], config['params'], cv=cv,␣
↪return_train_score=False)
            gs.fit(X,y)
            scores.append({
            'model': algo_name,
            'best_score': gs.best_score_,
            'best_params': gs.best_params_
            })



        return pd.DataFrame(scores,columns=['model','best_score','best_params'])



find_best_model_using_gridsearchcv(X,y)
```

[32]:

	model	best_score	best_params
0	NaiveBais	0.815916	{'var_smoothing': 1e-09}
1	decision_tree	0.923577	{'max_features': 'log2'}

'NaiveBais': { 'model': GaussianNB(), 'params': { 'C':[0.05,0.1], 'gamma':[0.1,0.2], 'kernel':['rbf','linear'] } },

```
    'KNeighborsClassifier': {
    'model':  KNeighborsClassifier(),
    'params': {
        'weights':['uniform','distance'],
        'algorithm': ['auto', 'ball_tree']
    }
    },
```

[ ]: