San Francisco Crime Classification

Who? Raphael Emberger

From? NUT

When? January 10, 2019

Challenge

Classify type of crime using:

- Date-time
- Weekday
- District name
- Address
- Latitude
- Longitude

into 39 labels.
Dataset from 2003 - 2015

Map of Crimes

Ranking

- Soft-max
- Log-loss:

$$loss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log\left(p_{ij}\right) \tag{1}$$

N : Number of cases in dataset.

M : Number of classes.

y_{ij} : Label for class. 1 if i is in j. Otherwise 0.

p_{ij} : Predicted probability that i belongs to j.

- 2335 submissions
- 34.53877 - 1.95936
- Rank 2241: 32.89183

Dataset

- **sampleSubmission.csv**(884k x 40)
- **test.csv**(884k x 7)
- **train.csv**(878k x 9)

Train Dataset

Dates	Category	Descript	DayOfWeek	PdDistrict
2015-05-13 23:53:00	WARRANTS	WARRANT ARREST	Wednesday	NORTHERN
2015-05-13 23:53:00	OTHER OFFENSES	TRAFFIC VIOLATION ARREST	Wednesday	NORTHERN
2015-05-13 23:33:00	OTHER OFFENSES	TRAFFIC VIOLATION ARREST	Wednesday	NORTHERN
2015-05-13 23:30:00	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Wednesday	NORTHERN
2015-05-13 23:30:00	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Wednesday	PARK
2015-05-13 23:30:00	LARCENY/THEFT	GRAND THEFT FROM UNLOCKED AUTO	Wednesday	INGLESIDE
2015-05-13 23:30:00	VEHICLE THEFT	STOLEN AUTOMOBILE	Wednesday	INGLESIDE
2015-05-13 23:30:00	VEHICLE THEFT	STOLEN AUTOMOBILE	Wednesday	BAYVIEW
2015-05-13 23:00:00	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Wednesday	RICHMOND
2015-05-13 23:00:00	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Wednesday	CENTRAL

Resolution	Address	X	Y
”ARREST, BOOKED”	OAK ST / LAGUNA ST	-122.425891675136	37.7745985956747
”ARREST, BOOKED”	OAK ST / LAGUNA ST	-122.425891675136	37.7745985956747
”ARREST, BOOKED”	VANNESS AV / GREENWICH ST	-122.42436302145	37.8004143219856
NONE	1500 Block of LOMBARD ST	-122.42699532676599	37.80087263276921
NONE	100 Block of BRODERICK ST	-122.438737622757	37.771541172057795
NONE	0 Block of TEDDY AV	-122.40325236121201	37.713430704116
NONE	AVALON AV / PERU AV	-122.423326976668	37.7251380403778
NONE	KIRKWOOD AV / DONAHUE ST	-122.371274317441	37.7275640719518
NONE	600 Block of 47TH AV	-122.508194031117	37.776601260681204
NONE	JEFFERSON ST / LEAVENWORTH ST	-122.419087676747	37.8078015516515

Table: train.csv(first 10 rows)

Test Dataset

Id	Dates	DayOfWeek	PdDistrict	Address	X	Y
0	2015-05-10 23:59:00	Sunday	BAYVIEW	2000 Block of THOMAS AV	-122.39958770418998	37.7350510103906
1	2015-05-10 23:51:00	Sunday	BAYVIEW	3RD ST / REVERE AV	-122.391522893042	37.7324323864471
2	2015-05-10 23:50:00	Sunday	NORTHERN	2000 Block of GOUGH ST	-122.426001954961	37.7922124386284
3	2015-05-10 23:45:00	Sunday	INGLESIDE	4700 Block of MISSION ST	-122.437393972517	37.7214120621391
4	2015-05-10 23:45:00	Sunday	INGLESIDE	4700 Block of MISSION ST	-122.437393972517	37.7214120621391
5	2015-05-10 23:40:00	Sunday	TARAVAL	BROAD ST / CAPITOL AV	-122.45902362242902	37.7131719025215
6	2015-05-10 23:30:00	Sunday	INGLESIDE	100 Block of CHENERY ST	-122.42561645123001	37.73935051446279
7	2015-05-10 23:30:00	Sunday	INGLESIDE	200 Block of BANKS ST	-122.41265203979201	37.739750156312105
8	2015-05-10 23:10:00	Sunday	MISSION	2900 Block of 16TH ST	-122.4187000097043	37.7651649409646
9	2015-05-10 23:10:00	Sunday	CENTRAL	TAYLOR ST / GREEN ST	-122.413934584561	37.798886450641604

Table: test.csv(first 10 rows)

Labels

ARSON	ASSAULT	BAD CHECKS
BRIBERY	BURGLARY	DISORDERLY CONDUCT
DRIVING UNDER THE INFLUENCE	DRUG/NARCOTIC	DRUNKENNESS
EMBEZZLEMENT	EXTORTION	FAMILY OFFENSES
FORGERY/COUNTERFEITING	FRAUD	GAMBLING
KIDNAPPING	LARCENY/THEFT	LIQUOR LAWS
LOITERING	MISSING PERSON	NON-CRIMINAL
OTHER OFFENSES	PORNOGRAPHY/OBSCENE MAT	PROSTITUTION
RECOVERED VEHICLE	ROBBERY	RUNAWAY
SECONDARY CODES	SEX OFFENSES FORCIBLE	SEX OFFENSES NON FORCIBLE
STOLEN PROPERTY	SUICIDE	SUSPICIOUS OCC
TREA	TRESPASS	VANDALISM
VEHICLE THEFT	WARRANTS	WEAPON LAWS

Table: Crime classes

First Approach

- Keras model
- Preprocessing:
  - Conversion to integer/float
  - Feature scaling
  - Mean normalization
- OOP
- Caching

First Approach

```
28        model = keras.Sequential([
29            keras.layers.Dense(16,
                  ↪ input_shape=(train_data.shape[1],),
                  ↪ activation='relu'),
30            keras.layers.Dense(64, activation='relu'),
31            keras.layers.Dense(39, activation='softmax')
32        ])
33        self.log.info("Constructed model")
34        optimizer = keras.optimizers.Adam(lr=0.04)
35        model.compile(optimizer=optimizer,
36                      loss='sparse_categorical_crossentropy',
37                      metrics=['accuracy'])
38        self.log.info("Compiled model")
39
40        model.fit(train_data, train_labels, epochs=5,
              ↪ batch_size=200)
41        self.log.info("Trained model")
```

Listing 1: Keras model - model.py

First Approach(last one, I swear)

```
34  trainsamplesfile = TrainDataCsvFile()
35  trainlabelfile = TrainLabelsCsvFile(trainsamplesfile)
36  testsamplesfile = TestDataCsvFile()
37  for file in [trainsamplesfile, trainlabelfile, testsamplesfile]:
38     if args.prep_data or not file.prep_file_exists():
39        file.parse()
40        file.save()
41     else:
42        file.load()
43
44  if args.train:
45     mdl = Model().get_model(
46        trainsamplesfile.toNpArray(),
47        trainlabelfile.toNpArray()
48     )
49  else:
50     mdl = Model().get_model()
51
52  predictions = mdl.predict(trainsamplesfile.toNpArray())
53  print("LogLoss: {}".format(log_loss(trainlabelfile.toNpArray(), predictions)))
54  predicted_crime = np.argmax(predictions, axis=1)
55  print("Accuracy: {}%".format(accuracy_score(trainlabelfile.toNpArray(),
        ↪ predicted_crime) * 100))
```

Listing 2: Classification process(first approach) - main.py

Results

- Rank 1058
- Accuracy converges to ca. 19.9%
- Mostly assumes ”Larceny/Theft”(19.92% of the Dataset)
- Tempering with parameters has no effect
- Different approach required

Second Approach

- Ramunno-Johnson's blog
- Bernoulli Naïves Bayes(Linear Regression for comparison)
- Preprocessing: Binarization
  - Minutes(m0 - m59)
  - Hours(H1 - H12)
  - Days(D1 - D31)
  - Months(M1 - M12)
  - Years(Y2003 - Y2015)
  - Weekdays(Monday - Sunday)
  - Districts(Bayview - Tenderloin)

Second Approach

```
30  def preprocess_dataframe(data: pd.DataFrame) -> Tuple[pd.DataFrame, list]:
31      print("Binarize data")
32      minute = pd.get_dummies(data.Dates.dt.minute).rename(columns=MINUTECOLUMNS)
33      hour = pd.get_dummies(data.Dates.dt.hour).rename(columns=HOURCOLUMNS)
34      day = pd.get_dummies(data.Dates.dt.day).rename(columns=DAYCOLUMNS)
35      month = pd.get_dummies(data.Dates.dt.month).rename(columns=MONTHCOLUMNS)
36      year = pd.get_dummies(data.Dates.dt.year).rename(columns=YEARCOLUMNS)
37      weekdays = pd.get_dummies(data.DayOfWeek)
38      districts = pd.get_dummies(data.PdDistrict)
39      x = data.X
40      y = data.Y
41      print("Assemble new array")
42      new_data = pd.concat([minute, hour, day, month, year, weekdays, districts, x,
            ↪ y], axis=1)
43      columns = new_data.keys().tolist()
44      return new_data, columns
45
46
47  def evaluate(prediction, labels):
48      print("LogLoss: {}".format(log_loss(labels, prediction)))
49      predicted_crime = np.argmax(prediction, axis=1)
50      print("Accuracy: {}%".format(accuracy_score(labels, predicted_crime) * 100))
```

Listing 3: Pre-processing method - newmain.py

$\Rightarrow$ simpler script

Second Approach(Bernoulli Naïves Bayes)

```
53  print("Load Data with pandas, and parse the first column into datetime")
54  train = pd.read_csv('train.csv', parse_dates=['Dates'])
55  test = pd.read_csv('test.csv', parse_dates=['Dates'])
56
57  print("Convert crime labels to numbers")
58  le_crime = preprocessing.LabelEncoder()
59  crime = le_crime.fit_transform(train.Category)
60
61  print("Build training data")
62  train_data, features = preprocess_dataframe(train)
63  train_data['crime'] = crime
64
65  print("Features[{}]: {}".format(len(features), np.array(features)))
66
67  print("Split up training data")
68  # training, validation = train_test_split(train_data, test_size=.20)
69  training = train_data
70  validation = train_data
71
72  # Bernoulli Naïve Bayes
73  print("Train Bernoulli Naïve Bayes classifier")
74  air_bnb = BernoulliNB()
75  air_bnb.fit(training[features], training['crime'])
76
77  print("Predict labels")
78  predicted = air_bnb.predict_proba(validation[features])
79
80  print("Validate prediction")
81  evaluate(predicted, validation['crime'])
```

Listing 4: Bernoulli Naïve Bayes fitting- newmain.py

Results

- Rank 675(before: 1058)
- Log-loss: 2.464(26.02%)
- Logistic Regression on rank 1149
- Log-loss: 2.591(24.43%)

Third Approach

- Keras model
- Combine
- Preprocessing: Same binarization

Third Approach

```
100 optimizer = keras.optimizers.Adam(lr=0.01)
101 model.compile(optimizer=optimizer,
102               loss='sparse_categorical_crossentropy',
103               metrics=['sparse_categorical_accuracy'])
104 model.fit(training[features], training['crime'],
        ↪ epochs=5, batch_size=1024)
105
106 print("Predict labels")
107 predicted = model.predict_proba(validation[features])
108
109 print("Validate prediction")
110 evaluate(predicted, validation['crime'])
111
112 # Logistic Regression
113 print("Train Logistic Regression for comparison")
114 lr = LogisticRegression(C=0.1, solver='lbfgs',
        ↪ multi_class='multinomial')
115 lr.fit(training[features], training['crime'])
```

Listing 5: Keras model integrated - newmain.py

Results

- Rank 664(before: 675)
- Log-loss: 2.456(26.39%) - before: 2.464(26.02%)

Conclusion and Future Considerations

Mistakes made:

- Too much time invested into first approach
- Too little knowledge and experience with NNs

Potential next steps:

- Gathering knowledge about NNs
- Enrich dataset
- Even out the dataset

End of Presentation

Thank you for your attention