Assignment 4

February 3, 2021

---

*You are currently looking at **version 1.1** of this notebook. To download notebooks and datafiles, as well as get help on Jupyter notebooks in the Coursera platform, visit the Jupyter Notebook FAQ course resource.*

```
In [ ]:
```

0.1 Assignment 4 - Understanding and Predicting Property Maintenance Fines

This assignment is based on a data challenge from the Michigan Data Science Team (MDST).

The Michigan Data Science Team (MDST) and the Michigan Student Symposium for Interdisciplinary Statistical Sciences (MSSISS) have partnered with the City of Detroit to help solve one of the most pressing problems facing Detroit - blight. Blight violations are issued by the city to individuals who allow their properties to remain in a deteriorated condition. Every year, the city of Detroit issues millions of dollars in fines to residents and every year, many of these fines remain unpaid. Enforcing unpaid blight fines is a costly and tedious process, so the city wants to know: how can we increase blight ticket compliance?

The first step in answering this question is understanding when and why a resident might fail to comply with a blight ticket. This is where predictive modeling comes in. For this assignment, your task is to predict whether a given blight ticket will be paid on time.

All data for this assignment has been provided to us through the Detroit Open Data Portal. **Only the data already included in your Coursera directory can be used for training the model for this assignment.** Nonetheless, we encourage you to look into data from other Detroit datasets to help inform feature creation and model selection. We recommend taking a look at the following related datasets:

- Building Permits
- Trades Permits
- Improve Detroit: Submitted Issues
- DPD: Citizen Complaints
- Parcel Map

We provide you with two data files for use in training and validating your models: train.csv and test.csv. Each row in these two files corresponds to a single blight ticket, and includes information about when, why, and to whom each ticket was issued. The target variable is compliance, which is True if the ticket was paid early, on time, or within one month of the hearing data, False if the ticket was paid after the hearing date or not at all, and Null if the violator was found not responsible. Compliance, as well as a handful of other variables that will not be available at test-time, are only included in train.csv.

Note: All tickets where the violators were found not responsible are not considered during evaluation. They are included in the training set as an additional source of data for visualization, and to enable unsupervised and semi-supervised approaches. However, they are not included in the test set.

File descriptions (Use only this data for training your model!)

```
readonly/train.csv - the training set (all tickets issued 2004-2011)
readonly/test.csv - the test set (all tickets issued 2012-2016)
readonly/addresses.csv & readonly/latlons.csv - mapping from ticket id to addresses
 Note: misspelled addresses may be incorrectly geolocated.
```

Data fields

train.csv & test.csv

```
ticket_id - unique identifier for tickets
agency_name - Agency that issued the ticket
inspector_name - Name of inspector that issued the ticket
violator_name - Name of the person/organization that the ticket was issued to
violation_street_number, violation_street_name, violation_zip_code - Address where
mailing_address_str_number, mailing_address_str_name, city, state, zip_code, non_us
ticket_issued_date - Date and time the ticket was issued
hearing_date - Date and time the violator's hearing was scheduled
violation_code, violation_description - Type of violation
disposition - Judgment and judgement type
fine_amount - Violation fine amount, excluding fees
admin_fee - $20 fee assigned to responsible judgments
```

state_fee - $10 fee assigned to responsible judgments late_fee - 10% fee assigned to responsible judgments discount_amount - discount applied, if any clean_up_cost - DPW clean-up or graffiti removal cost judgment_amount - Sum of all fines and fees grafitti_status - Flag for graffiti violations

train.csv only

```
payment_amount - Amount paid, if any
payment_date - Date payment was made, if it was received
payment_status - Current payment status as of Feb 1 2017
balance_due - Fines and fees still owed
collection_status - Flag for payments in collections
compliance [target variable for prediction]
 Null = Not responsible
 0 = Responsible, non-compliant
 1 = Responsible, compliant
compliance_detail - More information on why each ticket was marked compliant or nor
```

0.2 Evaluation

Your predictions will be given as the probability that the corresponding blight ticket will be paid on time.

The evaluation metric for this assignment is the Area Under the ROC Curve (AUC).

Your grade will be based on the AUC score computed for your classifier. A model which with an AUROC of 0.7 passes this assignment, over 0.75 will recieve full points. ___

For this assignment, create a function that trains a model to predict blight ticket compliance in Detroit using `readonly/train.csv`. Using this model, return a series of length 61001 with the data being the probability that each corresponding ticket from `readonly/test.csv` will be paid, and the index being the ticket_id.

Example:

```
ticket_id
   284932    0.531842
   285362    0.401958
   285361    0.105928
   285338    0.018572
             ...
   376499    0.208567
   376500    0.818759
   369851    0.018528
   Name: compliance, dtype: float32
```

0.2.1 Hints

- Make sure your code is working before submitting it to the autograder.
- Print out your result to see whether there is anything weird (e.g., all probabilities are the same).
- Generally the total runtime should be less than 10 mins. You should NOT use Neural Network related classifiers (e.g., MLPClassifier) in this question.
- Try to avoid global variables. If you have other functions besides blight_model, you should move those functions inside the scope of blight_model.
- Refer to the pinned threads in Week 4's discussion forum when there is something you could not figure it out.

```
In [1]: import pandas as pd
        df = pd.read_csv('train.csv',encoding = "ISO-8859-1")

        df.compliance = df.compliance.fillna(value=-1)
        df = df[df.compliance != -1]

        df.head(10)
```

```
/opt/conda/lib/python3.6/site-packages/IPython/core/interactiveshell.py:2717: Dtype
  interactivity=interactivity, compiler=compiler, result=result)
```

```
Out[1]:     ticket_id                                         agency_name  \
        0       22056  Buildings, Safety Engineering & Env Department
        1       27586  Buildings, Safety Engineering & Env Department
        5       22046  Buildings, Safety Engineering & Env Department
        6       18738  Buildings, Safety Engineering & Env Department
        7       18735  Buildings, Safety Engineering & Env Department
        8       18733  Buildings, Safety Engineering & Env Department
        9       28204  Buildings, Safety Engineering & Env Department
        12      18743  Buildings, Safety Engineering & Env Department
        13      18741  Buildings, Safety Engineering & Env Department
        14      18978  Buildings, Safety Engineering & Env Department

              inspector_name                        violator_name  \
        0   Sims, Martinzie     INVESTMENT INC., MIDWEST MORTGAGE
        1  Williams, Darrin            Michigan, Covenant House
        5   Sims, Martinzie                       KASIMU, UKWELI
        6  Williams, Darrin  Deerwood Development Group Inc, Deer
        7  Williams, Darrin       Rafee Auto Services L.L.C., RAF
        8  Williams, Darrin       Rafee Auto Services L.L.C., RAF
        9  Williams, Darrin                           Inc, Nanno
        12 Williams, Darrin                   Gardner Resale, GAR
        13 Williams, Darrin                      Hardaway, Kevin
        14 Williams, Darrin            TLC Hand Car Wash, a/k/a

            violation_street_number violation_street_name  violation_zip_code  \
        0                    2900.0                 TYLER                 NaN
        1                    4311.0               CENTRAL                 NaN
        5                    6478.0            NORTHFIELD                 NaN
        6                    8027.0             BRENTWOOD                 NaN
        7                    8228.0            MT ELLIOTT                 NaN
        8                    8228.0            MT ELLIOTT                 NaN
        9                   15307.0            SEVEN MILE                 NaN
        12                   9100.0              VAN DYKE                 NaN
        13                  20024.0              SCHAEFER                 NaN
        14                   9425.0              VAN DYKE                 NaN

            mailing_address_str_number mailing_address_str_name       city  \
        0                          3.0                S. WICKER    CHICAGO
        1                       2959.0       Martin Luther King    Detroit
        5                       2755.0                 E. 17TH  LOG BEACH
        6                        476.0                Garfield    Clinton
        7                       8228.0              Mt. Elliott    Detroit
        8                       8228.0              Mt. Elliott    Detroit
        9                       1537.0           E. Seven Mile    Detroit
```

```
12                      91.0                Van Dyke    Detroit
13                     224.0                Schaefer    Detroit
14                    9425.0                Van Dyke    Detroit

       ...     clean_up_cost  judgment_amount  payment_amount  balance_due  \
0      ...               0.0            305.0             0.0        305.0
1      ...               0.0            855.0           780.0         75.0
5      ...               0.0            305.0             0.0        305.0
6      ...               0.0            855.0             0.0        855.0
7      ...               0.0            140.0             0.0        140.0
8      ...               0.0            140.0             0.0        140.0
9      ...               0.0            855.0             0.0        855.0
12     ...               0.0            855.0             0.0        855.0
13     ...               0.0            855.0             0.0        855.0
14     ...               0.0            855.0             0.0        855.0

           payment_date      payment_status collection_status grafitti_stat
0                   NaN  NO PAYMENT APPLIED               NaN              N
1   2005-06-02 00:00:00        PAID IN FULL               NaN              N
5                   NaN  NO PAYMENT APPLIED               NaN              N
6                   NaN  NO PAYMENT APPLIED               NaN              N
7                   NaN  NO PAYMENT APPLIED               NaN              N
8                   NaN  NO PAYMENT APPLIED               NaN              N
9                   NaN  NO PAYMENT APPLIED               NaN              N
12                  NaN  NO PAYMENT APPLIED     IN COLLECTION              N
13                  NaN  NO PAYMENT APPLIED     IN COLLECTION              N
14                  NaN  NO PAYMENT APPLIED               NaN              N

                         compliance_detail  compliance
0             non-compliant by no payment         0.0
1   compliant by late payment within 1 month         1.0
5             non-compliant by no payment         0.0
6             non-compliant by no payment         0.0
7             non-compliant by no payment         0.0
8             non-compliant by no payment         0.0
9             non-compliant by no payment         0.0
12            non-compliant by no payment         0.0
13            non-compliant by no payment         0.0
14            non-compliant by no payment         0.0

[10 rows x 34 columns]
```

from sklearn.ensemble import RandomForestClassifier from sklearn.model_selection import train_test_split from sklearn.metrics import roc_auc_score from sklearn.metrics import roc_curve, auc from sklearn.preprocessing import LabelEncoder from sklearn.model_selection import GridSearchCV

def blight_model(): return ans

```
In [2]: import pandas as pd
```

```
import numpy as np
import math
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve, auc
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import GridSearchCV

df = pd.read_csv('train.csv', encoding = "ISO-8859-1")
df.index = df['ticket_id']
features_name = ['fine_amount', 'admin_fee', 'state_fee', 'late_fee']

df.compliance = df.compliance.fillna(value=-1)
df = df[df.compliance != -1]

X = df[features_name]
X.fillna(value = -1)

y = df.compliance

X_train, X_test, y_train, y_test = train_test_split(X, y, random_state = 0)

clf = RandomForestClassifier(n_estimators = 10, max_depth = 5).fit(X_train,

features_name = ['fine_amount', 'admin_fee', 'state_fee', 'late_fee']

df_test = pd.read_csv('readonly/test.csv', encoding = "ISO-8859-1")

df_test.index = df_test['ticket_id']

X_predict = clf.predict_proba(df_test[features_name])

ans = pd.Series(data = X_predict[:,1], index = df_test['ticket_id'], dtype=
    #return ans
```

```
/opt/conda/lib/python3.6/site-packages/IPython/core/interactiveshell.py:2717: Dtype
  interactivity=interactivity, compiler=compiler, result=result)
```

In [3]:
```
df.compliance = df.compliance.fillna(value=-1)
df = df[df.compliance != -1]
```

In [5]:
```
clf.score(X_test, y_test)
```

Out[5]: 0.92847135351513632

In [6]:
```
import pandas as pd
import numpy as np
```

```
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

df = pd.read_csv('train.csv', encoding = "ISO-8859-1")
df_test = pd.read_csv('readonly/test.csv', encoding = "ISO-8859-1")
```

```
/opt/conda/lib/python3.6/site-packages/IPython/core/interactiveshell.py:2717: Dtype
  interactivity=interactivity, compiler=compiler, result=result)
```

In [7]:
```
df.head(4)
```

Out[7]:
```
   ticket_id                                     agency_name  \
0      22056  Buildings, Safety Engineering & Env Department
1      27586  Buildings, Safety Engineering & Env Department
2      22062  Buildings, Safety Engineering & Env Department
3      22084  Buildings, Safety Engineering & Env Department

     inspector_name                      violator_name  \
0   Sims, Martinzie  INVESTMENT INC., MIDWEST MORTGAGE
1  Williams, Darrin           Michigan, Covenant House
2   Sims, Martinzie                    SANDERS, DERRON
3   Sims, Martinzie                       MOROSI, MIKE

   violation_street_number violation_street_name  violation_zip_code  \
0                   2900.0                 TYLER                 NaN
1                   4311.0               CENTRAL                 NaN
2                   1449.0            LONGFELLOW                 NaN
3                   1441.0            LONGFELLOW                 NaN

   mailing_address_str_number mailing_address_str_name     city     ...
0                         3.0                S. WICKER  CHICAGO     ...
1                      2959.0       Martin Luther King  Detroit     ...
2                     23658.0                 P.O. BOX  DETROIT     ...
3                         5.0                ST. CLAIR  DETROIT     ...

  clean_up_cost  judgment_amount payment_amount balance_due  \
0           0.0            305.0            0.0       305.0
1           0.0            855.0          780.0        75.0
2           0.0              0.0            0.0         0.0
3           0.0              0.0            0.0         0.0

          payment_date      payment_status collection_status grafitti_statu
0                  NaN  NO PAYMENT APPLIED               NaN               Na
1  2005-06-02 00:00:00        PAID IN FULL               NaN               Na
2                  NaN  NO PAYMENT APPLIED               NaN               Na
3                  NaN  NO PAYMENT APPLIED               NaN               Na
```

```
                      compliance_detail  compliance
0             non-compliant by no payment         0.0
1  compliant by late payment within 1 month         1.0
2           not responsible by disposition         NaN
3           not responsible by disposition         NaN

[4 rows x 34 columns]
```

In [8]:
```
df_test.head(4)
```

Out[8]:
```
   ticket_id                 agency_name      inspector_name  \
0     284932  Department of Public Works  Granberry, Aisha B
1     285362  Department of Public Works       Lusk, Gertrina
2     285361  Department of Public Works       Lusk, Gertrina
3     285338  Department of Public Works    Talbert, Reginald

       violator_name  violation_street_number violation_street_name  \
0   FLUELLEN, JOHN A                  10041.0             ROSEBERRY
1    WHIGHAM, THELMA                  18520.0             EVERGREEN
2    WHIGHAM, THELMA                  18520.0             EVERGREEN
3  HARABEDIEN, POPKIN                  1835.0               CENTRAL

  violation_zip_code mailing_address_str_number mailing_address_str_name  \
0                NaN                        141                ROSEBERRY
1                NaN                      19136              GLASTONBURY
2                NaN                      19136              GLASTONBURY
3                NaN                       2246                   NELSON

        city       ...          \
0    DETROIT       ...
1    DETROIT       ...
2    DETROIT       ...
3  WOODHAVEN       ...

                            violation_description            dispositic
0  Failure to secure City or Private solid waste ...  Responsible by Defaul
1  Allowing bulk solid waste to lie or accumulate...  Responsible by Defaul
2  Improper placement of Courville container betw...  Responsible by Defaul
3  Allowing bulk solid waste to lie or accumulate...  Responsible by Defaul

  fine_amount admin_fee state_fee late_fee discount_amount clean_up_cost
0       200.0      20.0      10.0     20.0             0.0           0.0
1      1000.0      20.0      10.0    100.0             0.0           0.0
2       100.0      20.0      10.0     10.0             0.0           0.0
3       200.0      20.0      10.0     20.0             0.0           0.0

  judgment_amount  grafitti_status
```

```
0          250.0              NaN
1         1130.0              NaN
2          140.0              NaN
3          250.0              NaN

[4 rows x 27 columns]
```

```
In [9]: df.shape
```

```
Out[9]: (250306, 34)
```

```
In [10]: df_test.shape
```

```
Out[10]: (61001, 27)
```

```
In [11]: df_test.describe()
```

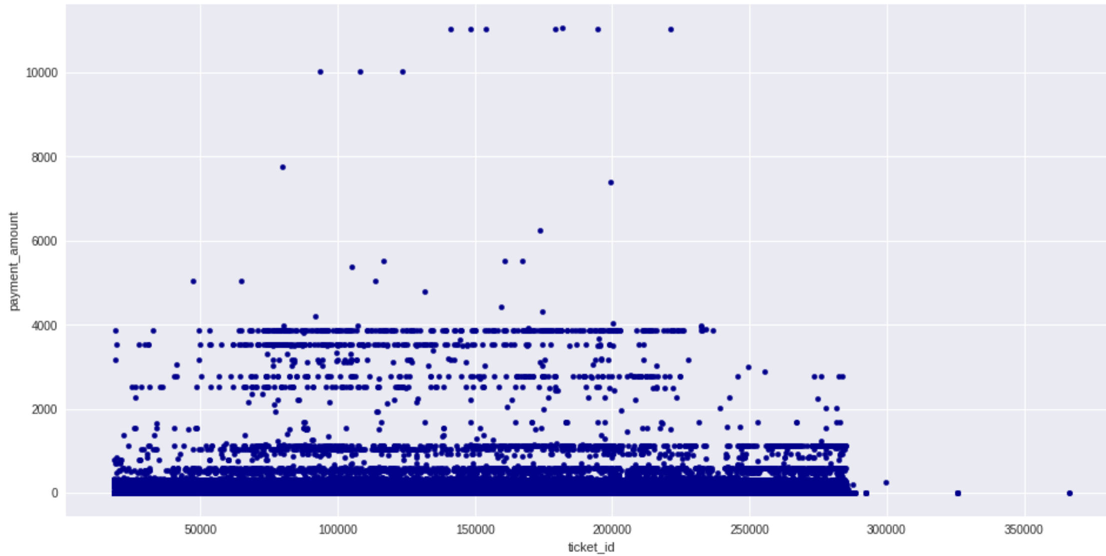
```
Out[11]:            ticket_id  violation_street_number  non_us_str_code   fine_amou
        count   61001.000000             6.100100e+04              0.0  61001.0000
        mean   331724.532811             1.256638e+04              NaN    272.7141
        std     25434.932141             1.414373e+05              NaN    360.1018
        min    284932.000000            -1.512600e+04              NaN      0.0000
        25%    310111.000000             6.008000e+03              NaN     50.0000
        50%    332251.000000             1.213400e+04              NaN    200.0000
        75%    353031.000000             1.716500e+04              NaN    250.0000
        max    376698.000000             2.010611e+07              NaN  10000.0000

               admin_fee  state_fee     late_fee  discount_amount  clean_up_cost
        count    61001.0    61001.0  61001.000000     61001.000000   61001.000000
        mean        20.0       10.0     25.116219         0.239340      20.649711
        std          0.0        0.0     36.310155         3.245894     242.375180
        min         20.0       10.0      0.000000         0.000000       0.000000
        25%         20.0       10.0      5.000000         0.000000       0.000000
        50%         20.0       10.0     10.000000         0.000000       0.000000
        75%         20.0       10.0     25.000000         0.000000       0.000000
        max         20.0       10.0   1000.000000       250.000000   15309.000000

               judgment_amount
        count     61001.000000
        mean        347.895541
        std         460.058043
        min           0.000000
        25%          85.000000
        50%         250.000000
        75%         305.000000
        max       15558.800000
```

```
In [15]: import matplotlib.pyplot as plt
         graph = df.iloc[::,::26]
         graph.plot.scatter(x=0,y=1,c='DarkBlue',figsize=(16,8))
         plt.show
```

Out[15]: <function matplotlib.pyplot.show>



In [20]:
```
import pandas as pd
import numpy as np
import math
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve, auc
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import GridSearchCV

def blight_model():

    # Your code here

    df = pd.read_csv('train.csv', encoding = "ISO-8859-1")

    df.index = df['ticket_id']

    features_name = ['fine_amount', 'admin_fee', 'state_fee', 'late_fee']

    df.compliance = df.compliance.fillna(value=-1)

    df = df[df.compliance != -1]

    X = df[features_name]

    X.fillna(value = -1)
```

```
        y = df.compliance

        X_train, X_test, y_train, y_test = train_test_split(X, y, random_state

        clf = RandomForestClassifier(n_estimators = 10, max_depth = 5).fit(X_t

        features_name = ['fine_amount', 'admin_fee', 'state_fee', 'late_fee']

        df_test = pd.read_csv('readonly/test.csv', encoding = "ISO-8859-1")

        df_test.index = df_test['ticket_id']

        X_predict = clf.predict_proba(df_test[features_name])
        predict = list(X_predict)
        ans = pd.Series(data = X_predict[:,1], index = df_test['ticket_id'], d

        return ans
```

```
In [21]: blight_model()
```

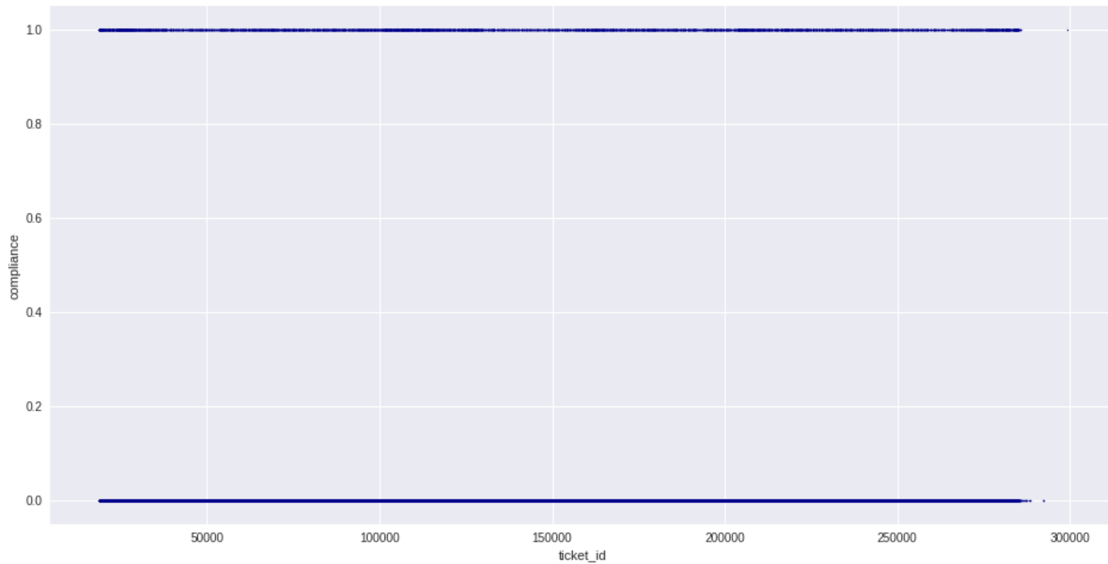
```
/opt/conda/lib/python3.6/site-packages/IPython/core/interactiveshell.py:2827: Dtype
  if self.run_code(code, result):
```

```
Out[21]: ticket_id
         284932    0.060381
         285362    0.026680
         285361    0.069929
         285338    0.060381
         285346    0.069929
         285345    0.060381
         285347    0.057021
         285342    0.402546
         285530    0.026680
         284989    0.026680
         285344    0.057021
         285343    0.026680
         285340    0.026680
         285341    0.057021
         285349    0.069929
         285348    0.060381
         284991    0.026680
         285532    0.026680
         285406    0.026680
         285001    0.026680
         285006    0.026680
         285405    0.026680
```

```
285337    0.026680
285496    0.057021
285497    0.060381
285378    0.026680
285589    0.026680
285585    0.060381
285501    0.069929
285581    0.026680
             ...
376367    0.026680
376366    0.036155
376362    0.036155
376363    0.060381
376365    0.026680
376364    0.036155
376228    0.036155
376265    0.036155
376286    0.367978
376320    0.036155
376314    0.036155
376327    0.367978
376385    0.367978
376435    0.465922
376370    0.367978
376434    0.057021
376459    0.069929
376478    0.005052
376473    0.036155
376484    0.024648
376482    0.026680
376480    0.026680
376479    0.026680
376481    0.026680
376483    0.036155
376496    0.026680
376497    0.026680
376499    0.069929
376500    0.069929
369851    0.303633
dtype: float32
```

In [16]:
```
import matplotlib.pyplot as plt

graph=df.iloc[::,0::33]
graph.plot.scatter(x=0,y=1,c='DarkBlue',figsize=(16,8),s=1)
plt.show()
```


1.0
0.8
0.6
0.4
0.2
0.0
compliance
50000
100000
150000
200000
250000
300000
ticket_id

```
In [ ]:
```