Walmart Sales Prediction

Justin Hilliard, Gus Henry

The Challenge

- Predict sales of weather-sensitive items
- During “significant weather events”
- Given weather data

  past sales for various stores

What we are given

- train.csv - training data, previous sales numbers for each store and day

	A	B	C	D
1	date	store_nbr	item_nbr	units
2	1/1/12	1	1	0
3	1/1/12	1	2	0
4	1/1/12	1	3	0
5	1/1/12	1	4	0
6	1/1/12	1	5	0
7	1/1/12	1	6	0
8	1/1/12	1	7	0
9	1/1/12	1	8	0
10	1/1/12	1	9	29

What we are given

- weather.csv - weather data for each date and station

station_nbr	date	tmax	tmin	tavg	depart	dewpoint	wetbulb	heat	cool
1	1/1/12	52	31	42	M	36	40	23	0
2	1/1/12	48	33	41	16	37	39	24	0
3	1/1/12	55	34	45	9	24	36	20	0
4	1/1/12	63	47	55	4	28	43	10	0
6	1/1/12	63	34	49	0	31	43	16	0
7	1/1/12	50	33	42	M	26	35	23	0
8	1/1/12	66	45	M	M	34	46	M	M
9	1/1/12	34	19	27	M	17	23	38	0
10	1/1/12	73	53	63	M	55	58	2	0

sunrise	sunset	codesum	snowfall	preciptotal	stnpressure	sealevel	resultspeed	resultdir	avgspeed
-	-	RA FZFG BR	M	0.05	29.78	29.92	3.6	20	4.6
716	1626	RA	0	0.07	28.82	29.91	9.1	23	11.3
735	1720		0	0	29.77	30.47	9.9	31	10
728	1742		0	0	29.79	30.48	8	35	8.2
727	1742		0	0	29.95	30.47	14	36	13.8
-	-		0	0	29.15	30.54	10.3	32	10.2
-	-	RA BR	M	0	30.05	M	11	36	10.9
-	-	UP	M	T	29.34	30.09	22.8	30	22.5
723	1738	FG+ FG BR	M	0	30.16	30.19	5.1	24	5.5

What we are given

- key.csv - table linking each store to a weather station

store_nbr	station_nbr
1	1
2	14
3	7
4	9
5	12
6	14
7	6
8	4
9	17
10	12
11	10

What we are given

- test.csv - dates, stores and items on which to test

date	store_nbr	item_nbr
4/1/13	2	1
4/1/13	2	2
4/1/13	2	3
4/1/13	2	4
4/1/13	2	5
4/1/13	2	6
4/1/13	2	7
4/1/13	2	8
4/1/13	2	9
4/1/13	2	10
4/1/13	2	11
4/1/13	2	12
4/1/13	2	13
4/1/13	2	14
4/1/13	2	15

Our Approach

- *k*-nearest neighbors algorithm
- loaded and reformatted data
- partitioned 10% of data for testing
- ran 90% training data through algorithm
- compared results to test data using Root Mean Squared Logarithmic Error

The Algorithm

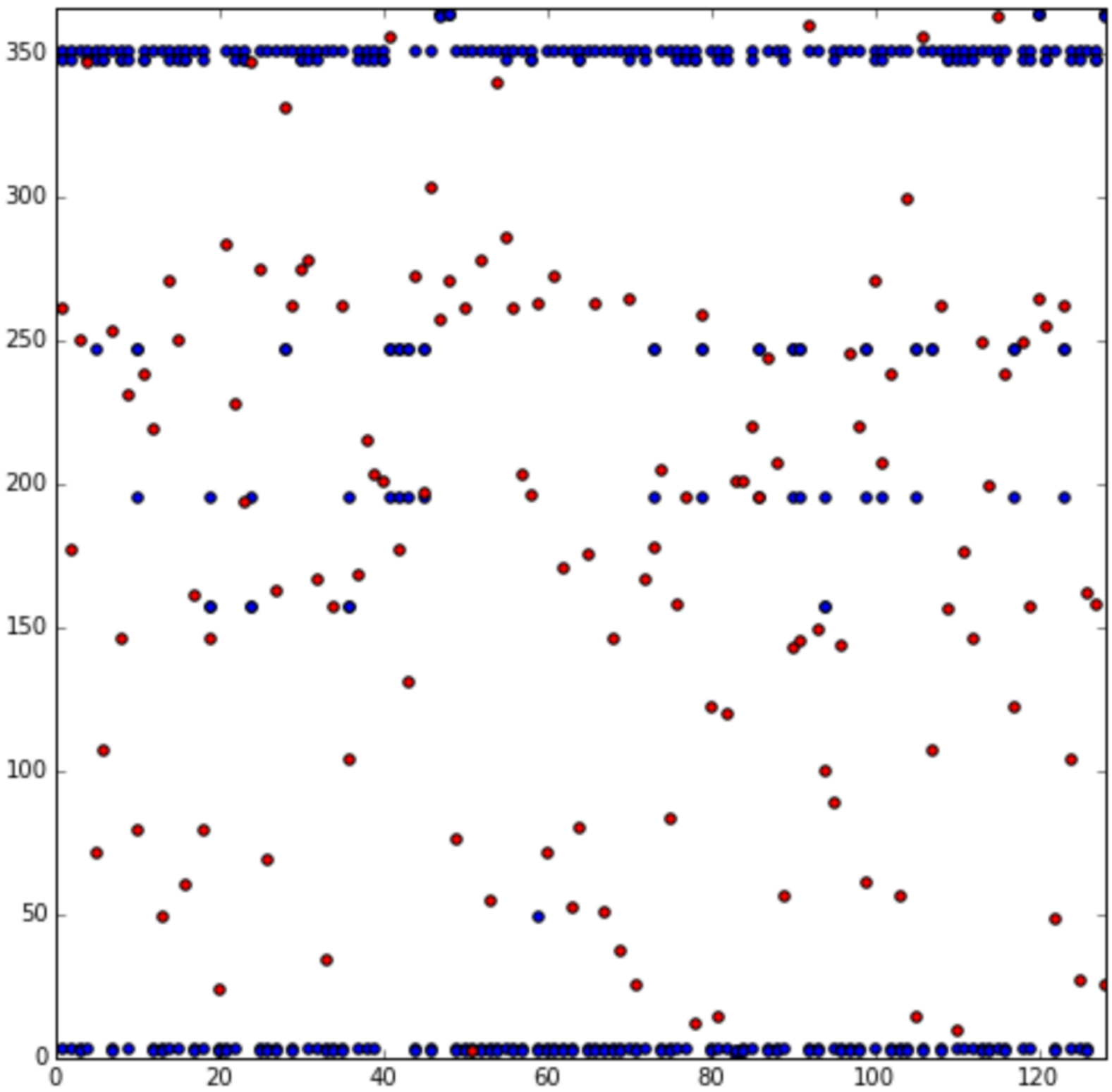
- Used date, temperature, store, and weather station as predictors

Variable Name	Description	Format
keyDict	storeKey matched to stationKey	{ 'storeKey' : [['stationKey']] }
weatherDict	Date matched with weather Data	{ 'stationKey': [['YYYY-MM-DD'...]...] }
trainDict	Train data with items and date	{ 'YYYY-MM-DD': [['store','item','units']...] }
notedDays	Hash with storekey and the noted days with weather info, 0 not noted days	{ 'storeKey' : [0,0,['YYYY-MM-DD'...]...]}
trainedPreStoreDict	StoreKey with noted days, features and items	{ 'storeKey' : [['YYYY-MM-DD','storeKey','tempurature','stationKey',i1..i111],...]
dataList	noted days array with freatures and item numbers	['DayOfYeay','storeKey','tempurature','stationKey',i1..i111]
testData	10% of the dataList	['DayOfYeay','storeKey','tempurature','stationKey',i1..i111]
trainingData	90% of the dataList	['DayOfYeay','storeKey','tempurature','stationKey',i1..i111]
tempStdev	Standard Deviation of training Data	float
simDict	score for all 90% train against each day data in 10% test data	{ IndexInTestData : [(Score, [i1...i111])...] }
predictedDict	predicted Hash with all predicted items	{ IndexInTestData : [pi1...pi111] }
RSMLEList	list of all the RSMLE across the predicted items	[RSMLE1...]

Judging Prediction Results

- Root Mean Squared Logarithmic Error
- $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i+1)-\log(a_i+1))^2}$
- n = number of rows in test set
- p = predicted units sold
- a = actual units sold

Predictions vs. Actual using Date as predictor



Predictions vs. Actual using Temperature as predictor