

Medicare fraud detection by using open source data



```
class author(object):  
  
    name = "Xinyu (Max) Liu"  
  
    email = "xinyulrsm@gmail.com"  
  
    create = "01/16/2016"  
  
    address = "Waltham, MA 02453"
```

Outline

- Fraud detection is important in healthcare.
- The data source
- Feature engineering
- Models
- Results

Year	People Charged	False Billings, in Million
2010	94	\$251
2011	91	\$295
2012	2	\$1.9
2013	89	\$223
2014		
2015	243	\$712

Data source: [en.Wikipedia.org/wiki/Medicare_fraud](https://en.wikipedia.org/wiki/Medicare_fraud)

The cluster (5 nodes Hadoop/yarn/spark)

The image shows a terminal window with the output of the 'top' command for a multi-node cluster. The output is organized into columns for each node, showing CPU usage, memory usage, and running processes. Overlaid on the terminal are four white boxes with black text identifying the nodes: 'Node-1 2 core 4 G mem', 'Node-2 2 core 4 G mem', 'Node-3 1 core 4 G mem', and 'Node-4 2 core 8 G mem'. The terminal text includes headers for each node's top output, such as 'd19D USER PRI NI VIRT RES SHR S CPU% MEM% TIME Command' for Node-1, and similar headers for the other nodes. The processes listed include various Java applications, system utilities like 'sshd', and monitoring tools like 'iostat'.

Another way to monitor the cluster

```

Every 1.0s: ./w.sh          Thu Feb 25 14:51:45 2016          Thu Feb 25 14:51:42 2016
pocoyo-1
21884 Jps
7083 ResourceManager
7235 NodeManager
7701 NameNode
7922 DataNode
8300 JobHistoryServer
9145 Master
9360 Worker
9526 SparkSubmit

Every 1.0s: ./w.sh          Thu Feb 25 14:50:43 2016          Thu Feb 25 14:51:44 2016
pocoyo-2
28009 Jps
24214 NodeManager
24530 DataNode
24743 SecondaryNameNode
25834 Worker

Every 1.0s: ./w.sh          Thu Feb 25 14:51:16 2016
pocoyo-3
13122 Jps
18951 NodeManager
19247 DataNode
20258 Worker

Spencer
12030 Jps
3217 NodeManager
3550 DataNode
4797 Worker

harold
1724 Nailgun
2478 Jps
2654 NodeManager
3135 DataNode
32096
4173 SparkSubmit
4512 Worker
5715
  
```

Copy files to Hadoop then move to Hive warehouse

Copy data to hadoop

```

%%bash
$HADOOP_HOME/bin/hdfs dfs -mkdir -p /data/
$HADOOP_HOME/bin/hdfs dfs -put /home/max/data/PARTD_PRESCRIBER_PUF_NPI_DRUG_13.tab /data
$HADOOP_HOME/bin/hdfs dfs -put /home/max/data/OP_DTL_GNRL_PGYR2013_P01152016.csv /data
  
```

Hadoop Overview Datanodes Snapshot Startup Progress Utilities -

Browse Directory

/user/hive/warehouse/payment_13

Go!

Permission	Owner	Group	Size	Replication	Block Size	Name
-rwxrwxr-x	hduser	supergroup	1.98 GB	3	128 MB	OP_DTL_GNRL_PGYR2013_P01152016.csv
Permission	Owner	Group	Size	Replication	Block Size	Name
-rwxrwxr-x	hduser	supergroup	2.65 GB	3	128 MB	PARTD_PRESCRIBER_PUF_NPI_DRUG_13.tab

Pig scripts for data cleaning and table join

Writefile dataPrep.pig

Register '/usr/local/pig/lib/piggybank.jar';

```
partd_raw = load '/data/HealthCare/PARTD_PRESCRIBER_PUF_NPI_DRUG_13.tab'
using org.apache.pig.piggybank.storage.CSVExcelStorage('\t', 'YES_MULTILINE', 'NOCHANGE', 'SKIP_INPUT_HEADER')
as {
```

```
  NPI:-->int,
  NPPES_PROVIDER_LAST_ORG_NAME: chararray,
  NPPES_PROVIDER_FIRST_NAME:-->chararray,
  NPPES_PROVIDER_CITY:-->chararray,
  ...
  TOTAL_DRUG_COST_GE65:-->float
  );
```

npi_drugs = group partd_raw by NPI;

```
npi_drugs table = foreach npi_drugs {
  specilty = limit partd_raw.SPECIALTY_DESC 1;
  lastname = limit partd_raw.NPPES_PROVIDER_LAST_ORG_NAME 1;
  firstname = limit partd_raw.NPPES_PROVIDER_FIRST_NAME 1;
  city = limit partd_raw.NPPES_PROVIDER_CITY 1;
  state = limit partd_raw.NPPES_PROVIDER_STATE 1;
```

```
  generate
  group as gp,
```

```
  ...
  COUNT(partd_raw),
  SUM(partd_raw.TOTAL_DRUG_COST);
};
```

df_partd.head()

	npi	count	specialty	claim_min	claim_max	claim_sum	supply_min	supply_max	supply_sum	drug_min	drug_max	drug_sum
0	1528364486	2	Dentist	14	19	33	176	305	481	97.54	162.96	260.500008
1	1437177490	5	Dentist	14	89	188	46	2955	4726	106.12	4346.47	5468.750191
2	1154586170	2	Dentist	12	17	29	141	170	311	37.09	119.95	157.039997
3	1215042155	9	General Surgery	17	87	522	98	410	2107	84.80	377.74	2120.089966
4	1104847011	1	Dentist	11	11	11	78	78	78	59.26	59.26	59.259998

```
drugs = group partd_raw by (NPI, GENERIC_NAME);
```

```
npi_drug_table 2 = foreach drugs {
  specilty = limit partd_raw.SPECIALTY_DESC 1;
  lastname = limit partd_raw.NPPES_PROVIDER_LAST_ORG_NAME 1;
  firstname = limit partd_raw.NPPES_PROVIDER_FIRST_NAME 1;
  city = limit partd_raw.NPPES_PROVIDER_CITY 1;
  state = limit partd_raw.NPPES_PROVIDER_STATE 1;
  generate
  flatten(group),
  COUNT(partd_raw),
  SUM(partd_raw.TOTAL_CLAIM_COUNT),
  SUM(partd_raw.TOTAL_DAY_SUPPLY),
  SUM(partd_raw.TOTAL_DRUG_COST);
};
rmf /data/to_hadoop/partd_13_npi_drug_all.csv
store npi_drug_table_2 into '/data/to_hadoop/partd_13_npi_drug_all.csv' using PigStorage('\t');
```

npi_drug.head()

	npi	drug	count	total_claim_count	total_day_supply	total_drug_cost
0	1003000126	LISINAPRIL	1	1.301030	2.756636	2.005909
1	1003000126	SIMVASTATIN	1	1.255273	2.688420	2.013090
2	1003000126	WARFARIN SODIUM	1	1.079181	2.511883	2.221284
3	1003000142	BACLOFEN	1	1.204120	2.654177	2.148633
4	1003000142	MELOXICAM	1	1.505150	2.924796	2.275749

```

payment_raw = load '/data/HealthCare/OP_DTL_GNRL_PGYR2013_P01152016.csv'
using org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'YES_MULTILINE', 'NOCHANGE', 'SKIP_INPUT_HEADER')
as (
  Covered_Recipient_Type:chararray,
  Teaching_Hospital_ID:chararray,
  ...
  Name_of_Associated_Covered_Device_or_Medical_Supply4:chararray,
  Name_of_Associated_Covered_Device_or_Medical_Supply5:chararray,
  Program_Year:chararray,
  Payment_Publication_Date:chararray
);

npi_payment = group payment_raw by (Physician_First_Name, Physician_Last_Name,
                                   Recipient_City, Recipient_State);
npi_payment_table = foreach npi_payment {
  generate
  /*flatten(group),*/
  UPPER(group.Physician_First_Name) as first_name,
  UPPER(group.Physician_Last_Name) as last_name,
  UPPER(group.Recipient_City) as city,
  UPPER(group.Recipient_State) as state,
  COUNT(payment_raw),
  SUM(payment_raw.Total_Amount_of_Payment_USDollars)
};

```

```
df.head()
```

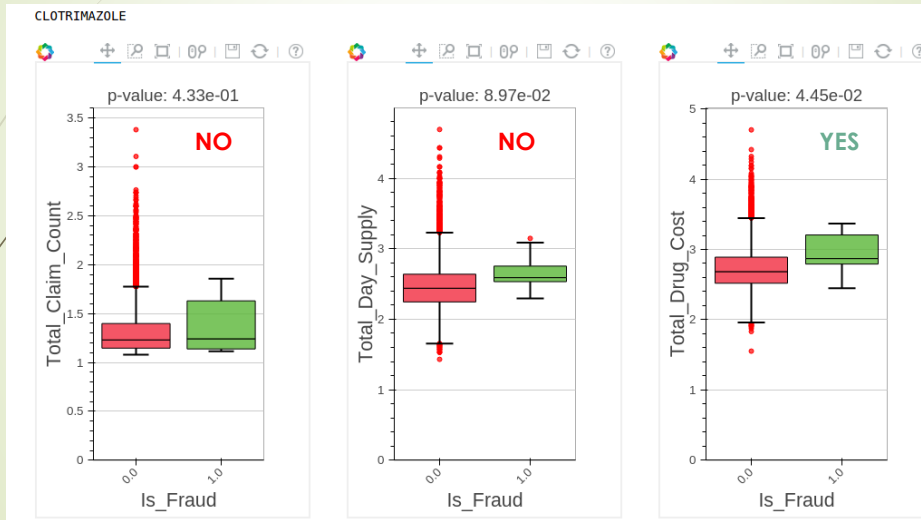
ty	claim_min	claim_max	claim_sum	supply_min	supply_max	supply_sum	drug_min	drug_max	drug_sum	payment_count	total_payment
14	19	33	176	305	481	97.54	162.96	260.500008	0	0	
14	89	188	46	2955	4726	106.12	4346.47	5468.750191	0	0	
12	17	29	141	170	311	37.09	119.95	157.039997	0	0	
17	87	522	98	410	2107	84.80	377.74	2120.089966	0	0	
11	11	11	78	78	78	59.26	59.26	59.259998	0	0	

Npi exclusions database

```
df.head()
```

min	claim_max	claim_sum	supply_min	supply_max	supply_sum	drug_min	drug_max	drug_sum	payment_count	total_payment	is_fraud
19	33	176	305	481	97.54	162.96	260.500008	0	0	0	
89	188	46	2955	4726	106.12	4346.47	5468.750191	0	0	0	
17	29	141	170	311	37.09	119.95	157.039997	0	0	0	
87	522	98	410	2107	84.80	377.74	2120.089966	0	0	0	
11	11	78	78	78	59.26	59.26	59.259998	0	0	0	

An example
Select the feature with $P \leq 0.05$

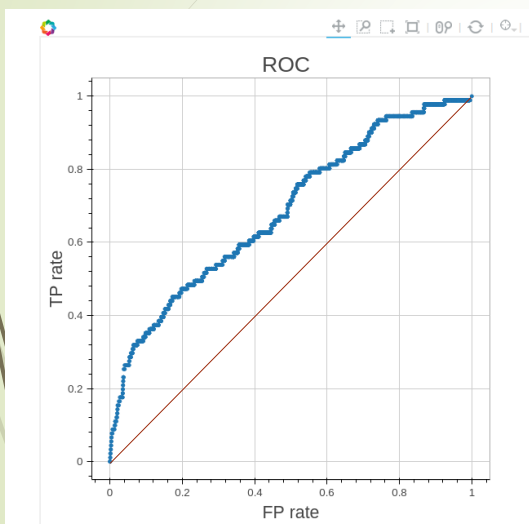


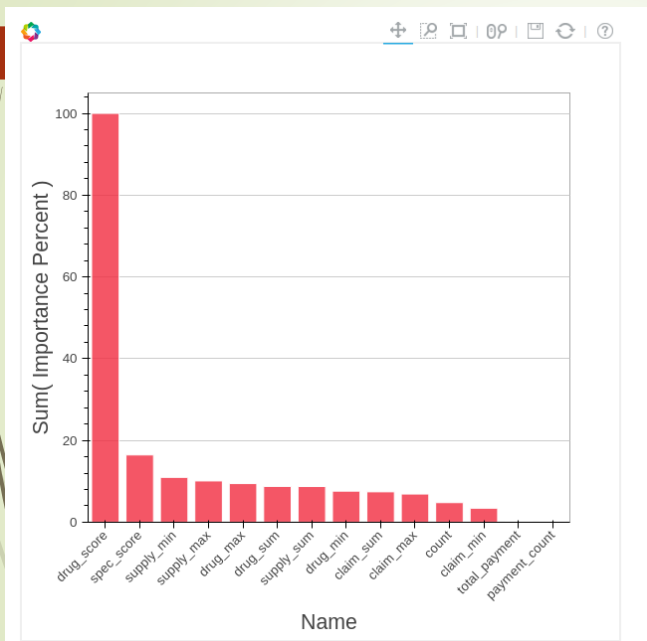
Plots for the data, histogram or bar chart

Fitting results

AUC from RandomForest


AUC = 0.685





Ways to improve

- Use more data, for example the Part-B dataset.
- Add Page rank as a feature.
- Model blend might help.



References

- PageRank for Anomaly Detection by Ofer Mendelevitch and Jiwon Seo
<https://www.versustexas.com/criminal/healthcare-medicare-medicaid-fraud/>