Model3

February 27, 2021

[1]:
```
import turicreate as tc
import os
```

[2]:
```
data=tc.SFrame.read_csv("crash1.csv",column_type_hints=[str]*76+[int])
data2=tc.SFrame.read_csv("nearcrash1.csv",column_type_hints=[str]*76+[int])
data=data.append(data2)
```

```
Finished parsing file /media/bhavya_bhardwaj/Bhavya/Hackathon/Sweden-India Hackathon/VirginiaDa

Parsing completed. Parsed 29030 lines in 0.313519 secs.

Read 216148 lines. Lines per second: 52154.7

Finished parsing file /media/bhavya_bhardwaj/Bhavya/Hackathon/Sweden-India Hackathon/VirginiaDa

Parsing completed. Parsed 343033 lines in 5.53497 secs.
```

[3]:
```
data.export_csv("CompleteData.csv")
```

[4]:
```
data=data.shuffle()
data_train,data_test=tc.SFrame.random_split(data,0.7)
data_test1, data_validation=tc.SFrame.random_split(data_test,0.5)
```

[5]:
```
model=tc.classifier.create(data,target='Targets',validation_set=data_test1)
```

```
PROGRESS: The following methods are available for this type of problem.
PROGRESS: LogisticClassifier, SVMClassifier
PROGRESS: The returned model will be chosen according to validation accuracy.

Logistic regression:

--------------------------------------------------------

Number of examples          : 372063
```

```
Number of classes           : 2

Number of feature columns   : 76

Number of unpacked features : 76

Number of coefficients      : 410708

Starting L-BFGS

--------------------------------------------------------

+-----------+----------+-----------+--------------+-------------------+---------------------+
| Iteration | Passes   | Step size | Elapsed Time | Training Accuracy | Validation Accuracy |
+-----------+----------+-----------+--------------+-------------------+---------------------+
| 0         | 3        | 0.500000  | 1.832373     | 0.921976          | 0.922304            |
| 1         | 9        | 4.500000  | 3.622921     | 0.970094          | 0.969682            |
| 2         | 10       | 4.500000  | 4.089906     | 0.497246          | 0.499184            |
| 3         | 16       | 1.692008  | 5.858177     | 0.986701          | 0.986599            |
| 4         | 17       | 1.692008  | 6.324767     | 0.995947          | 0.996179            |
| 9         | 27       | 1.000000  | 10.114798    | 0.999312          | 0.999282            |
+-----------+----------+-----------+--------------+-------------------+---------------------+

SVM:

--------------------------------------------------------

Number of examples          : 372063
```

```
Number of classes           : 2

Number of feature columns   : 76

Number of unpacked features : 76

Number of coefficients    : 410708

Starting L-BFGS

--------------------------------------------------------

+-----------+----------+-----------+--------------+-------------------+---------------------+

| Iteration | Passes   | Step size | Elapsed Time | Training Accuracy | Validation Accuracy |

+-----------+----------+-----------+--------------+-------------------+---------------------+

| 0         | 3        | 0.500000  | 0.806774     | 0.921976          | 0.922304            |

| 1         | 7        | 10.500000 | 1.992905     | 0.975246          | 0.975513            |

| 2         | 9        | 8.729078  | 2.719603     | 0.097825          | 0.097447            |

| 3         | 14       | 0.259867  | 4.210410     | 0.993399          | 0.993291            |

| 4         | 15       | 0.324833  | 4.648111     | 0.993523          | 0.993398            |

| 9         | 20       | 0.991313  | 6.956193     | 0.998516          | 0.998619            |

+-----------+----------+-----------+--------------+-------------------+---------------------+

PROGRESS: Model selection based on validation accuracy:
PROGRESS: ---------------------------------------------
PROGRESS: LogisticClassifier              : 0.9992824211111709
PROGRESS: SVMClassifier                   : 0.998618660639004
PROGRESS: ---------------------------------------------
PROGRESS: Selecting LogisticClassifier based on validation set performance.
```

[8]:
```
result=model.predict(data_validation)
right=0
wrong=0
for i in range(len(result)):
    if result[i]==data_validation['Targets'][i]:
        right+=1
    else:
        wrong+=1
print("Right=",right)
print("Wrong=",wrong)
```

```
Right= 55339
Wrong= 30
```

[7]:
```
model.save("Model3")
```

[ ]:
```

```