PREDICTING PROCEDURE TIMES

Problem

- Medical procedure scheduling is often times mostly a crapshoot
- It is very easy for procedures to get backed up, and waiting times of several hours for the patients, causing some to leave
- Doctors can also be to fast, causing on missed opportunities to schedule extra patients
- Procedures can be very expensive so scheduling well saves $

"The doctor will be with you
in just five more minutes."

Objectives

- Eventually I want to predict the total time in the room for each patient
- Total time is measured in minutes (Time Patient Exits Room –Time Patient Enters Room)
- I will measure my score in RMSE

Data Source

- I got my data, procedure information from one Spinal Intervention Specialist at Fresno Surgery Hospital for the past year

- The data is given in a clumsy text format which was a headache to clean, however eventually all data is mapped to a clean csv-file

- Much of the data had to be deleted because features were missing.
  - Only 671 Rows after this data was deleted – very small!

USER: GARBMI Procedure Times by Surgeon/Date with Patient Demographics

Date	Account Number	Age	Gender	ASA Class	BMI	Procedure Name	Into Rm	Start	End	Procedure Time	Out of Room
01/04/16	V185428	64	F		34.0	LEFT LUMBAR MEDIAL BRANCH RADIOFREQUEN..		1123		1137	

Coded Allergy:

NO KNOWN DRUG ALLERGIES

CODEINE

ASPIRIN

Uncoded Allergy:

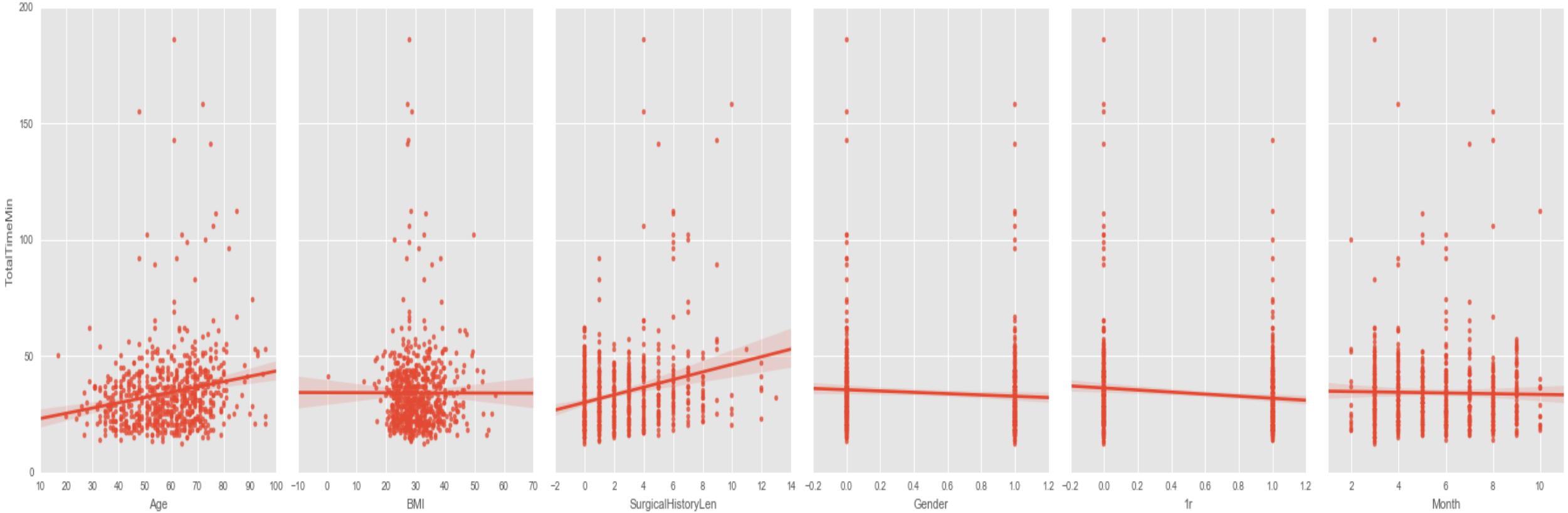
CPT Code	ICD-10 Code	Insurance
64635	M47.816	BLUE SHIELD HMO
64636	G54.4	
	M51.36	
	M51.06	
	M96.1	

Features

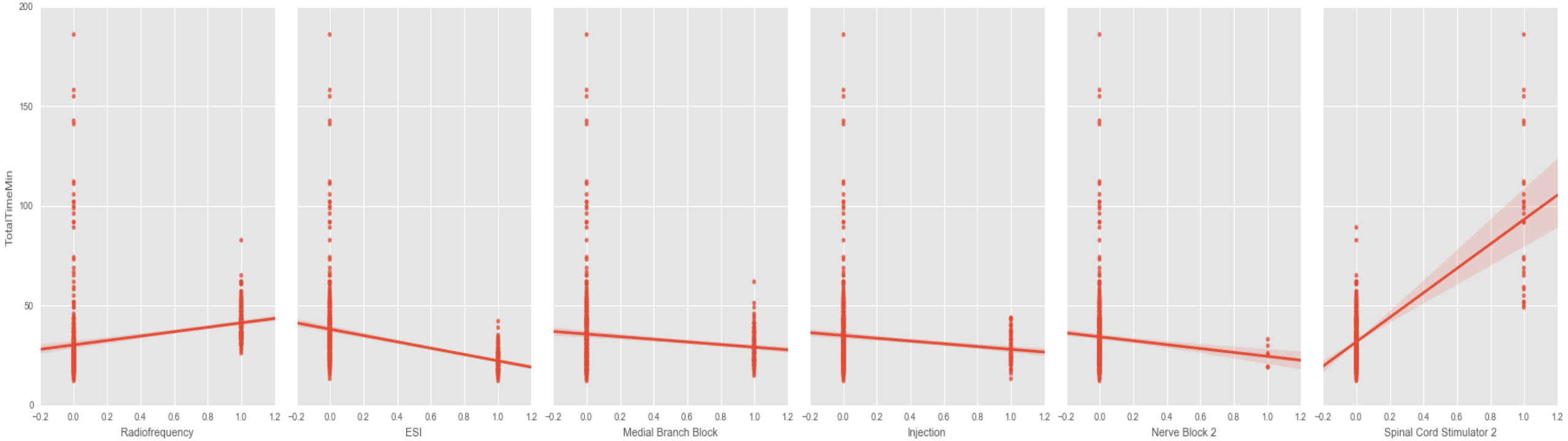
- Procedure: The type of procedure being formed, aka ‘Facet Joint Injection’
  - Two columns for procedure, a specific procedure and a more generalized feature – IE: Lumbar ESI vs ESI or Cervical Radiofrequency vs Radiofrequency
- CPT Codes: What the doctor actually bills for and gets paid for, can be multiple codes per procedure

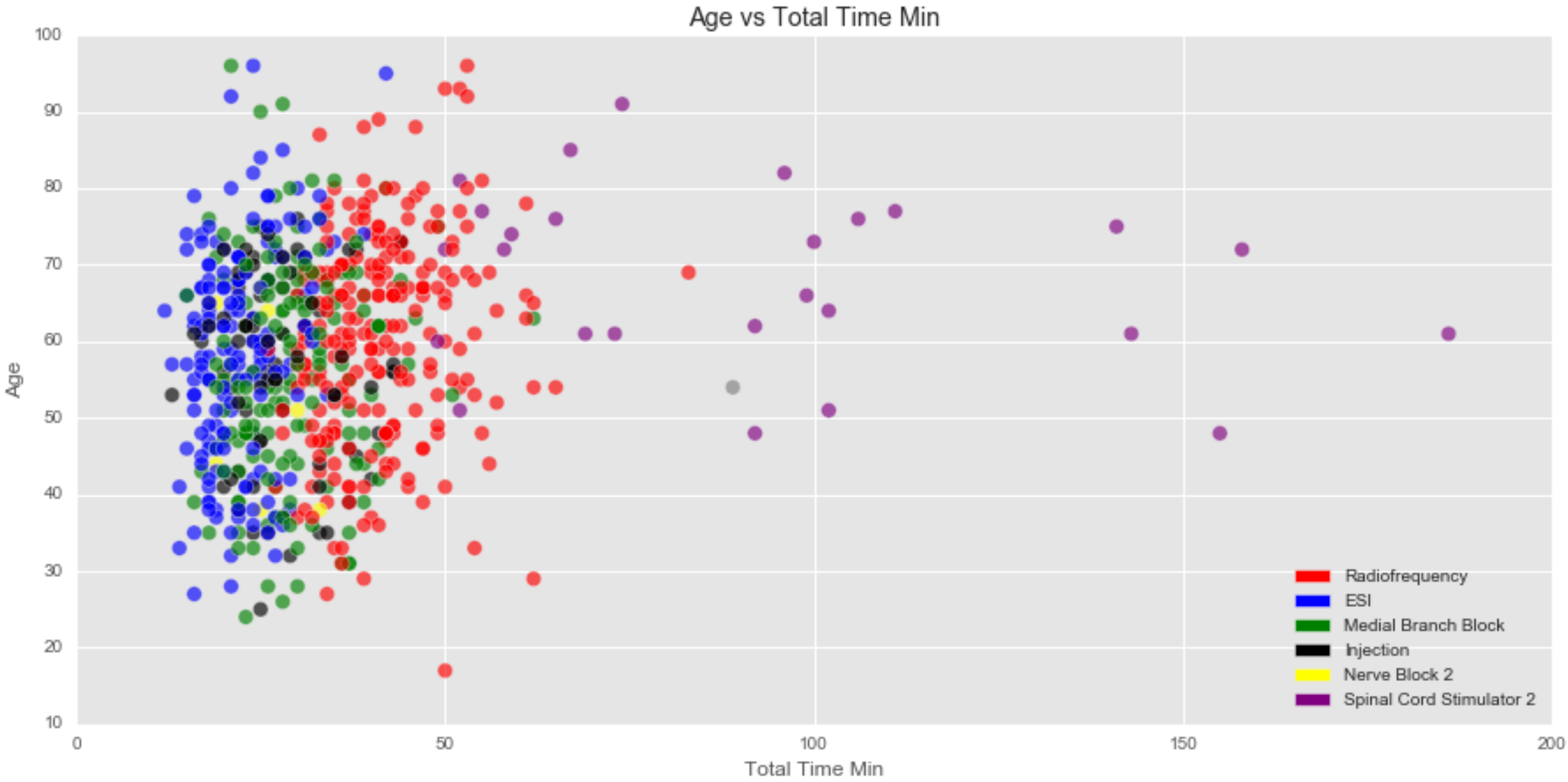
  Ex: ['64640', '63650', '64634‘]
- Demographic Data –Ex: Age, Gender etc..
- Specific procedure Data, Ex: What was the month of the procedure? Was there one or two rooms available?

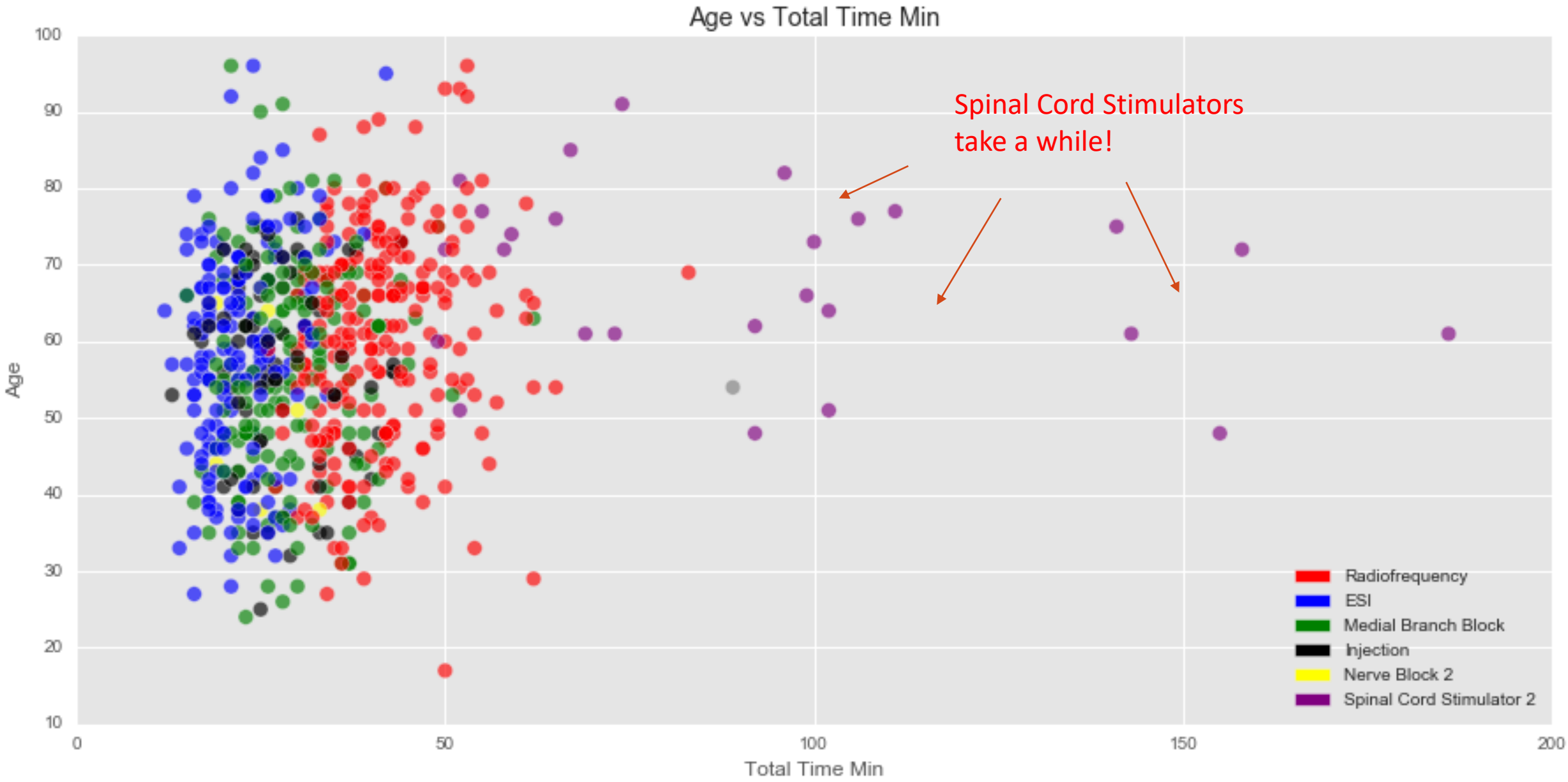
Demographic/Specific data does not seem to be very predictive:



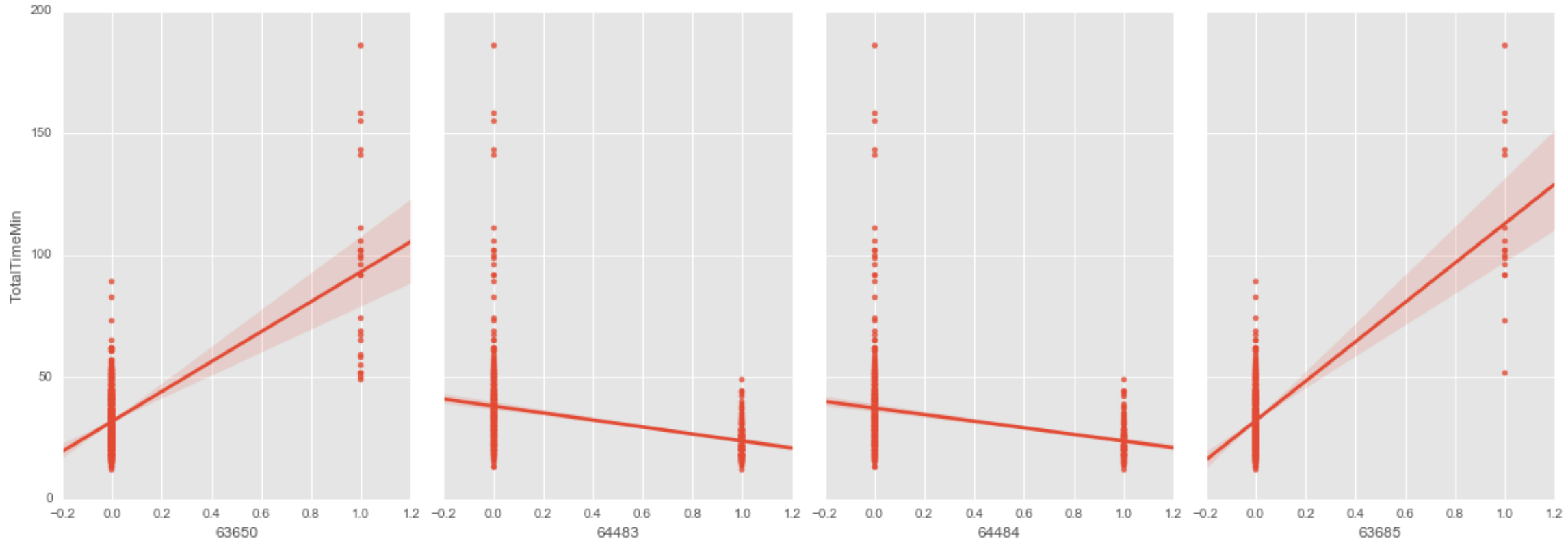
…Procedures are better




Age vs Total Time Min
Age
Total Time Min
100
90
80
70
60
50
40
30
20
10
0
50
100
150
200
Radiofrequency
ESI
Medial Branch Block
Injection
Nerve Block 2
Spinal Cord Stimulator 2


Age vs Total Time Min
Spinal Cord Stimulators
take a while!
Age
Total Time Min
100
90
80
70
60
50
40
30
20
10
0
50
100
150
200
Radiofrequency
ESI
Medial Branch Block
Injection
Nerve Block 2
Spinal Cord Stimulator 2

However … CPT codes are best



Average values and Null Accuracy

Average Time of Entire Dataset: 34.18 MIN

Null score is using the average value:

RMSE : 17.04

ABS: 11.06

Tougher Null Score using average lengths of procedures:

RMSE: 10.23

ABS: 6.62

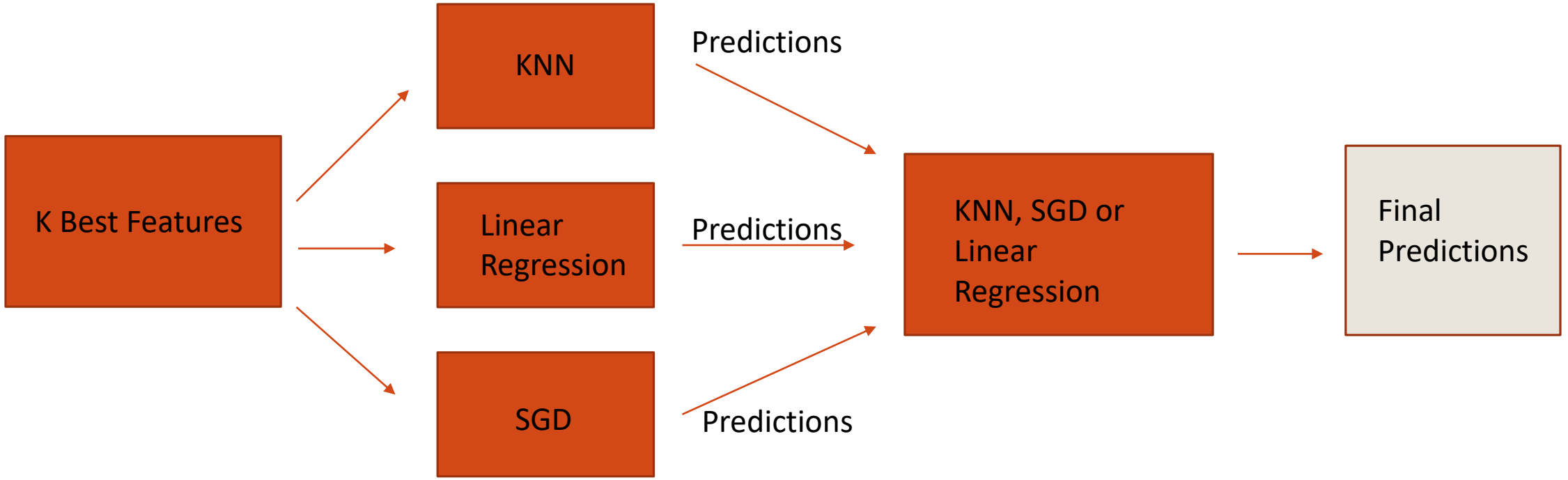
Algorithm Selection

- My most competitive algorithms were Linear Regression, or combinations of Linear Regression, SGD and KNN ensembled

- Random Forests did not work as well because of small # of samples

- To create the ensemble I used the predictions of some combination of Linear Regression, KNN and SGD as features which I fed into a another Linear Regression Algorithm

Ensemble Creation



Feature Selection

I used stepwise feature selection, selecting the ‘Kbest’ features at 1, 2,3,4 # of features… and so on

```
feature_cols = ['64640', '63650', '64634', '64480', '64635', '64636', '64484', '20605', '63685', '63688', '62311',
                '62310', '63661', '64494', '64495', '64633', '64490', '64491', '64492', '64493', '64450', '27096',
                'Gender', 'Age', 'F17', '3', '1r', 'ESI', 'Lumbar Radiofrequency', 'Nerve Block 2', 'Radiofrequency']
  X = dfNotNull[feature_cols]
  y = dfNotNull['TotalTimeMin']
  for i in range (1, len(feature_cols)+1):
     #gridsearch with select kbest
     neighbs_k = range(1, 30)
     param_grid = dict(neigh__n_neighbors=neighbs_k )
     neigh = KNeighborsRegressor()
     filter1 = SelectKBest(k=i) # select the best 2 features
     pipe = Pipeline([('anova', filter1), ('neigh', neigh)])
     grid1 = GridSearchCV(pipe, param_grid, cv=5, scoring='mean_squared_error').fit(X,y)
```

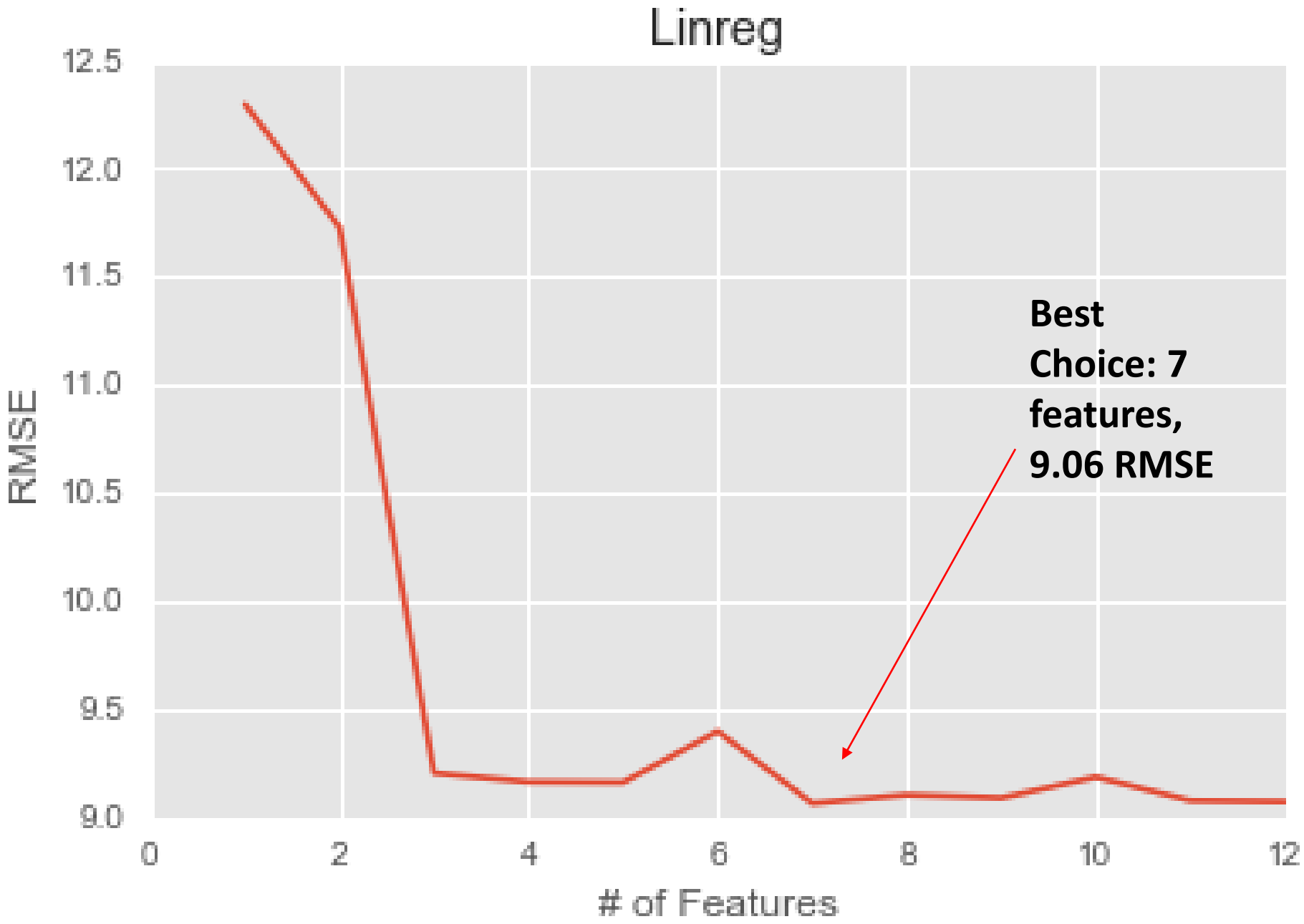
Feature Selection

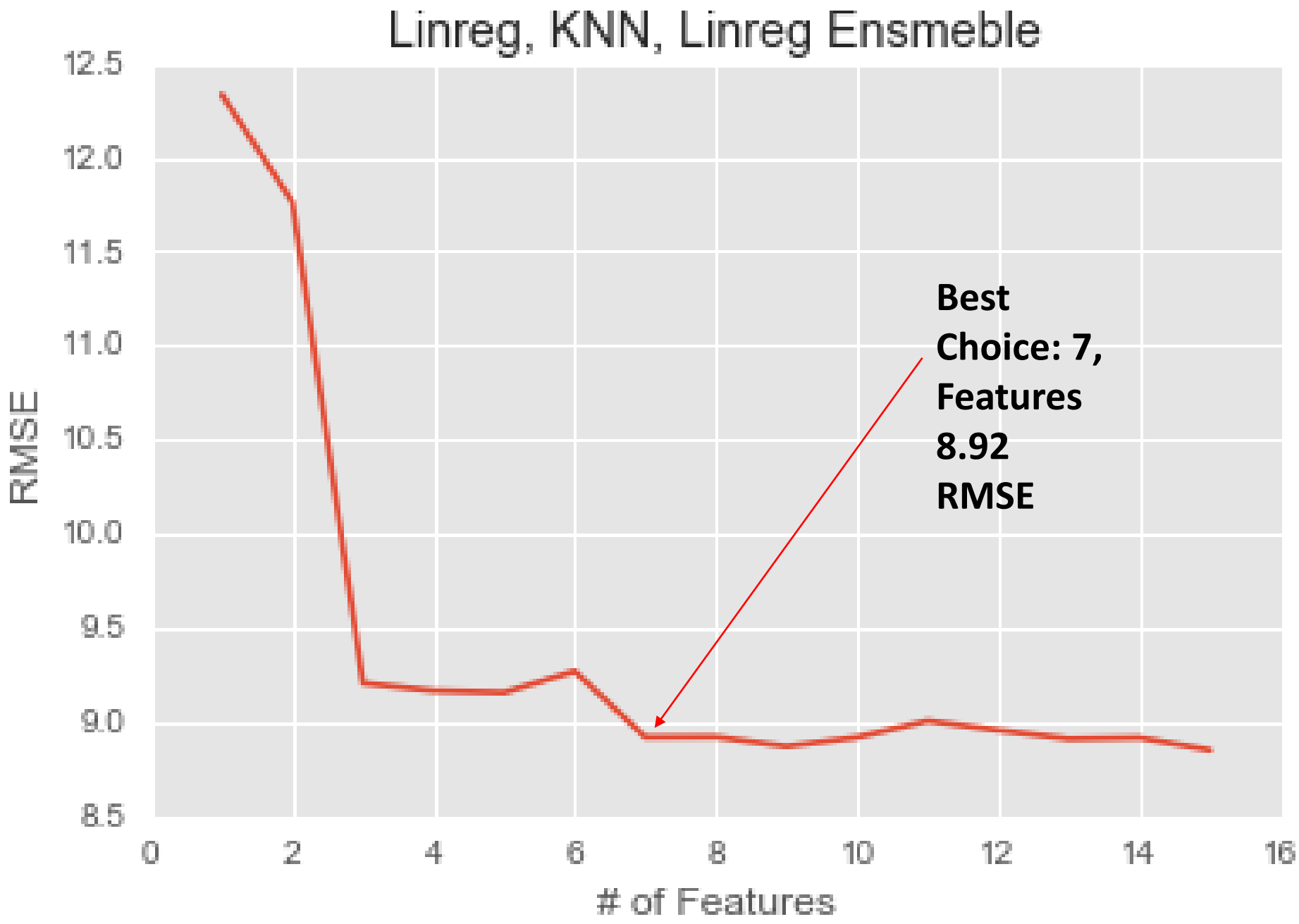
- Unfortunately, when I tried to use the features I got with the select K best feature selection again in another algorithm, I would get huge RMSE's

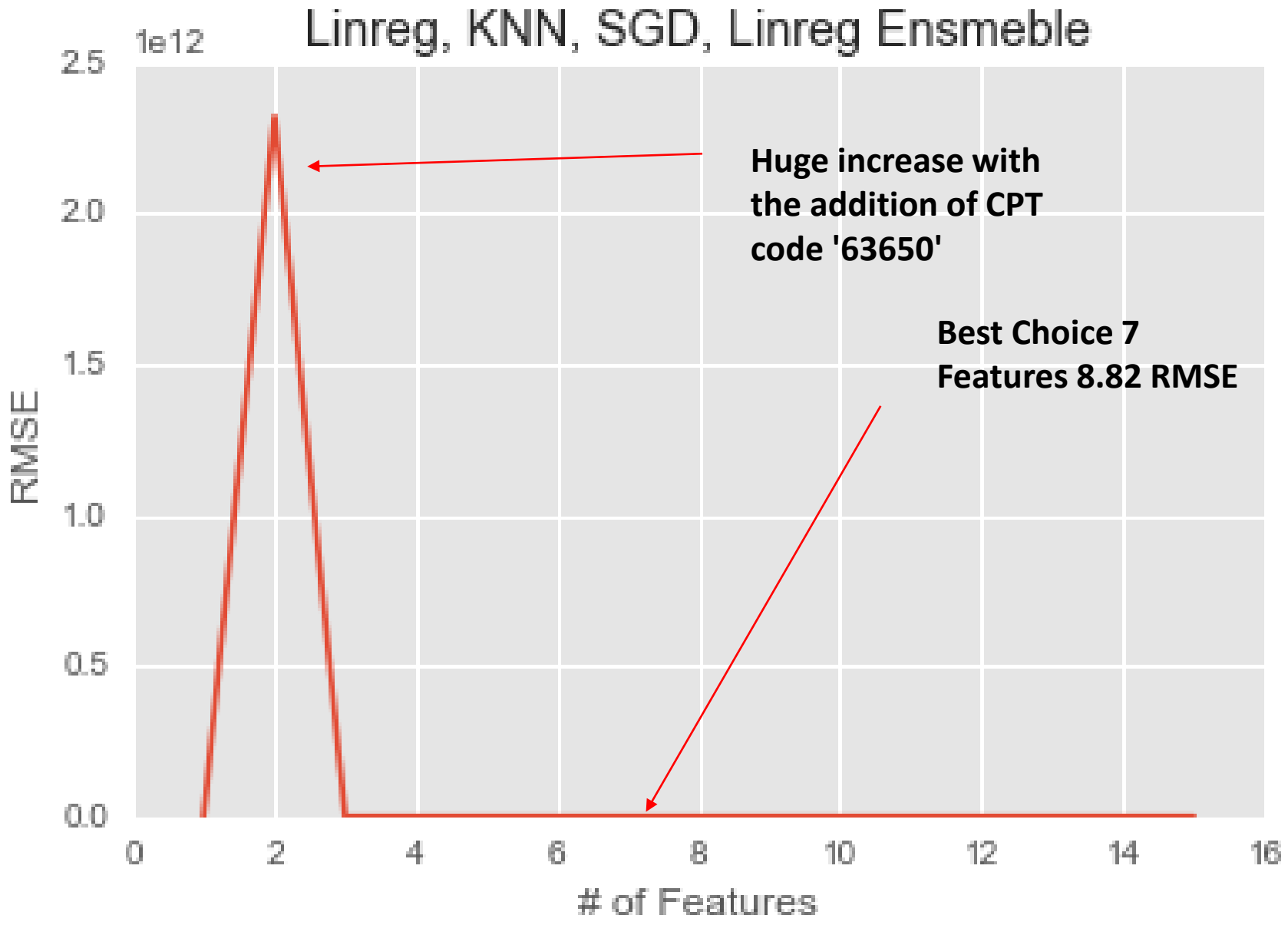
  Ex: 8.9 vs 12,0208 - Over 10,000x my first result!!!

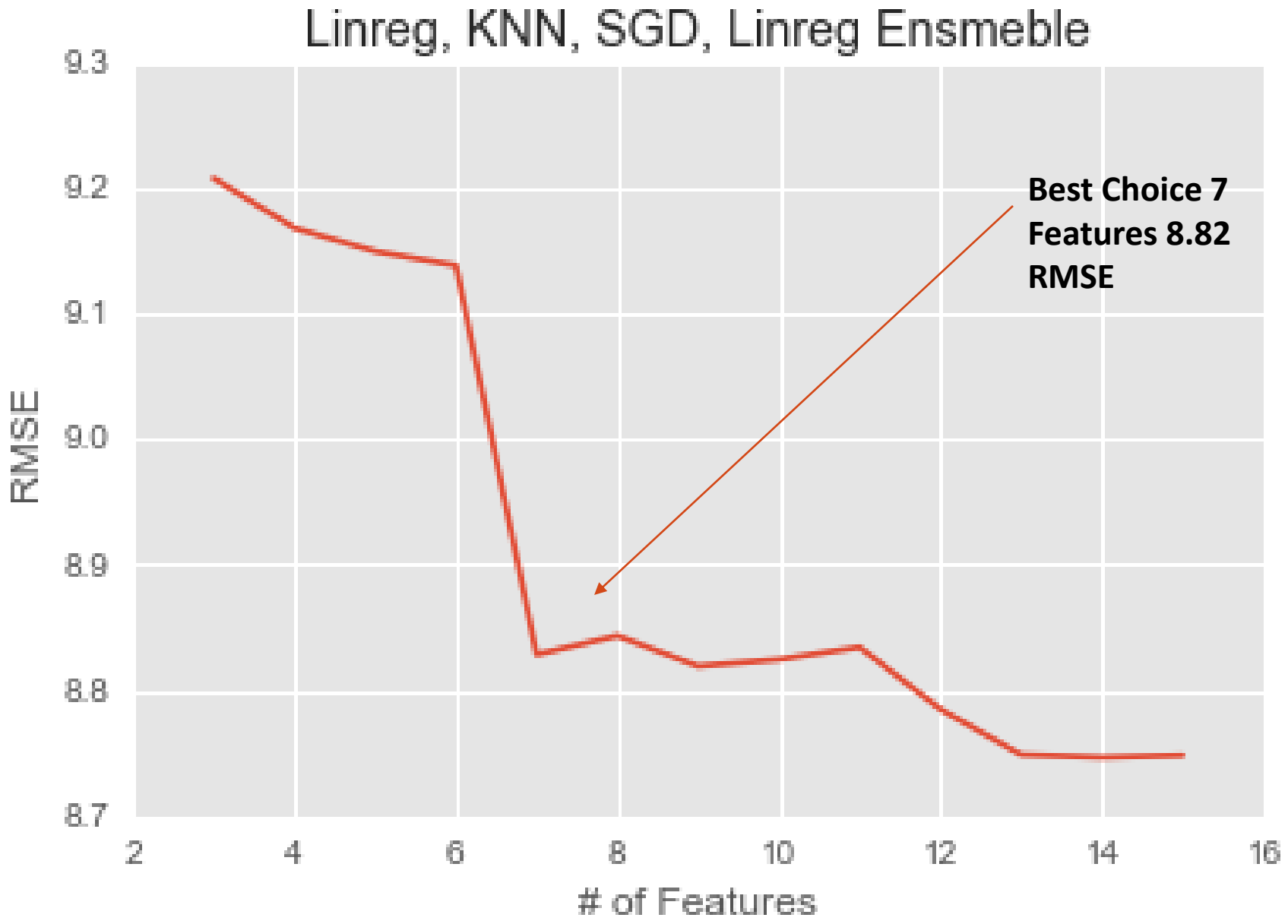
- This is most likely due to highly correlated features in the dataset, which 'Select Kbest' does not catch every time

- To combat this I added up the list of the features used most by select Kbest(Thanks Sinan!!)


Linreg
RMSE
12.5
12.0
11.5
11.0
10.5
10.0
9.5
9.0
0
2
4
6
8
10
12
# of Features
Best Choice: 7 features, 9.06 RMSE


Linreg, KNN, Linreg Ensmeble
RMSE
12.5
12.0
11.5
11.0
10.5
10.0
9.5
9.0
8.5
0
2
4
6
8
10
12
14
16
# of Features
Best Choice: 7, Features 8.92 RMSE


Linreg, KNN, SGD, Linreg Ensmeble
1e12
2.5
2.0
1.5
1.0
0.5
0.0
RMSE
0
2
4
6
8
10
12
14
16
# of Features
Huge increase with the addition of CPT code '63650'
Best Choice 7 Features 8.82 RMSE


Linreg, KNN, SGD, Linreg Ensmeble
RMSE
9.3
9.2
9.1
9.0
8.9
8.8
8.7
2
4
6
8
10
12
14
16
# of Features
Best Choice 7
Features 8.82
RMSE

Final Features

- I chose the Linreg, KNN, SGD Ensemble method as it had the lowest value at 7 features

- The final features I found are: ['63685', '63650', 'Radiofrequency', 'Lumbar Radiofrequency', '64636', '64635', 'ESI‘, ‘1r’]

- All CPT-Codes or procedure names except ‘1r’ which tells whether there is one or two rooms on that day, which I was told by the doctor was important.

Final Scores

- My final RMSE is 8.8 –Beats null by 48%, Beats tougher null by 14%

- My final Absolute Error Score is 6.13

Next Steps

- Try to find exactly why I am getting such large RMSE values
- Try feature clustering to see if that is helpful
- Create a website with Heroku where the doctor can get predictions for each procedure
- See if I can apply ordinal classification