ork New folder New folder Google Analytics Outlook Web App imac - Google Sea... CS372: Solutions ... single.dvi Google Cloud Plat... Nonpa

Overview Data Kernels Discussion **Leaderboard** Rules Team My Submissions **Submit Predictions**

#	△1w	Team Name	Kernel	Team Members	Score	Entries	Last
1	new	**Sammed Kagi**			0.08783	3	3d
2	▾1	**Shubham**			0.72880	1	8d
3	▾1	**Nipun Batra**			565.24888	1	8d

Your Best Entry ↑

Your submission scored 565.24888, which is not an improvement of your best score. Keep trying!

#	△1w	Team Name	Kernel	Team Members	Score	Entries	Last
4	new	**Deepanshu Singh**			12149.825...	1	11h
5	new	**Darshita**			25971.881...	2	1d
6	new	**RaunakSwarnkar**			30855.104...	2	~10s
7	new	**ayushgarg34**			33810.031...	1	14h
8	new	**Shubham Garg**			36199.514...	1	12h
9	new	**chandan**			38190.305...	3	19h
10	new	**karan kumar**			38454.845...	2	3d
11	new	**Nitiksha**			38666.247...	1	16h
12	new	**Dhananjay**			39172.401...	1	17m
13	new	**Chinmay Sonar**			39235.754...	2	1d
14	new	**RA GM1**			39361.001	5	16h

Practical Machine Learning

Kaggle leaderboard

Kaggle (A few baseline models, ...)

① output = Mean(Train)

② o/p = mode(~~Tr~~) or Median(Train)

③ o/p = Random(Min(Train), Max(Train))

Feature selection

To predict : Sales price
Features : Tire size, Blade-Type, ...

$\underbrace{\hspace{6em}}$ 40 odd features.

Brute force / Exhaustive enumeration

length	Features
1	Tire size
1	Blade-Type
	⋮
2	(Tire-size, Blade-Type)
	Tire-size, year of manufacturing . . .

$F_1 \ldots\ldots F_d$

$$\left.\begin{array}{l} 1\,0\,0\,0\,0 \ldots\ldots 0 \\ 0\,1\,0\,0\,0\,0\,0\,0\,0 \ldots 0 \\ \vdots \end{array}\right\}$$ Feature of length 1

$$\left.\begin{array}{l} 1\,1\,0\,0 \ldots\ldots 0 \\ 0\,1\,1 \ldots\ldots 0 \end{array}\right\}$$

$1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\ldots 1$ length d

$2^d - 1$ feature combinat^ns...

1. Train 2^d such classifiers/regressors
2. See the performance of 2^d on valid^n set
3. Pick the best performing one

Stepwise Forward Selection (SFS) : greedy

sel_features = { }

for i = 1 to d :

for feature in $f_1 \ldots f_d$:

train [feature]

valid_performance [sel_feature ∪ feature]

find feature which has best valid^n performance

sel_features = sel_features ∪ $feature_{best}$

f_1, f_2, f_3, f_4

S = { }

Valid^n score

$\mathcal{F}$	Valid ~~Score~~ Accuracy
$\boxed{F_1}$	$\boxed{1}$
F_2	0
F_3	.5
F_4	.8

$$S = \{\} \cup F_1 = \{F_1\}$$

$\mathcal{F}$	Accuracy
F_1, F_2	.8
$\boxed{F_1, F_3}$	$\boxed{.9}$
F_1, F_4	.7

$$S = \{F_1\} \cup \{F_3\} = \{F_1, F_3\}$$

Step wise

Backward selectⁿ

* $S = \{F_1, \ldots F_d\}$
* Same as SFS, but in opp. directⁿ
* Remove feature, which reduces the accuracy the least...

$$SFS \ \& \ SBS = O(d^2)$$

$$(d) + (d-1) + \ldots \quad (1)$$

$$\leq d \quad \leq d \quad \ldots$$

$$\leq d(d)$$

$$\leq O(d^2)$$