Daily spot price comparison of Amazon EC2* by Decision Tree

Erdi Ölmezoğulları – Kaan Dandin

*Elastic Computing Cloud

Overview

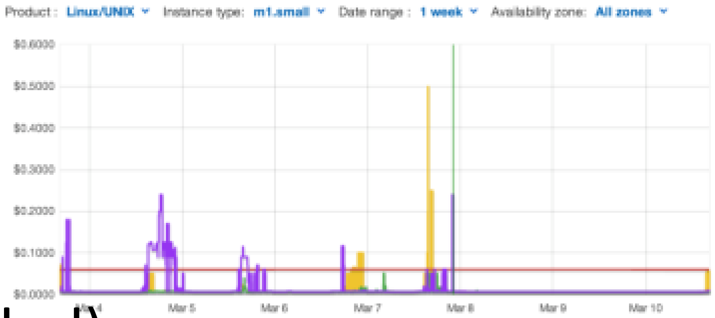
- Briefly introducing to AWS* , EC2 and Spot Price
- Our methodology
- First glance to data
- Challenges (Preprocessing)
- Challenges (Data Exploring)

*Amazon Web Services

AWS and EC2

- AWS which has a bunch of cloud based services is providing to customers (It is really like amazon.com ☺) (e.g. S3, EC2, AmazonDB etc.)
- Amazon EC2 is one of them.
- EC2 is a kind of virtual machine (VM) which is being created in AWS's cloud.
- It has different payment options
  1. Free, but it is really limited one.
  2. Dedicated VM monthly payment over a year.
  3. Hourly payment.
  4. **Spot Price**

Spot Price



Product : Linux/UNIX
Instance type: m1.small
Date range : 1 week
Availability zone: All zones

- It is like stock market (e.g. Borsa Istanbul).
- Or, it can be called as an auction for VMs.
- It can be more cheaper than regular payment options overall. However, VM may be interrupted if market price exceeds our max bid.
- But, Market price is not constant. It is determining in real-time with respect to bid prices which belong to customers.
- Even if the customer has to beat AWS’s market price, the customer is paying just market price.
- If a customer beat market price, his/her machine will be working still. Otherwise, it will be terminating by AWS.

Our Methodology From Scratch

- A few operations were applied in the preprocessing phase before seriously processing data.
- Data was separated to epoch by day-wise.
- So, the data is consisting for 31 days.
- Each model, corresponding to the specific epoch was generated. It means that 31 models will be created at the end of day.
- Since we have 31 models for each day overall, the each model was testified by each day.
- At the end of this comparison, 31*31 (961) accuracy results were generated.

First glance to data

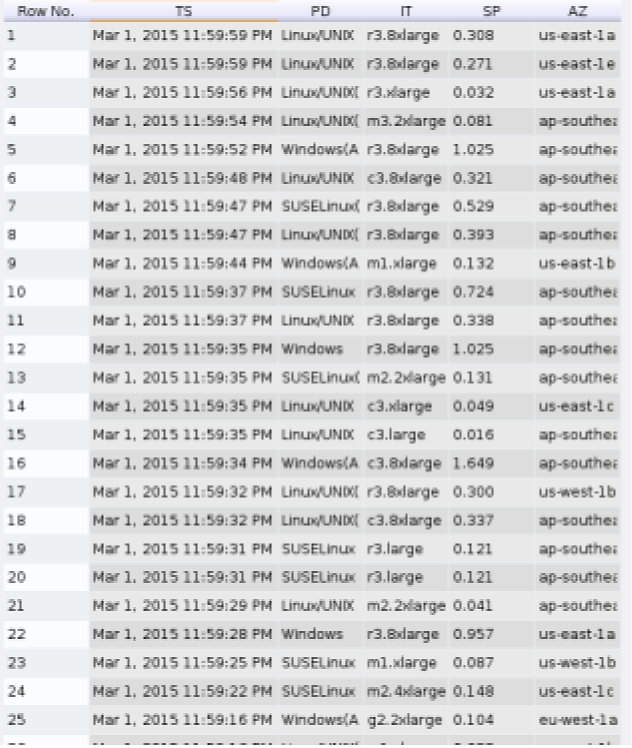
- We retrieved **Amazon Spot Price** data which was collected by a third party person.
- Data has 5 fields **<Timestamp, ProductDescription, InstanceType, SpotPrice, AvailabilityZone>** in **JSON format**.
- According to data fields;

1. Timestamp (TS) is a time when the tuple was collected.
2. Product Description (PD) is referring to kind of operating system on an instance of Virtual Machine (VM) where Operating System will be installed according to customer's demands. It is consisting of 6 unique operating systems.
3. Instance type (IT) is referring to type of VM. Since IT can be picked with respect to business' goal of customer, it was taken down into wide brands with 33 unique VM types by AWS.
4. Spot Price (SP) is showing the current market price for each IT and Availability Zone (AZ).
5. AZ is consisting of 22 unique zones in different countries across the world.

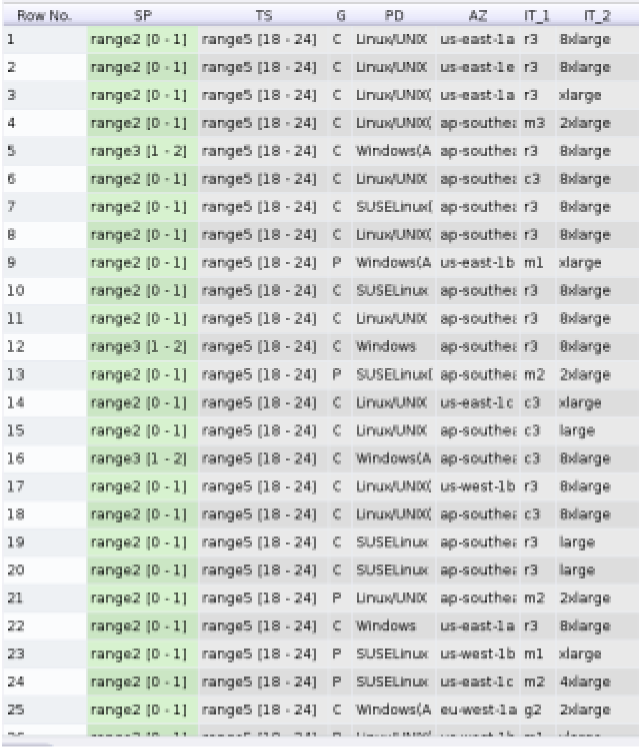
Challenges (Preprocessing)

- To transform data format
- To split data again by day since non-well done.
- To get rid of duplicated data
- To apply discretization (e.g. size or frequency base) for numerical attributes
- To generate a new reasonable attributes from data.
- To pick the reasonable attribute for labelling.

Pre vs Post processing



Row No.	TS	PD	IT	SP	AZ
1	Mar 1, 2015 11:59:59 PM	Linux/UNIX	r3.8xlarge	0.308	us-east-1a
2	Mar 1, 2015 11:59:59 PM	Linux/UNIX	r3.8xlarge	0.271	us-east-1e
3	Mar 1, 2015 11:59:56 PM	Linux/UNIX(	r3.xlarge	0.032	us-east-1a
4	Mar 1, 2015 11:59:54 PM	Linux/UNIX(	m3.2xlarge	0.081	ap-southea
5	Mar 1, 2015 11:59:52 PM	Windows(A	r3.8xlarge	1.025	ap-southea
6	Mar 1, 2015 11:59:48 PM	Linux/UNIX	c3.8xlarge	0.321	ap-southea
7	Mar 1, 2015 11:59:47 PM	SUSELinux(	r3.8xlarge	0.529	ap-southea
8	Mar 1, 2015 11:59:47 PM	Linux/UNIX(	r3.8xlarge	0.393	ap-southea
9	Mar 1, 2015 11:59:44 PM	Windows(A	m1.xlarge	0.132	us-east-1b
10	Mar 1, 2015 11:59:37 PM	SUSELinux	r3.8xlarge	0.724	ap-southea
11	Mar 1, 2015 11:59:37 PM	Linux/UNIX	r3.8xlarge	0.338	ap-southea
12	Mar 1, 2015 11:59:35 PM	Windows	r3.8xlarge	1.025	ap-southea
13	Mar 1, 2015 11:59:35 PM	SUSELinux(	m2.2xlarge	0.131	ap-southea
14	Mar 1, 2015 11:59:35 PM	Linux/UNIX	c3.xlarge	0.049	us-east-1c
15	Mar 1, 2015 11:59:35 PM	Linux/UNIX	c3.large	0.016	ap-southea
16	Mar 1, 2015 11:59:34 PM	Windows(A	c3.8xlarge	1.649	ap-southea
17	Mar 1, 2015 11:59:32 PM	Linux/UNIX(	r3.8xlarge	0.300	us-west-1b
18	Mar 1, 2015 11:59:32 PM	Linux/UNIX(	c3.8xlarge	0.337	ap-southea
19	Mar 1, 2015 11:59:31 PM	SUSELinux	r3.large	0.121	ap-southea
20	Mar 1, 2015 11:59:31 PM	SUSELinux	r3.large	0.121	ap-southea
21	Mar 1, 2015 11:59:29 PM	Linux/UNIX	m2.2xlarge	0.041	ap-southea
22	Mar 1, 2015 11:59:28 PM	Windows	r3.8xlarge	0.957	us-east-1a
23	Mar 1, 2015 11:59:25 PM	SUSELinux	m1.xlarge	0.087	us-west-1b
24	Mar 1, 2015 11:59:22 PM	SUSELinux	m2.4xlarge	0.148	us-east-1c
25	Mar 1, 2015 11:59:16 PM	Windows(A	g2.2xlarge	0.104	eu-west-1a

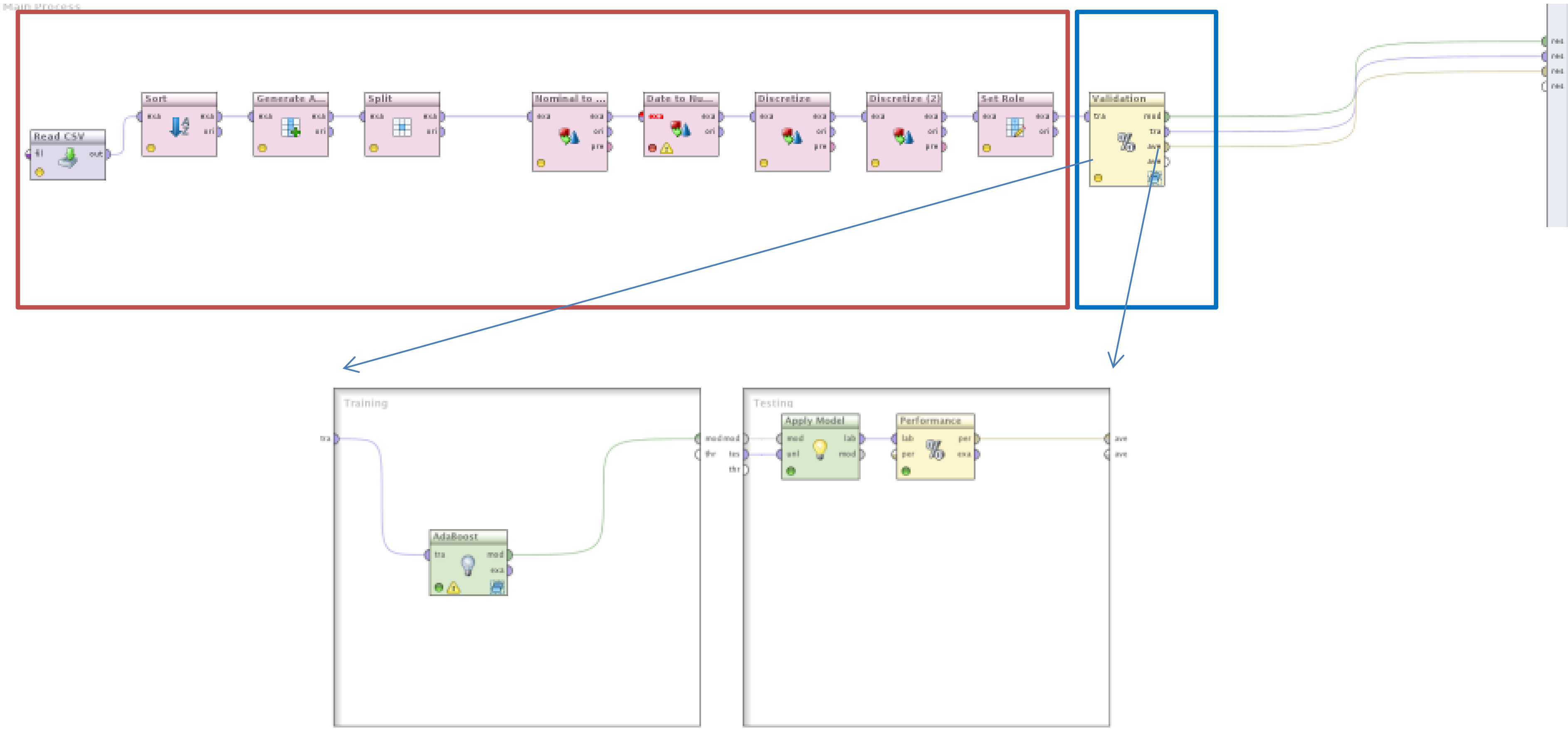


Row No.	SP	TS	G	PD	AZ	IT_1	IT_2
1	range2 [0 - 1]	range5 [18 - 24]	C	Linux/UNIX	us-east-1a	r3	8xlarge
2	range2 [0 - 1]	range5 [18 - 24]	C	Linux/UNIX	us-east-1e	r3	8xlarge
3	range2 [0 - 1]	range5 [18 - 24]	C	Linux/UNIX(	us-east-1a	r3	xlarge
4	range2 [0 - 1]	range5 [18 - 24]	C	Linux/UNIX(	ap-southea	m3	2xlarge
5	range3 [1 - 2]	range5 [18 - 24]	C	Windows(A	ap-southea	r3	8xlarge
6	range2 [0 - 1]	range5 [18 - 24]	C	Linux/UNIX	ap-southea	c3	8xlarge
7	range2 [0 - 1]	range5 [18 - 24]	C	SUSELinux(	ap-southea	r3	8xlarge
8	range2 [0 - 1]	range5 [18 - 24]	C	Linux/UNIX(	ap-southea	r3	8xlarge
9	range2 [0 - 1]	range5 [18 - 24]	P	Windows(A	us-east-1b	m1	xlarge
10	range2 [0 - 1]	range5 [18 - 24]	C	SUSELinux	ap-southea	r3	8xlarge
11	range2 [0 - 1]	range5 [18 - 24]	C	Linux/UNIX	ap-southea	r3	8xlarge
12	range3 [1 - 2]	range5 [18 - 24]	C	Windows	ap-southea	r3	8xlarge
13	range2 [0 - 1]	range5 [18 - 24]	P	SUSELinux(	ap-southea	m2	2xlarge
14	range2 [0 - 1]	range5 [18 - 24]	C	Linux/UNIX	us-east-1c	c3	xlarge
15	range2 [0 - 1]	range5 [18 - 24]	C	Linux/UNIX	ap-southea	c3	large
16	range3 [1 - 2]	range5 [18 - 24]	C	Windows(A	ap-southea	c3	8xlarge
17	range2 [0 - 1]	range5 [18 - 24]	C	Linux/UNIX(	us-west-1b	r3	8xlarge
18	range2 [0 - 1]	range5 [18 - 24]	C	Linux/UNIX(	ap-southea	c3	8xlarge
19	range2 [0 - 1]	range5 [18 - 24]	C	SUSELinux	ap-southea	r3	large
20	range2 [0 - 1]	range5 [18 - 24]	C	SUSELinux	ap-southea	r3	large
21	range2 [0 - 1]	range5 [18 - 24]	P	Linux/UNIX	ap-southea	m2	2xlarge
22	range2 [0 - 1]	range5 [18 - 24]	C	Windows	us-east-1a	r3	8xlarge
23	range2 [0 - 1]	range5 [18 - 24]	P	SUSELinux	us-west-1b	m1	xlarge
24	range2 [0 - 1]	range5 [18 - 24]	P	SUSELinux	us-east-1c	m2	4xlarge
25	range2 [0 - 1]	range5 [18 - 24]	C	Windows(A	eu-west-1a	g2	2xlarge

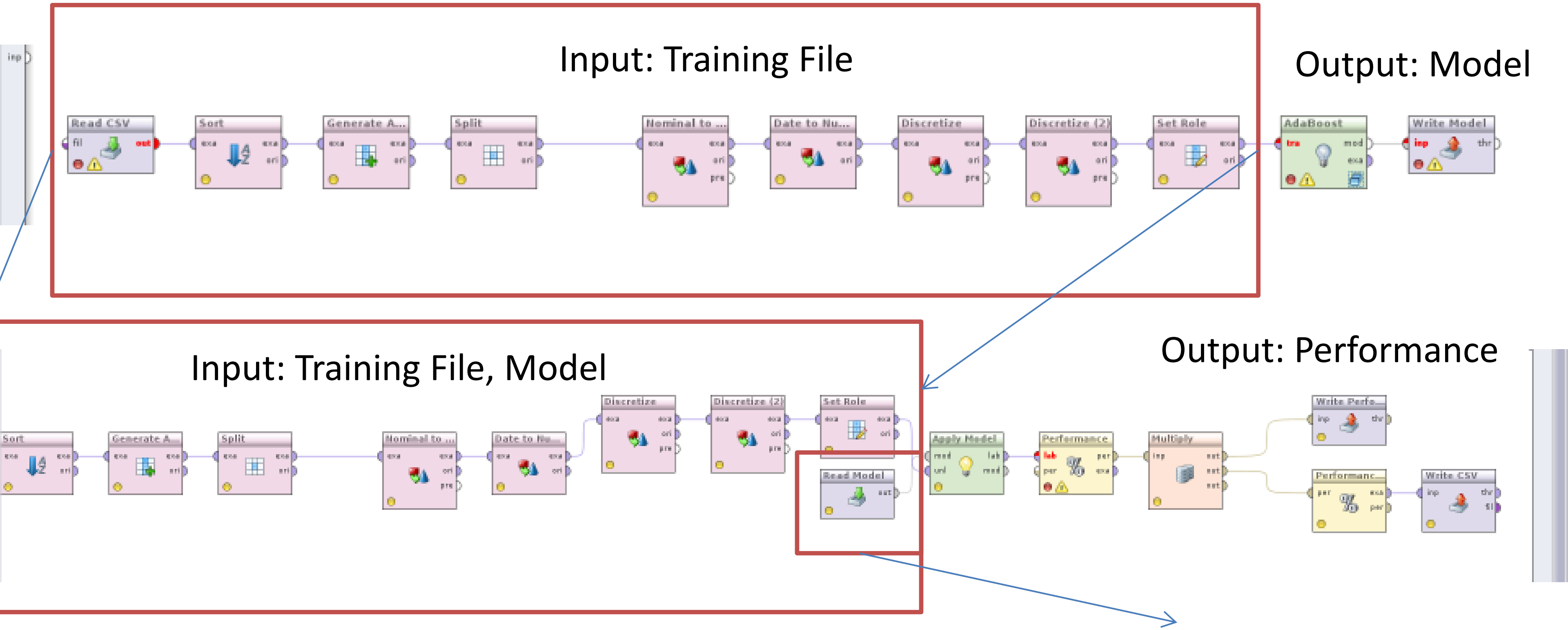
Challenges (Data Exploring)

- To generate an accurate decision tree for each epoch, we referred to **AdaBoost**.
- It depends on many parameters. But, we tune on parameters below within limited time.
  - Number of iterations
  - The depth of tree
  - Split size of node in the tree
  - Pre and post-pruning parameters
  - **The numerical attribute, spot price since it is our labeled attribute (polynomial) → Discretization is important!**
  - **Labelling is too important!**

Our 1st Design in Rapidminer



Our 2nd Design in Rapidmider



Input: Training File

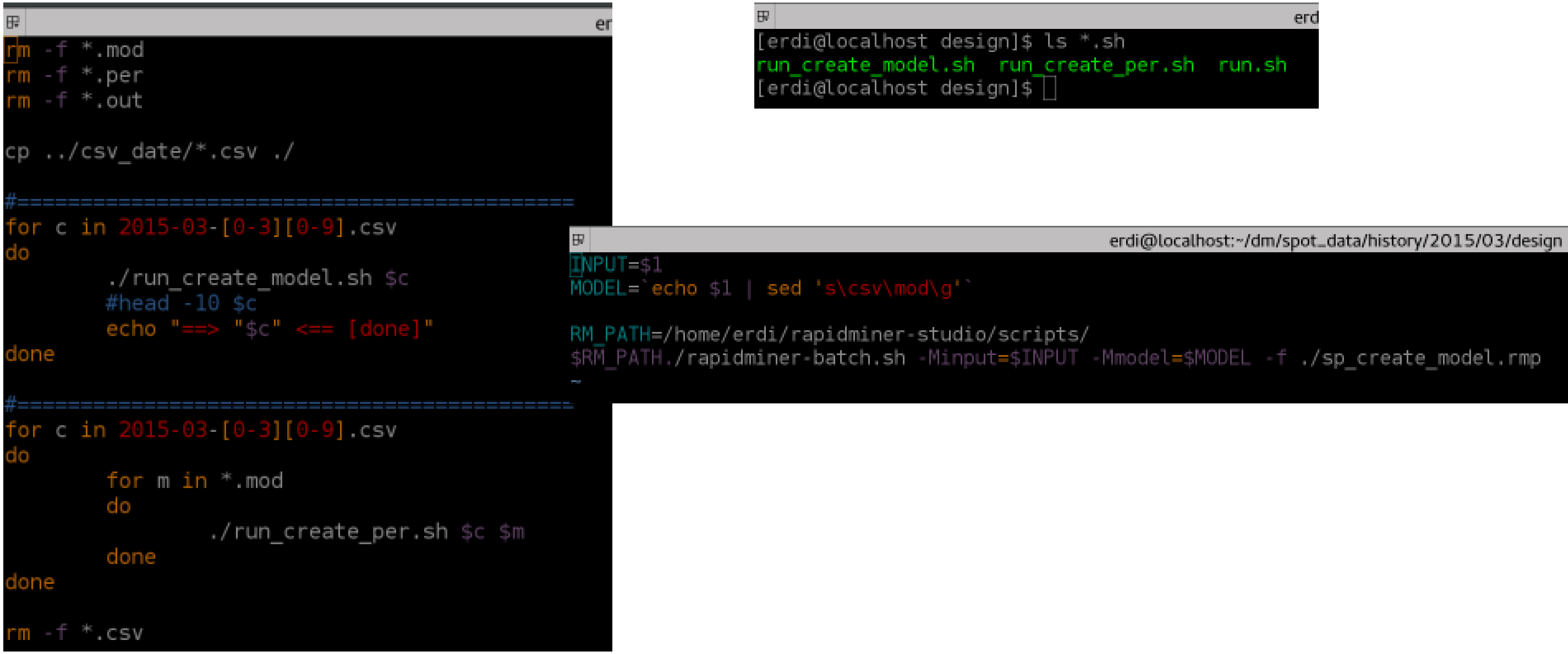
Output: Model

Input: Training File, Model

Output: Performance

Model

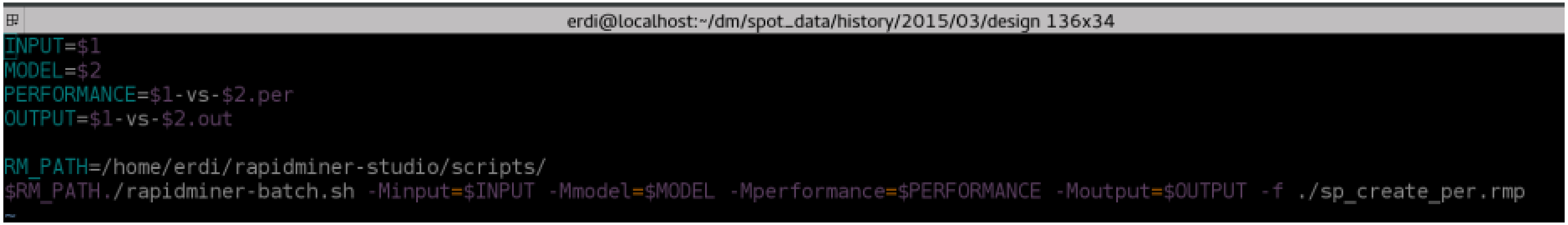
Our Shell Scripts



```
rm -f *.mod
rm -f *.per
rm -f *.out

cp ../csv_date/*.csv ./

#=============================================
for c in 2015-03-[0-3][0-9].csv
do
        ./run_create_model.sh $c
        #head -10 $c
        echo "==> "$c" <== [done]"
done

#=============================================
for c in 2015-03-[0-3][0-9].csv
do
        for m in *.mod
        do
                ./run_create_per.sh $c $m
        done
done

rm -f *.csv
```

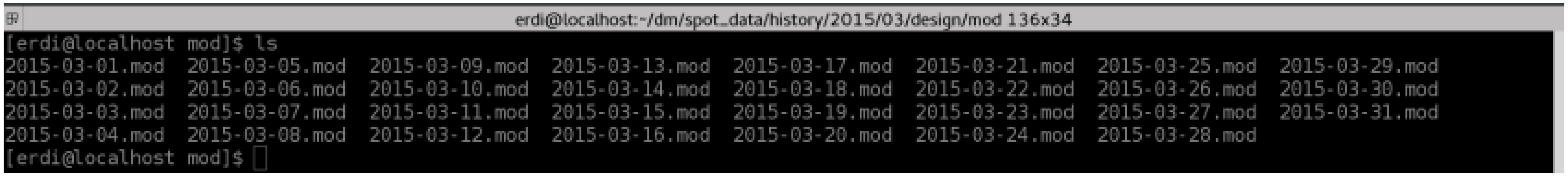
```
[erdi@localhost design]$ ls *.sh
run_create_model.sh  run_create_per.sh  run.sh
[erdi@localhost design]$ 
```

```
INPUT=$1
MODEL=`echo $1 | sed 's\csv\mod\g'`

RM_PATH=/home/erdi/rapidminer-studio/scripts/
$RM_PATH./rapidminer-batch.sh -Minput=$INPUT -Mmodel=$MODEL -f ./sp_create_model.rmp
~
```

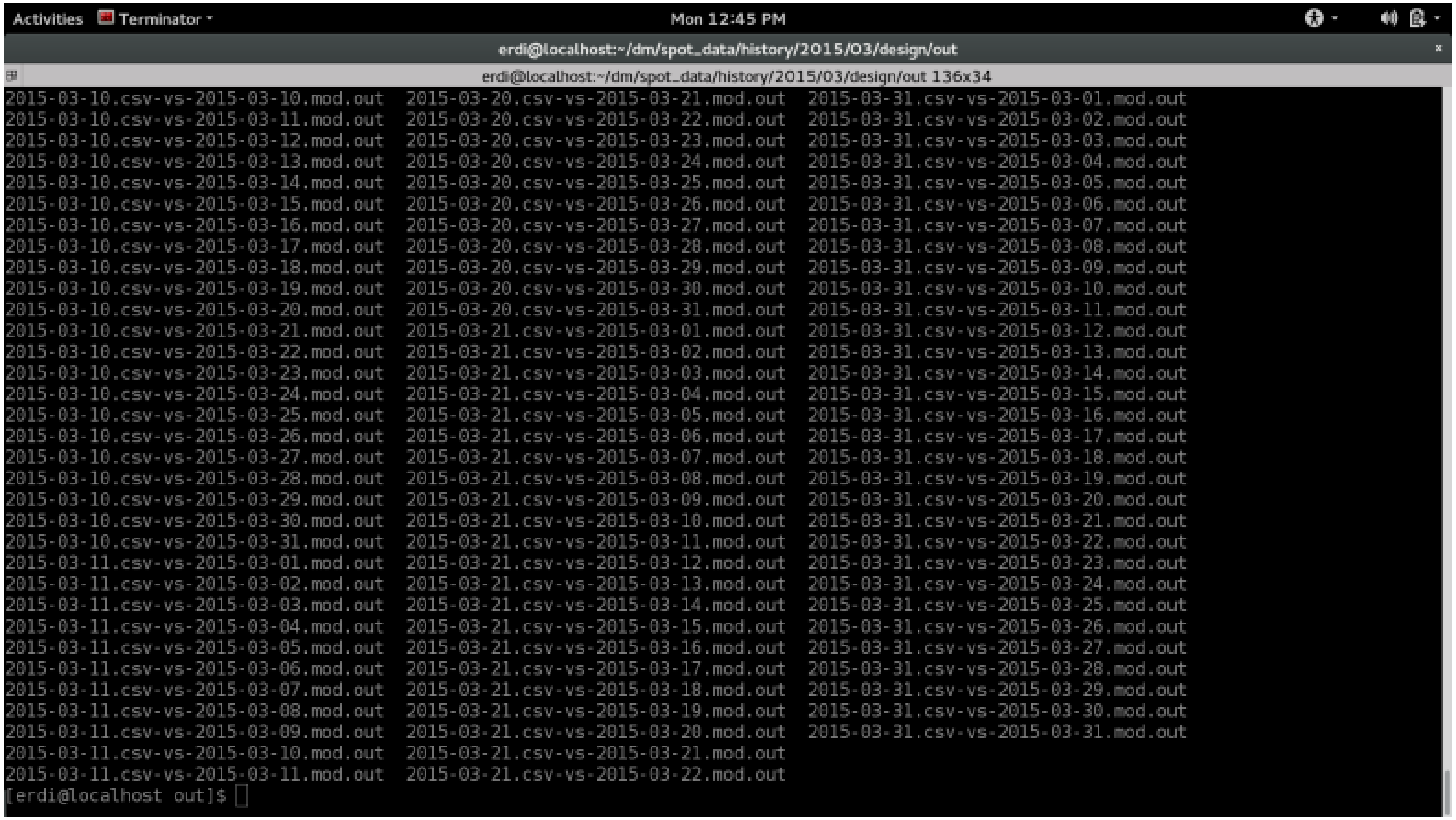


```
INPUT=$1
MODEL=$2
PERFORMANCE=$1-vs-$2.per
OUTPUT=$1-vs-$2.out

RM_PATH=/home/erdi/rapidminer-studio/scripts/
$RM_PATH./rapidminer-batch.sh -Minput=$INPUT -Mmodel=$MODEL -Mperformance=$PERFORMANCE -Moutput=$OUTPUT -f ./sp_create_per.rmp
~
```

Results (Models and Performances)

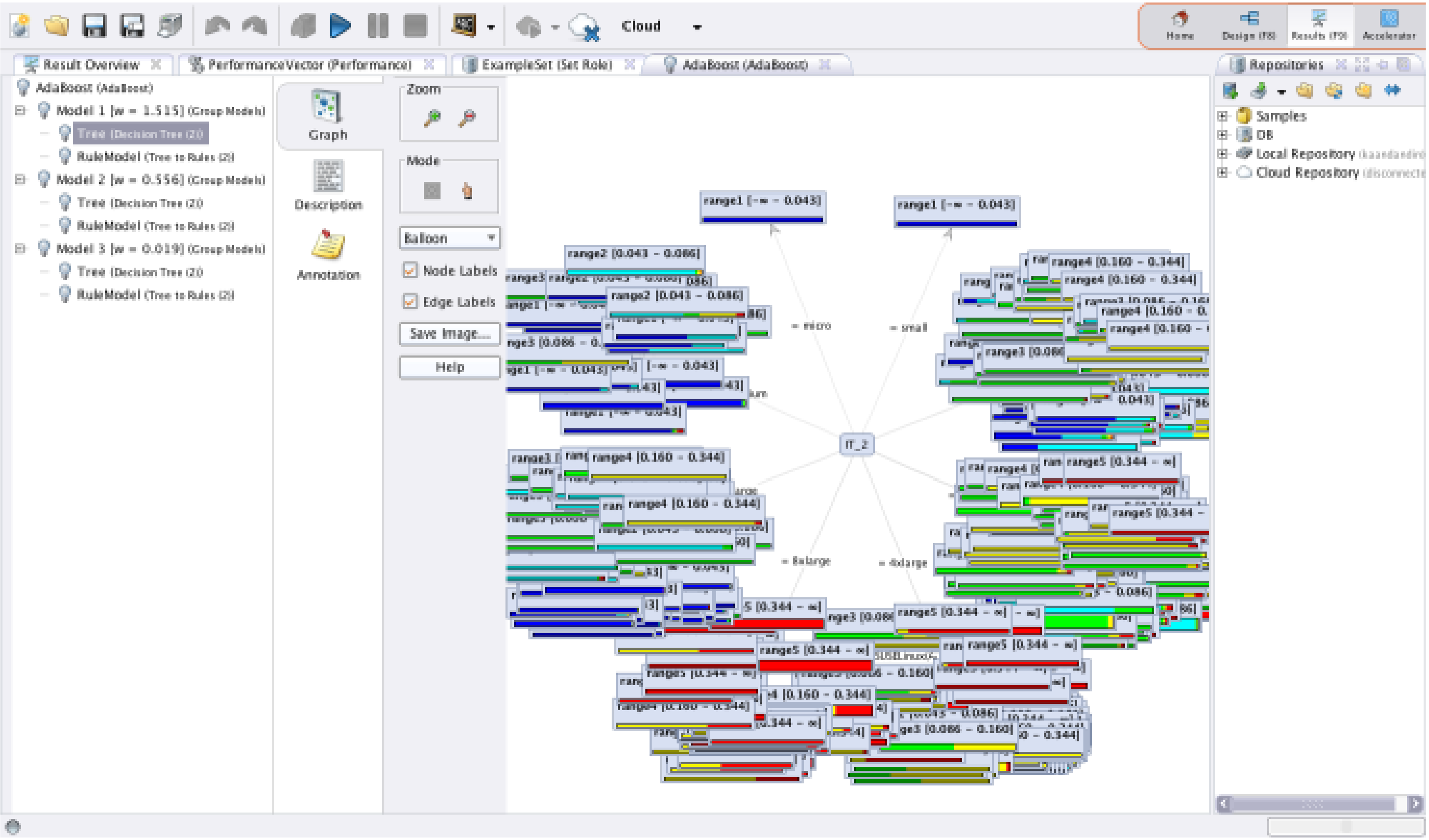

erdi@localhost:~/dm/spot_data/history/2015/03/design/mod 136x34

```
[erdi@localhost mod]$ ls
2015-03-01.mod  2015-03-05.mod  2015-03-09.mod  2015-03-13.mod  2015-03-17.mod  2015-03-21.mod  2015-03-25.mod  2015-03-29.mod
2015-03-02.mod  2015-03-06.mod  2015-03-10.mod  2015-03-14.mod  2015-03-18.mod  2015-03-22.mod  2015-03-26.mod  2015-03-30.mod
2015-03-03.mod  2015-03-07.mod  2015-03-11.mod  2015-03-15.mod  2015-03-19.mod  2015-03-23.mod  2015-03-27.mod  2015-03-31.mod
2015-03-04.mod  2015-03-08.mod  2015-03-12.mod  2015-03-16.mod  2015-03-20.mod  2015-03-24.mod  2015-03-28.mod
[erdi@localhost mod]$ 
```

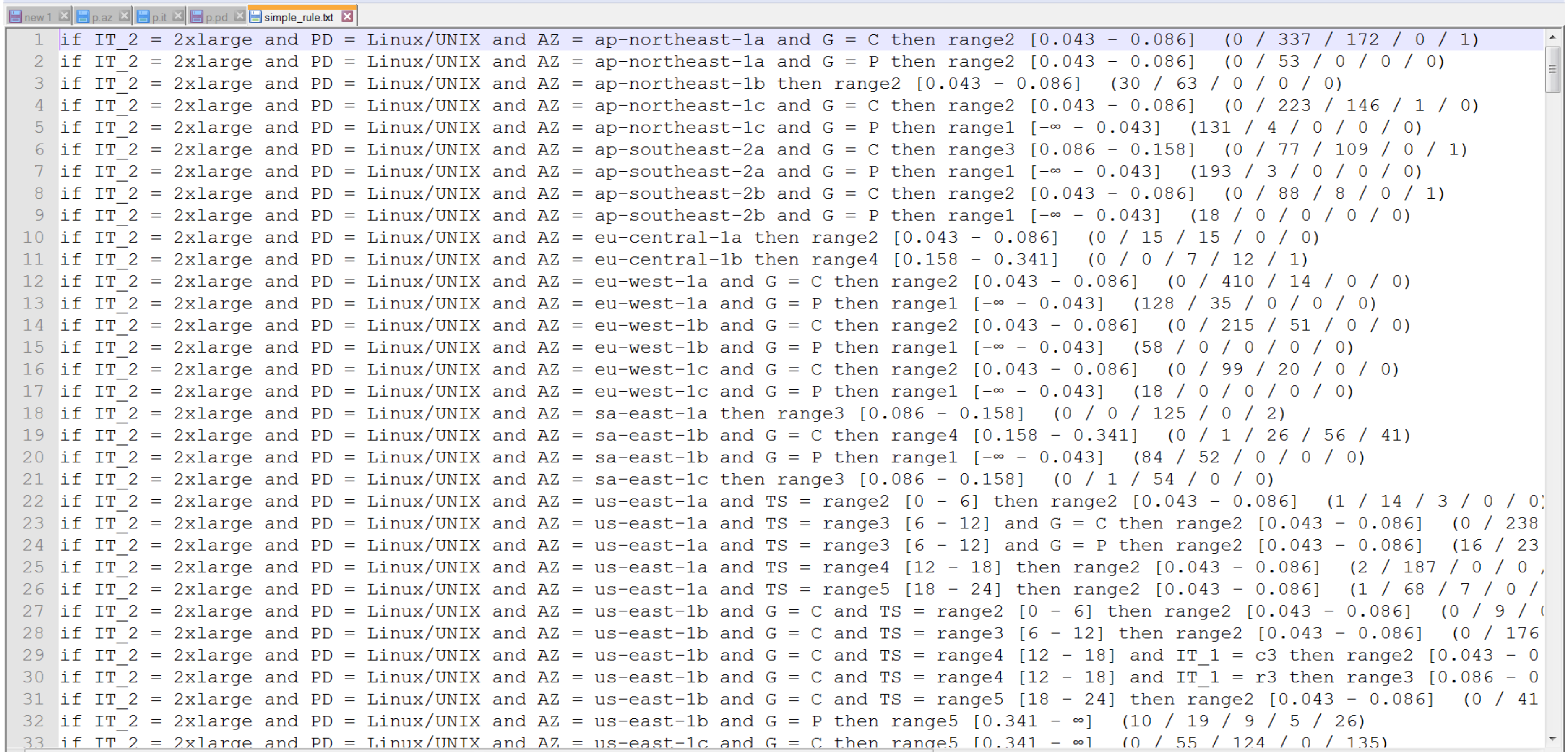

Activities
Terminator
Mon 12:45 PM
erdi@localhost:~/dm/spot_data/history/2015/03/design/out
erdi@localhost:~/dm/spot_data/history/2015/03/design/out 136x34

```
2015-03-10.csv-vs-2015-03-10.mod.out  2015-03-20.csv-vs-2015-03-21.mod.out  2015-03-31.csv-vs-2015-03-01.mod.out
2015-03-10.csv-vs-2015-03-11.mod.out  2015-03-20.csv-vs-2015-03-22.mod.out  2015-03-31.csv-vs-2015-03-02.mod.out
2015-03-10.csv-vs-2015-03-12.mod.out  2015-03-20.csv-vs-2015-03-23.mod.out  2015-03-31.csv-vs-2015-03-03.mod.out
2015-03-10.csv-vs-2015-03-13.mod.out  2015-03-20.csv-vs-2015-03-24.mod.out  2015-03-31.csv-vs-2015-03-04.mod.out
2015-03-10.csv-vs-2015-03-14.mod.out  2015-03-20.csv-vs-2015-03-25.mod.out  2015-03-31.csv-vs-2015-03-05.mod.out
2015-03-10.csv-vs-2015-03-15.mod.out  2015-03-20.csv-vs-2015-03-26.mod.out  2015-03-31.csv-vs-2015-03-06.mod.out
2015-03-10.csv-vs-2015-03-16.mod.out  2015-03-20.csv-vs-2015-03-27.mod.out  2015-03-31.csv-vs-2015-03-07.mod.out
2015-03-10.csv-vs-2015-03-17.mod.out  2015-03-20.csv-vs-2015-03-28.mod.out  2015-03-31.csv-vs-2015-03-08.mod.out
2015-03-10.csv-vs-2015-03-18.mod.out  2015-03-20.csv-vs-2015-03-29.mod.out  2015-03-31.csv-vs-2015-03-09.mod.out
2015-03-10.csv-vs-2015-03-19.mod.out  2015-03-20.csv-vs-2015-03-30.mod.out  2015-03-31.csv-vs-2015-03-10.mod.out
2015-03-10.csv-vs-2015-03-20.mod.out  2015-03-20.csv-vs-2015-03-31.mod.out  2015-03-31.csv-vs-2015-03-11.mod.out
2015-03-10.csv-vs-2015-03-21.mod.out  2015-03-21.csv-vs-2015-03-01.mod.out  2015-03-31.csv-vs-2015-03-12.mod.out
2015-03-10.csv-vs-2015-03-22.mod.out  2015-03-21.csv-vs-2015-03-02.mod.out  2015-03-31.csv-vs-2015-03-13.mod.out
2015-03-10.csv-vs-2015-03-23.mod.out  2015-03-21.csv-vs-2015-03-03.mod.out  2015-03-31.csv-vs-2015-03-14.mod.out
2015-03-10.csv-vs-2015-03-24.mod.out  2015-03-21.csv-vs-2015-03-04.mod.out  2015-03-31.csv-vs-2015-03-15.mod.out
2015-03-10.csv-vs-2015-03-25.mod.out  2015-03-21.csv-vs-2015-03-05.mod.out  2015-03-31.csv-vs-2015-03-16.mod.out
2015-03-10.csv-vs-2015-03-26.mod.out  2015-03-21.csv-vs-2015-03-06.mod.out  2015-03-31.csv-vs-2015-03-17.mod.out
2015-03-10.csv-vs-2015-03-27.mod.out  2015-03-21.csv-vs-2015-03-07.mod.out  2015-03-31.csv-vs-2015-03-18.mod.out
2015-03-10.csv-vs-2015-03-28.mod.out  2015-03-21.csv-vs-2015-03-08.mod.out  2015-03-31.csv-vs-2015-03-19.mod.out
2015-03-10.csv-vs-2015-03-29.mod.out  2015-03-21.csv-vs-2015-03-09.mod.out  2015-03-31.csv-vs-2015-03-20.mod.out
2015-03-10.csv-vs-2015-03-30.mod.out  2015-03-21.csv-vs-2015-03-10.mod.out  2015-03-31.csv-vs-2015-03-21.mod.out
2015-03-10.csv-vs-2015-03-31.mod.out  2015-03-21.csv-vs-2015-03-11.mod.out  2015-03-31.csv-vs-2015-03-22.mod.out
2015-03-11.csv-vs-2015-03-01.mod.out  2015-03-21.csv-vs-2015-03-12.mod.out  2015-03-31.csv-vs-2015-03-23.mod.out
2015-03-11.csv-vs-2015-03-02.mod.out  2015-03-21.csv-vs-2015-03-13.mod.out  2015-03-31.csv-vs-2015-03-24.mod.out
2015-03-11.csv-vs-2015-03-03.mod.out  2015-03-21.csv-vs-2015-03-14.mod.out  2015-03-31.csv-vs-2015-03-25.mod.out
2015-03-11.csv-vs-2015-03-04.mod.out  2015-03-21.csv-vs-2015-03-15.mod.out  2015-03-31.csv-vs-2015-03-26.mod.out
2015-03-11.csv-vs-2015-03-05.mod.out  2015-03-21.csv-vs-2015-03-16.mod.out  2015-03-31.csv-vs-2015-03-27.mod.out
2015-03-11.csv-vs-2015-03-06.mod.out  2015-03-21.csv-vs-2015-03-17.mod.out  2015-03-31.csv-vs-2015-03-28.mod.out
2015-03-11.csv-vs-2015-03-07.mod.out  2015-03-21.csv-vs-2015-03-18.mod.out  2015-03-31.csv-vs-2015-03-29.mod.out
2015-03-11.csv-vs-2015-03-08.mod.out  2015-03-21.csv-vs-2015-03-19.mod.out  2015-03-31.csv-vs-2015-03-30.mod.out
2015-03-11.csv-vs-2015-03-09.mod.out  2015-03-21.csv-vs-2015-03-20.mod.out  2015-03-31.csv-vs-2015-03-31.mod.out
2015-03-11.csv-vs-2015-03-10.mod.out  2015-03-21.csv-vs-2015-03-21.mod.out
2015-03-11.csv-vs-2015-03-11.mod.out  2015-03-21.csv-vs-2015-03-22.mod.out
[erdi@localhost out]$ 
```

Sample Tree Results

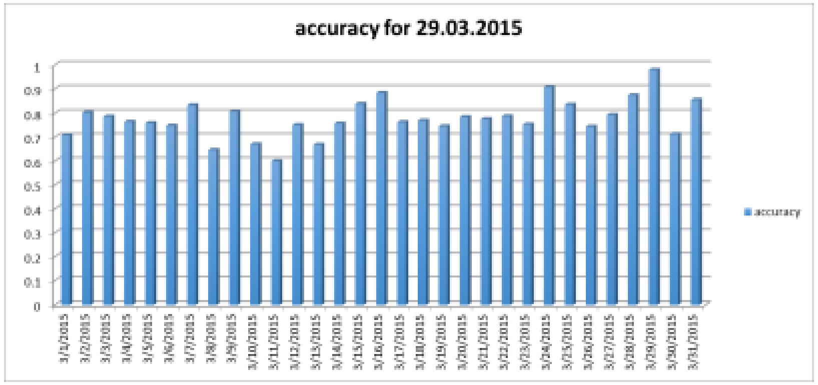


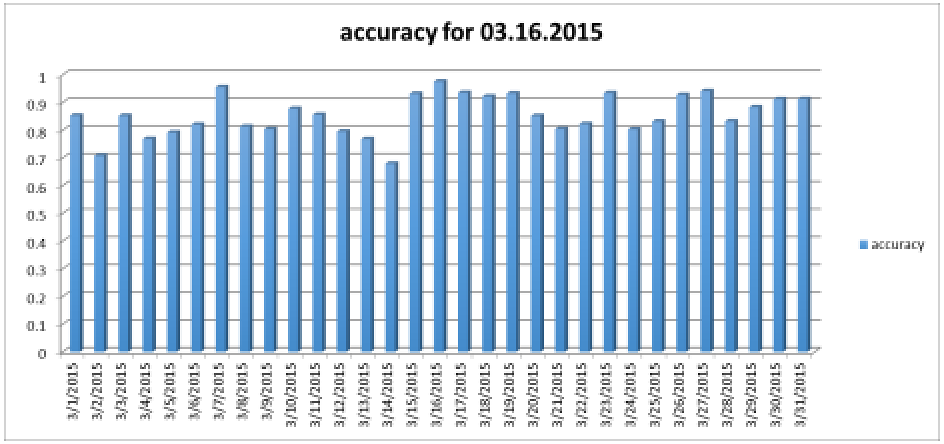
Sample Rules over Decision Tree

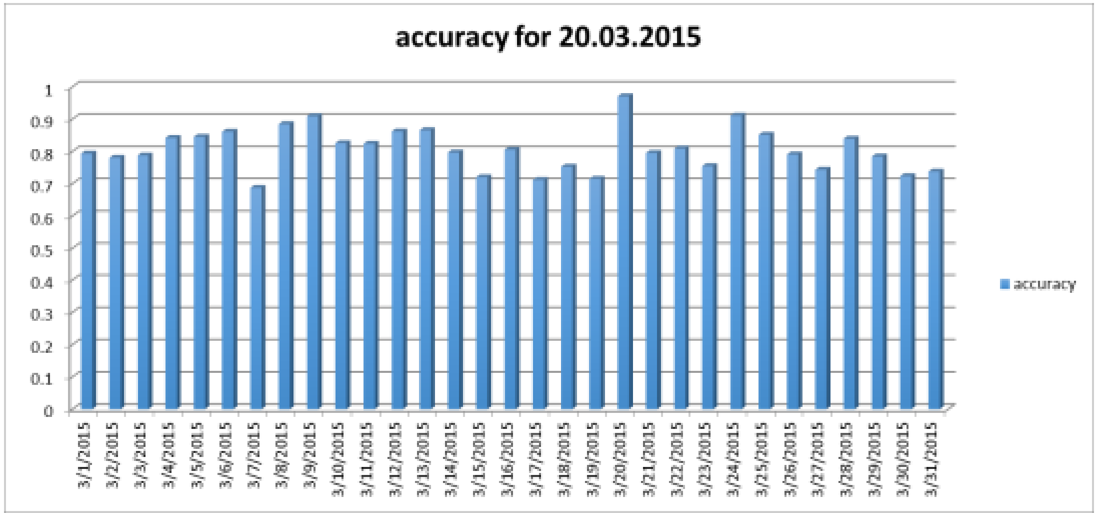


```
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = ap-northeast-1a and G = C then range2 [0.043 - 0.086]  (0 / 337 / 172 / 0 / 1)
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = ap-northeast-1a and G = P then range2 [0.043 - 0.086]  (0 / 53 / 0 / 0 / 0)
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = ap-northeast-1b then range2 [0.043 - 0.086]  (30 / 63 / 0 / 0 / 0)
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = ap-northeast-1c and G = C then range2 [0.043 - 0.086]  (0 / 223 / 146 / 1 / 0)
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = ap-northeast-1c and G = P then range1 [-∞ - 0.043]  (131 / 4 / 0 / 0 / 0)
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = ap-southeast-2a and G = C then range3 [0.086 - 0.158]  (0 / 77 / 109 / 0 / 1)
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = ap-southeast-2a and G = P then range1 [-∞ - 0.043]  (193 / 3 / 0 / 0 / 0)
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = ap-southeast-2b and G = C then range2 [0.043 - 0.086]  (0 / 88 / 8 / 0 / 1)
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = ap-southeast-2b and G = P then range1 [-∞ - 0.043]  (18 / 0 / 0 / 0 / 0)
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = eu-central-1a then range2 [0.043 - 0.086]  (0 / 15 / 15 / 0 / 0)
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = eu-central-1b then range4 [0.158 - 0.341]  (0 / 0 / 7 / 12 / 1)
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = eu-west-1a and G = C then range2 [0.043 - 0.086]  (0 / 410 / 14 / 0 / 0)
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = eu-west-1a and G = P then range1 [-∞ - 0.043]  (128 / 35 / 0 / 0 / 0)
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = eu-west-1b and G = C then range2 [0.043 - 0.086]  (0 / 215 / 51 / 0 / 0)
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = eu-west-1b and G = P then range1 [-∞ - 0.043]  (58 / 0 / 0 / 0 / 0)
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = eu-west-1c and G = C then range2 [0.043 - 0.086]  (0 / 99 / 20 / 0 / 0)
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = eu-west-1c and G = P then range1 [-∞ - 0.043]  (18 / 0 / 0 / 0 / 0)
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = sa-east-1a then range3 [0.086 - 0.158]  (0 / 0 / 125 / 0 / 2)
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = sa-east-1b and G = C then range4 [0.158 - 0.341]  (0 / 1 / 26 / 56 / 41)
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = sa-east-1b and G = P then range1 [-∞ - 0.043]  (84 / 52 / 0 / 0 / 0)
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = sa-east-1c then range3 [0.086 - 0.158]  (0 / 1 / 54 / 0 / 0)
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = us-east-1a and TS = range2 [0 - 6] then range2 [0.043 - 0.086]  (1 / 14 / 3 / 0 / 0)
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = us-east-1a and TS = range3 [6 - 12] and G = C then range2 [0.043 - 0.086]  (0 / 238
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = us-east-1a and TS = range3 [6 - 12] and G = P then range2 [0.043 - 0.086]  (16 / 23
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = us-east-1a and TS = range4 [12 - 18] then range2 [0.043 - 0.086]  (2 / 187 / 0 / 0 /
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = us-east-1a and TS = range5 [18 - 24] then range2 [0.043 - 0.086]  (1 / 68 / 7 / 0 /
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = us-east-1b and G = C and TS = range2 [0 - 6] then range2 [0.043 - 0.086]  (0 / 9 /
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = us-east-1b and G = C and TS = range3 [6 - 12] then range2 [0.043 - 0.086]  (0 / 176
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = us-east-1b and G = C and TS = range4 [12 - 18] and IT_1 = c3 then range2 [0.043 - 0
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = us-east-1b and G = C and TS = range4 [12 - 18] and IT_1 = r3 then range3 [0.086 - 0
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = us-east-1b and G = C and TS = range5 [18 - 24] then range2 [0.043 - 0.086]  (0 / 41
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = us-east-1b and G = P then range5 [0.341 - ∞]  (10 / 19 / 9 / 5 / 26)
if IT_2 = 2xlarge and PD = Linux/UNIX and AZ = us-east-1c and G = C then range5 [0.341 - ∞]  (0 / 55 / 124 / 0 / 135)
```

The model which has lowest accuracy


accuracy for 29.03.2015


accuracy for 03.16.2015


accuracy for 20.03.2015

Conclusion

- It is possible to help customers for offering the correct price range according to their demands, time of day and machine type
- This information will help for customers to give the best possible price and decrease their cost of cloud services usage
- In future work,
  - at the end of the this process, a emergence rules can be revealed by using rule induction.
  - This project can be implemented by using stream processing technologies (e.g. Esper, Apache Spark)