KOREA ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY(KAIST)

Introudction to Data Science(CS361)

Data Science project final report

Melese Medhin

20210727

I. Introduction

This data science project aims to predict the fakeness of import declarations. We are given with a set of features which helps us to figure out whether a given data is declared fake or not. The main work of this project has been to determine those some features that will enable us to predcit, and discarding the remaining ones. We can choose from the many models that helps us to predict. But for this project I specifically chose KNN. The reason I chose KNN is because the model works based on feature similariity. And our data is labled, it is noise free(the labels are either 0 or 1) and the data is relatively small.

As mentioned above the process of feature selections in an important step for the model that is chosen. And so it is crucial to identify those feature which create a distinction between the two labels of fakeness. Histogram, unique value counts and the perecentage of fake individual column values are some of the methods used for future selection.

II. **Feature selection**

My first analysis was on the columns with the missing values. The columns with the missing values are SellerID and ExpressID as shown in the Fig below. I thought it is important to investigate the effect of the exclusion and inclusion of these features. For the Express ID, since the number of missing values closely matches that the number of fake import declaration, I intially thought the presences of the NAN values in this column might one indication of fakeness.Moreover, I tried to explore the effect by changing the NAN values with the mean of the column. In, both cases the accuracy of the model was not good, and in fact, the accuracy got better when I dropped this column. I followed the same reasoning for the SellerID.

```
SellerID      3931
ExpressID    29831
dtype: int64
```

Fig1: shows the count of missing values under SellerID and ExpressID columns

```
after_drop['Fake'].value_counts()
```

```
0    29719
1     8028
Name: Fake, dtype: int64
```

Fig2: shows the count of fake and not-fake values under the column 'Fake'

I also tought knowing the numerical variables, and catogorical values might suggest something. Furthermore, knowing whether the catogorical features are ordinal or nominal helps. TaxRate, TotalGrossMassMeasure(KG), and AdValoremTaxBaseAmount(won) are identified to be numerical values.

Another one is looking for the catogorical columns with relatively small number of unique values. Some of these features have unique values whose count is very close to the number of fake import declarations. From this, it is not very illogical to conjecture that values that exist in the same proportion as the number of fakes might indicate 'fakeness'. If the unique values are distributed equally, the influence can not be told, and hence it is better to abstain from making any conclusion.

```
after_drop.nunique()
```

```
ID                               37747
IssueDateTime                      364
DeclarationOfficeID                 44
ProcessType                          3
TransactionNature                   28
Type                                17
PaymentType                         11
BorderTransportMeans                 7
DeclarerID                        1209
ImporterID                       14908
ClassificationID                  3634
ExportationCountry                 110
OriginCountry                      116
TaxRate                            131
DutyRegime                          50
DisplayIndicator                     6
TotalGrossMassMeasure(KG)         5784
AdValoremTaxBaseAmount(Won)      16635
Fake                                 2
dtype: int64
```

Fig3: count of each column's unique values

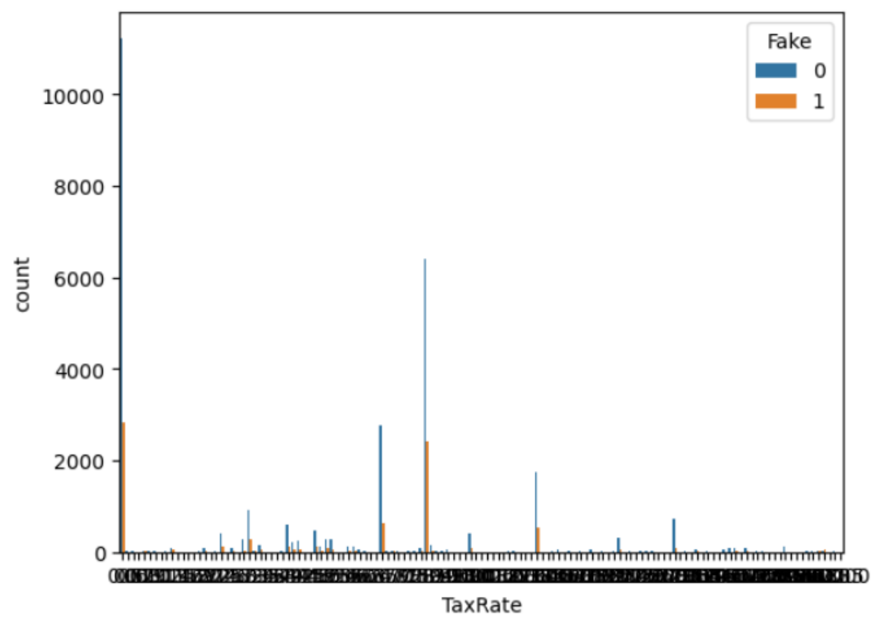
In addition to identifying the number of unique values, it is helpful to know the perecentage those values that are fake not fake. This helps us to make better conclusion. From surface, it might seem that values which occur in the same proprtion as the count of fake imply false import declaration. But, with the help fake perecentage proportion, some of these values have a very low fake import declarations. Figure below shows the above reasoning.

	perc_faketype	value_counts
1	23.809524	21
33	22.857143	245
13	22.340426	564
43	21.810634	6263
18	21.803499	2972
0	21.778584	551
11	21.099883	23948
14	20.905255	2607
21	19.444444	360
24	18.181818	11
12	18.048780	205

	perc_faketype	value_counts
D	29.411765	85
B	21.265870	37492
A	17.647059	170

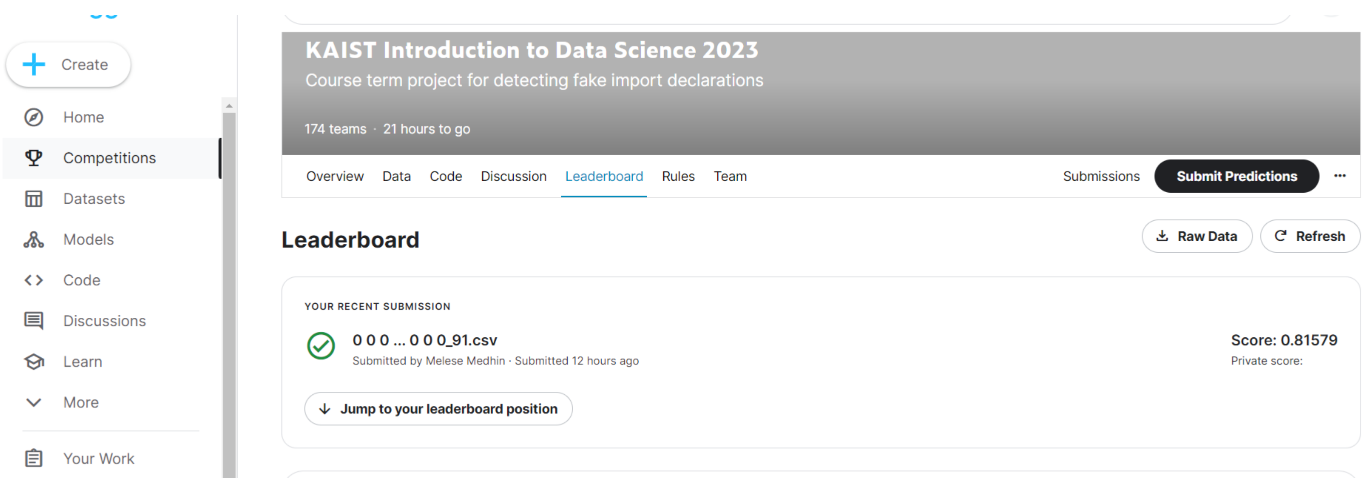
Fig4: shows the % of fake declared columns' values

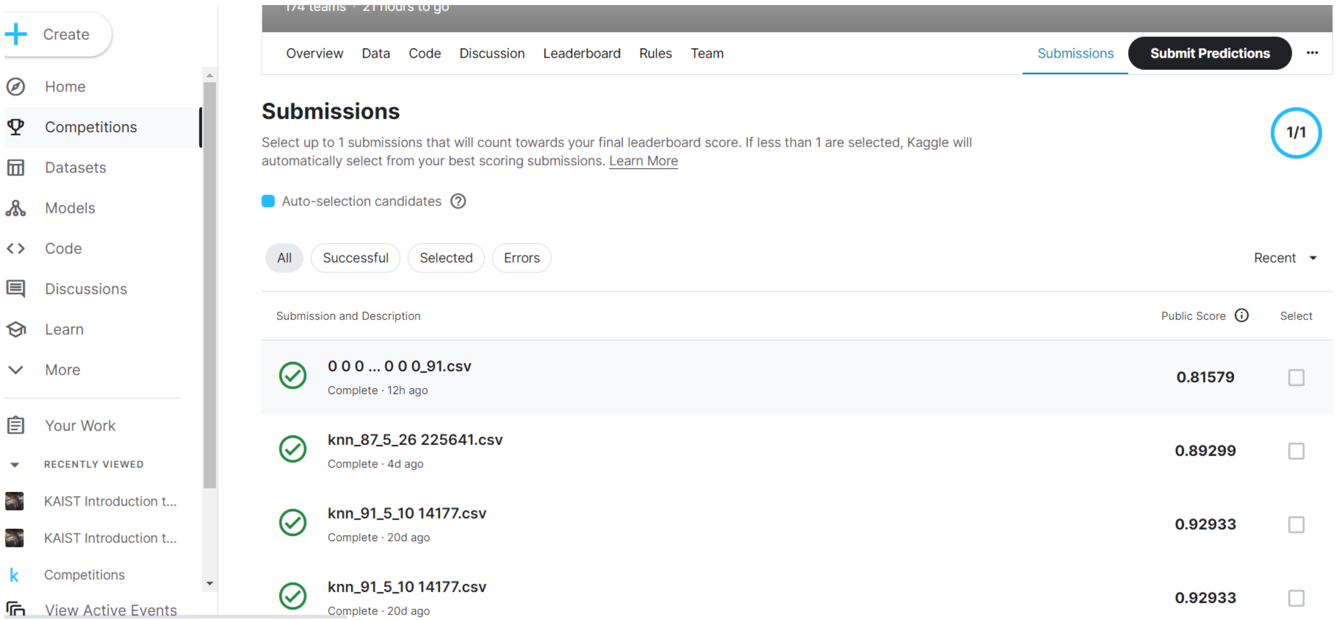
<Axes: xlabel='TaxRate', ylabel='count'>

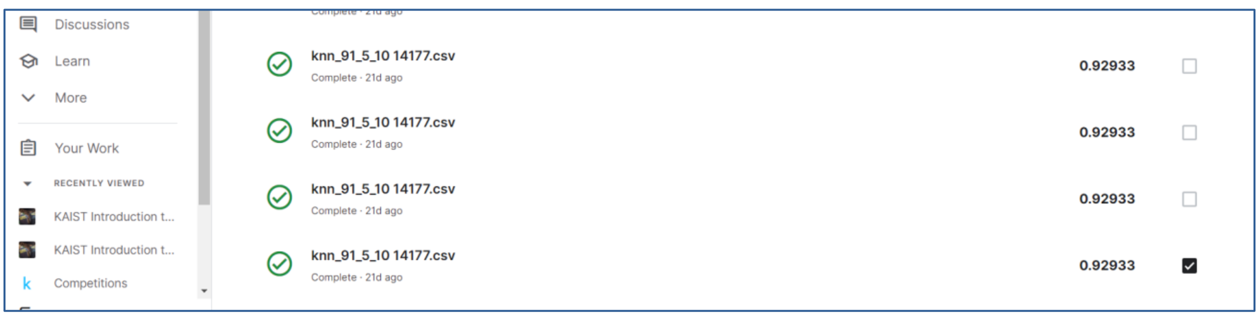


III. **Conclusion**

For the Knn model, the k that gives the highest accuracy is determined to be 301, and the features that were dropped for in the process are SellerID,ExpresssID,ProcessType,DeclareType,ImporterType, TransactionNature and Type. It didn't take much times to decide to drop the first two varaibles, but the rest were more or less omitted through trial.




Create
Home
Competitions
Datasets
Models
Code
Discussions
Learn
More
Your Work
RECENTLY VIEWED
KAIST Introduction t...
KAIST Introduction t...
Competitions
View Active Events
Overview Data Code Discussion Leaderboard Rules Team
Submissions
Submit Predictions
Submissions
Select up to 1 submissions that will count towards your final leaderboard score. If less than 1 are selected, Kaggle will automatically select from your best scoring submissions. Learn More
1/1
Auto-selection candidates
All Successful Selected Errors
Recent
Submission and Description
Public Score
Select
0 0 0 ... 0 0 0_91.csv
Complete · 12h ago
0.81579
knn_87_5_26 225641.csv
Complete · 4d ago
0.89299
knn_91_5_10 14177.csv
Complete · 20d ago
0.92933
knn_91_5_10 14177.csv
Complete · 20d ago
0.92933


Discussions
Learn
More
Your Work
RECENTLY VIEWED
KAIST Introduction t...
KAIST Introduction t...
Competitions
knn_91_5_10 14177.csv
Complete · 21d ago
0.92933
knn_91_5_10 14177.csv
Complete · 21d ago
0.92933
knn_91_5_10 14177.csv
Complete · 21d ago
0.92933
knn_91_5_10 14177.csv
Complete · 21d ago
0.92933