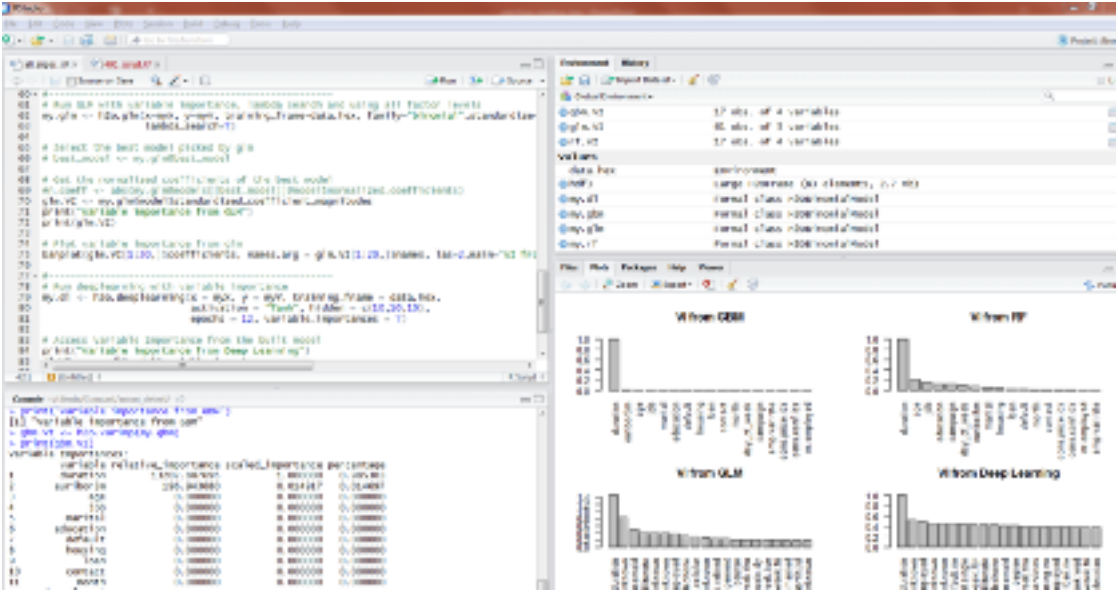
$H_2O$.ai

# R, Python and Flow


Variable Importance from GBM
Variable Importance from RF
Variable Importance from GLM
Variable Importance from Deep Learning

H2O FLOW Flow Cell Data Model Score Admin Help

K-Means_Example

# Setup Parse

PARSE CONFIGURATION

Sources: http://s3.amazonaws.com/h2o-public-test-data/smalldata/flow_examples/seeds_dataset.txt

ID: Key_Frame__http___s3_amazonaws_com_h2o_public_test_data_smalldata_flow_examples_seeds_dataset.hex

Parser: CSV

Separator: HT '\t' (horizontal tab: '0x'

Column Headers: Auto

First row contains column names

First row contains data

Options: Enable single quotes as a field quotation character

Delete on done

EDIT COLUMN NAMES AND TYPES

Search by column name...

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | | Numeric | 15.26 | 14.88 | 14.29 | 13.84 | 16.14 | 14.38 |
| 2 | | Numeric | 14.84 | 14.57 | 14.09 | 13.94 | 14.99 | 14.21 |
| 3 | | Numeric | 0.871 | 0.8811 | 0.905 | 0.8955 | 0.9034 | 0.8951 |
| 4 | | Numeric | 5.763 | 5.554 | 5.291 | 5.324 | 5.658 | 5.386 |
| 5 | | Numeric | 3.312 | 3.333 | 3.337 | 3.379 | 3.562 | 3.312 |
| 6 | | Numeric | 2.221 | 1.018 | 2.699 | 2.259 | 1.355 | 2.462 |
| 7 | | Numeric | 5.22 | 4.956 | 4.825 | 4.805 | 5.175 | 4.956 |
| 8 | | Numeric | 1 | 1 | 1 | 1 | 1 | 1 |

Previous page Next page

Ready

localhost:8888/notebooks/Documents/Python%20Scripts/H2O_chicago_crimes%20(1).ipynb

jupyter H2O_chicago_crimes (1) Last Checkpoint: 20 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Help | Python 3

Code CellToolbar

```
deeplearning Model Build Progress: [##################################################] 100%
```

In [7]:

```
# GBM performance on train/test data
train_auc_gbm = data_gbm.model_performance(train).auc()
test_auc_gbm  = data_gbm.model_performance(test) .auc()

# Deep Learning performance on train/test data
# train_auc_dl = data_dl.model_performance(train).auc()
# test_auc_dl  = data_dl.model_performance(test) .auc()

# Make a pretty HTML table printout of the results
header = ["Model", "AUC Train", "AUC Test"]
table  = [
          ["GBM", train_auc_gbm, test_auc_gbm],
#          ["DL ", train_auc_dl, test_auc_dl]
         ]
h2o.display.H2ODisplay(table, header)
```

| Model | AUC Train | AUC Test |
|---|---|---|
| GBM | 0.9568221 | 0.9007979 |
| DL | 0.8956055 | 0.8841564 |

Out[7]:

| Model | AUC Train | AUC Test |
|---|---|---|
| GBM | 0.9568221 | 0.9007979 |
| DL | 0.8956055 | 0.8841564 |

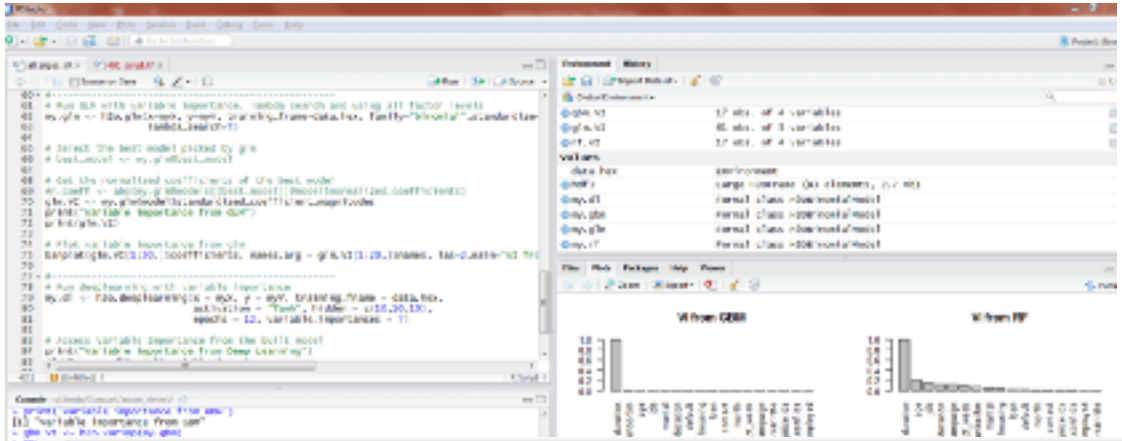
In [8]:

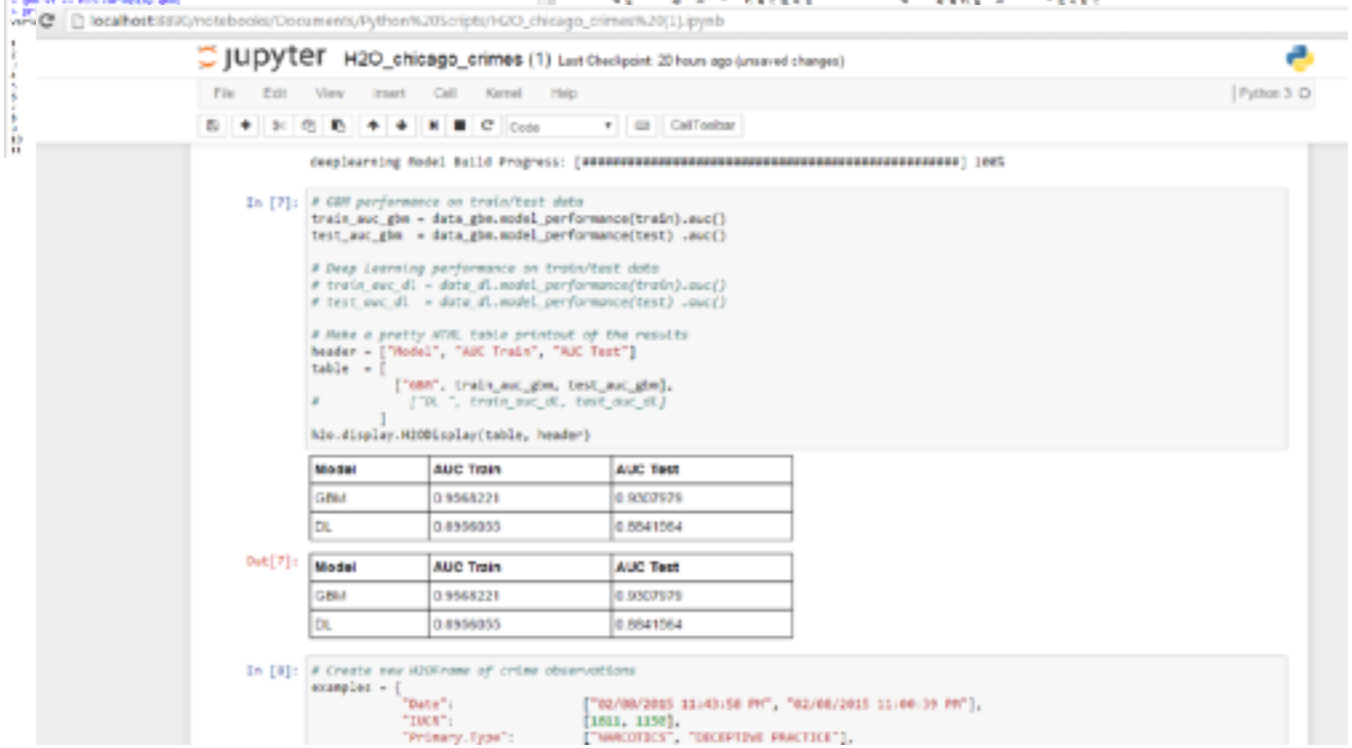
```
# Create new H2OFrame of crime observations
examples = {
            "Date":                  ["02/08/2015 11:43:58 PM", "02/08/2015 11:00:39 PM"],
            "IUCR":                  [1811, 1150],
            "Primary Type":          ["NARCOTICS", "DECEPTIVE PRACTICE"],
```

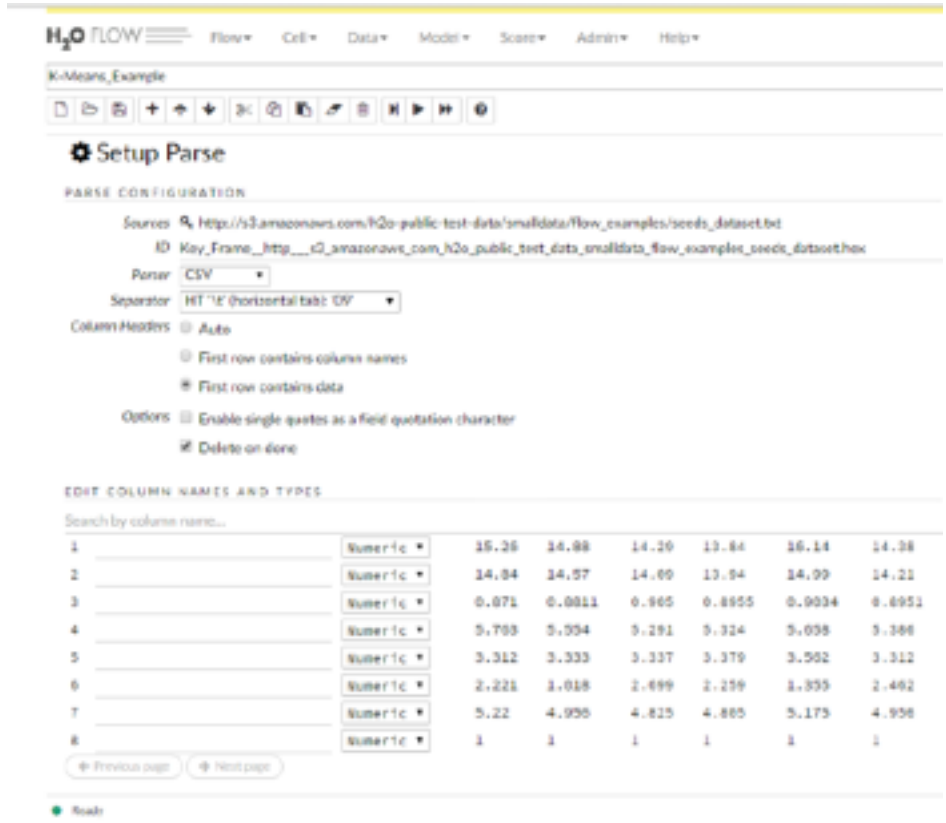
# R Interface Overview

| Action | R | H2O |
|---|---|---|
| Reading data | `read_csv(data_path)` | `h2o.importFile(data_path)` |
| Summarizing data | `summary(data_frame)` | `h2o.summary(h2o_frame)` |
| Summary statistics | `mean(data_frame[["x"]])` | `h2o.mean(h2o_frame)` |
| Combining rows | `rbind(data_frame1, data_frame2)` | `h2o.rbind(h2o_frame1, h2o_frame2)` |
| Combining columns | `cbind(data_frame1, data_frame2)` | `h2o.cbind(h2o_frame1, h2o_frame2)` |
| Data selection | `data_frame[,]` | `h2o_frame[,]` |
| Transforming columns | `log(data_frame[,"x"])`<br>`sqrt(data_frame[,"x"])` | `log(h2o_frame[,"x"])`<br>`sqrt(h2o_frame[,"x"])` |
| Building Random Forest | `model = randomForest(y ~ x, data_frame)` | `model = h2o.randomForest(x, y, train_frame)` |
| Model Prediction | `predict(model, data_frame)` | `h2o.predict(model, h2o_frame)` |
| Model Metrics | `performance(model)`<br>`auc(model)` | `metrics = model.model_performance(frame)`<br>`h2o.auc(model)` |

# R, Python and Flow






H2O.ai