

SEATTLE CITY – COLLISION LOG

CRIS-DM

Erik Draganov Santos

1. Business Understanding:

I have decided to work on the example dataset provided by Seattle City Traffic Department, consisting of collision records registered since 2004. Collisions are classified by 37 features, including its severity degree, which can vary from 1 to 2. *The last degree is associated with injuries.*

Machine Learning algorithms can be extremely helpful in this scenario, which presents **a usual classification problem**. By examining the variables associated with the incident and its respective severity code, the algorithm can indicate the contributing factors to more severe accidents.

The results may be useful to Seattle Authorities as a basis **for developing public policies associated with road safety**, or even identifying traffic areas in the city in **need of maintenance or safety improvements**.

2. Data understanding:

The dataset is comprised of **194.673 records** in total. Besides the target variable, SEVERITYCODE, **there are 37 other columns**, which can be used as a feature to assess the severity of the accident. However, there are columns **with missing or unregular data**. Furthermore, the dataset is imbalanced, having **124.258 accidents associated with the code “01”**, while **55.809 registers associated with the code “02”**.

```
df['SEVERITYCODE'].value_counts()
```

```
1    124258
```

```
2     55809
```

```
Name: SEVERITYCODE, dtype: int64
```

Regarding the features, there are columns specifying the conditions in which the accident occurred, as *weather, light and road conditions*. At first the model will be trained by using only external conditions variables as input, as intuitively they seem to contribute more to errors. Depending on the results, more features such as *collision type and location* will be included.

3. Data Preparation:

I will be conducting the following procedures in the data preparation stage:

- a) **Balancing the dataset:** the first task to be completed in order to avoid biased models. An undersampling approach will be applied, reducing the

numbers of records in the majority class (01) instead of expanding the numbers of records in the minority class (02). Since the difference between each class is considerable, it is intuitively better to shuffle and reduce the data instead of expanding records based on faulty assumptions.

b) Data cleaning: the following task involves deleting columns without substantial data. If a column does not have at least 70% of useful data, it should be deleted. Rows with no information or data are to be deleted as well.

c) Data transformation: some columns require data transformation, since records are mixed, having for instance 1, 0, Y, N as records. A specific pattern should be applied, preferably 1,0.

Example:

```
df['UNDERINFL'] = df['UNDERINFL'].replace(['Y'], '1')
df['UNDERINFL'] = df['UNDERINFL'].replace(['N'], '0')
```

As indicated by the following summary, there are columns in the dataset with substantial lack of information, or even no information at all:

Missing Information Analysis (False = identifiable data | True = no data).

```
SEVERITYCODE
False      194673
Name: SEVERITYCODE, dtype: int64
```

```
X
False      189339
True       5334
Name: X, dtype: int64
```

```
Y
False      189339
True       5334
Name: Y, dtype: int64
```

```
OBJECTID
False      194673
Name: OBJECTID, dtype: int64
```

```
INCKEY
False      194673
Name: INCKEY, dtype: int64
```

```
COLDETKEY
False      194673
Name: COLDETKEY, dtype: int64
```

```
REPORTNO
False      194673
Name: REPORTNO, dtype: int64
```

STATUS
False 194673
Name: STATUS, dtype: int64

ADDRTYPE
False 192747
True 1926
Name: ADDRTYPE, dtype: int64

INTKEY
True 129603
False 65070
Name: INTKEY, dtype: int64

LOCATION
False 191996
True 2677
Name: LOCATION, dtype: int64

EXCEPTRSNCODE
True 109862
False 84811
Name: EXCEPTRSNCODE, dtype: int64

EXCEPTRSNDESC
True 189035
False 5638
Name: EXCEPTRSNDESC, dtype: int64

SEVERITYCODE.1
False 194673
Name: SEVERITYCODE.1, dtype: int64

SEVERITYDESC
False 194673
Name: SEVERITYDESC, dtype: int64

COLLISIONTYPE
False 189769
True 4904
Name: COLLISIONTYPE, dtype: int64

PERSONCOUNT
False 194673
Name: PERSONCOUNT, dtype: int64

PEDCOUNT
False 194673
Name: PEDCOUNT, dtype: int64

PEDCYLCOUNT
False 194673
Name: PEDCYLCOUNT, dtype: int64

VEHCOUNT
False 194673

Name: VEHCOUNT, dtype: int64

INCDATE

False 194673

Name: INCDATE, dtype: int64

INCDTTM

False 194673

Name: INCDTTM, dtype: int64

JUNCTIONTYPE

False 188344

True 6329

Name: JUNCTIONTYPE, dtype: int64

SDOT_COLCODE

False 194673

Name: SDOT_COLCODE, dtype: int64

SDOT_COLDESC

False 194673

Name: SDOT_COLDESC, dtype: int64

INATTENTIONIND

True 164868

False 29805

Name: INATTENTIONIND, dtype: int64

UNDERINFL

False 189789

True 4884

Name: UNDERINFL, dtype: int64

WEATHER

False 189592

True 5081

Name: WEATHER, dtype: int64

ROADCOND

False 189661

True 5012

Name: ROADCOND, dtype: int64

LIGHTCOND

False 189503

True 5170

Name: LIGHTCOND, dtype: int64

PEDROWNOTGRNT

True 190006

False 4667

Name: PEDROWNOTGRNT, dtype: int64

SDOTCOLNUM

False 114936

True 79737

Name: SDOTCOLNUM, dtype: int64

```
SPEEDING
True      185340
False     9333
Name: SPEEDING, dtype: int64
```

```
ST_COLCODE
False     194655
True      18
Name: ST_COLCODE, dtype: int64
```

```
ST_COLDESC
False     189769
True      4904
Name: ST_COLDESC, dtype: int64
```

```
SEGLANEKEY
False     194673
Name: SEGLANEKEY, dtype: int64
```

```
CROSSWALKKEY
False     194673
Name: CROSSWALKKEY, dtype: int64
```

```
HITPARKEDCAR
False     194673
Name: HITPARKEDCAR, dtype: int64
```

4. Modeling:

As mentioned, the problem under analysis herein is centered in classification. As such, the following algorithms and methods will be trained, assessed, and compared:

- a) **KNeighborsClassifier:** initially the best “k” value should be identified, so the model can be trained accordingly.
- b) **LogisticRegression:** basic algorithm.
- c) **DecisionTreeClassifier:** preferably with a higher number of levels, due to the quantity of features (complexity).
- d) **SVC – Support Vector Machines:** if possible, since the algorithm executes complex and hardware demanding calculations.

5. Evaluation:

A comparison table will be presented, comparing the four models in regard to their “Jaccard Index” and “F1-Scores”.

6. Deployment:

A final report will be developed, with the associated code published on GitHub.