Human Activity recognition using Smart Phone Dataset

- Sravan Phani Kumar.V

26.09.2013

Problem Statement

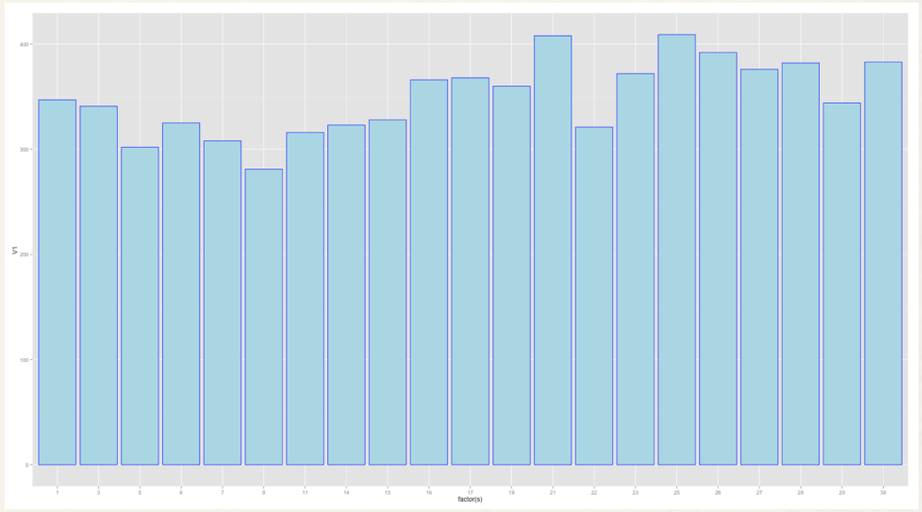
- Given a dataset of features captured from a Smart-phone, we should be able to predict the activity being performed by the user.
- This would have many potential applications with respect to wearable computing devices that are set to come into the market.
-

Dataset Analysis

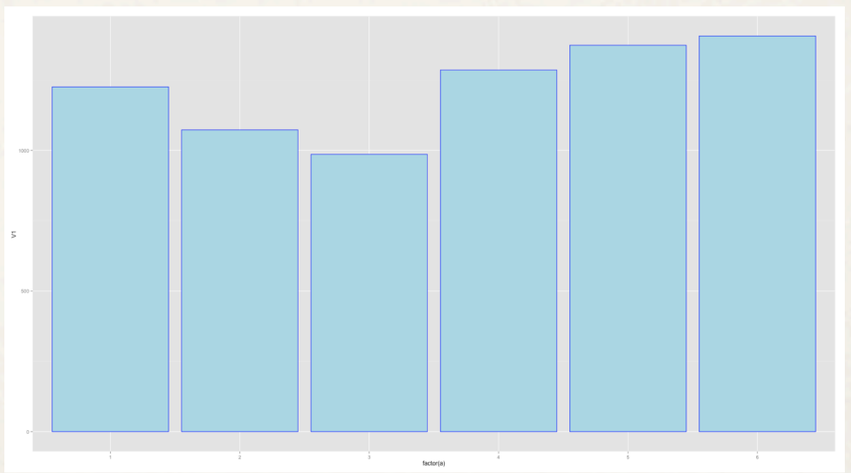
- As part of data exploration, the following steps were performed:
  - Calculated the summary statistics
  - Data skewness by subjects and activities
  - Within Group variances and Between Group variances for subjects and activities.
  - Correlation between variables.

Dataset Analysis

- # of records per subject & activities
  - not all subjects were present in the training set.



- data skewness was not largely present.



Dataset Analysis

- Between group variance/Within group variance
- For subjects -

V475	0.0431655	4.35894	100.98208
V56	0.126435	12.76346	100.94879
V65	0.1256107	11.73026	93.385827
V483	0.0320361	2.8614093	89.318238
V487	0.0388846	3.1720263	81.575489
V491	0.0168369	1.3602663	80.790807
V477	0.0127634	1.0303696	80.728199
V484	0.0154966	1.2486935	80.578289
V441	0.0349577	2.7551562	78.814003
V138	0.0345248	2.6858253	77.794176
V498	0.0227891	1.7159334	75.296234
V331	0.0458297	3.263579	71.211014
V178	0.0175861	1.2300001	69.941502
V410	0.0261765	1.7979842	68.686973
V339	0.0443903	2.9120834	65.601782
V249	0.0789145	5.162685	65.421217
V343	0.0492794	2.9986463	60.84987
V548	0.0165654	0.9970566	60.188967
V265	0.0518709	3.0723057	59.22985
V284	0.0512521	3.0288934	59.097984

For activities -

V367	0.02164292	7.99E+02	3.69E+04
V41	0.01273386	3.74E+02	2.94E+04
V53	0.01283156	3.62E+02	2.82E+04
V368	0.02695913	7.58E+02	2.81E+04
V50	0.01358531	3.63E+02	2.67E+04
V57	0.02608634	6.78E+02	2.60E+04
V524	0.02631461	6.22E+02	2.36E+04
V235	0.03122136	7.32E+02	2.34E+04
V288	0.03130352	7.27E+02	2.32E+04
V103	0.0264323	5.89E+02	2.23E+04
V10	0.0200482	4.07E+02	2.03E+04
V369	0.02987958	5.66E+02	1.90E+04
V105	0.02752299	5.14E+02	1.87E+04
V104	0.03020173	5.53E+02	1.83E+04
V272	0.01613208	2.95E+02	1.83E+04
V4	0.01504081	2.74E+02	1.82E+04
V559	0.01966967	3.56E+02	1.81E+04
V281	0.01778969	3.13E+02	1.76E+04
V16	0.01801731	3.01E+02	1.67E+04
V269	0.01696772	2.83E+02	1.67E+04

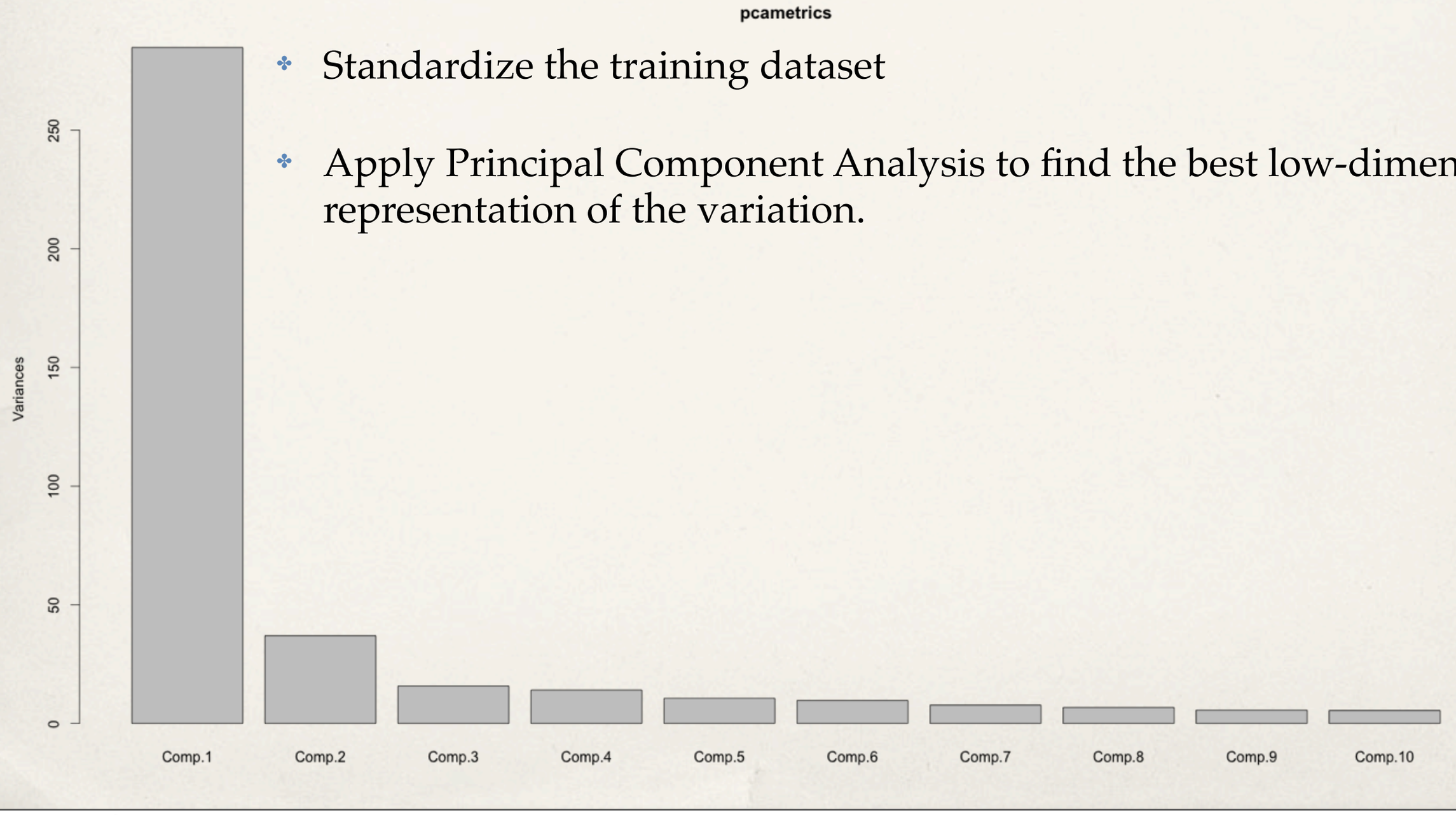
Dataset Analysis

- Correlation table:
  - When generating the correlation table it was observed that for most of the variables were showing to be perfectly correlated.
  - These could be possible duplicate columns.
  - Shown beside are the top 30 correlated attributes and their correlation

V201	V206	1
V201	V214	1
V206	V214	1
V202	V215	1
V203	V216	1
V204	V217	1
V205	V218	1
V201	V219	1
V206	V219	1
V214	V219	1
V207	V220	1
V208	V221	1
V209	V222	1
V210	V223	1
V211	V224	1
V212	V225	1
V213	V226	1
V227	V232	1
V240	V245	1
V253	V258	1
V503	V508	1
V516	V521	1
V529	V534	1
V542	V547	1
V99	V363	0.9999998
V98	V362	0.9999997
V97	V361	0.9999994
V282	V315	0.9998784
V440	V473	0.9997666
V283	V329	0.9996611

Dimensionality Reduction

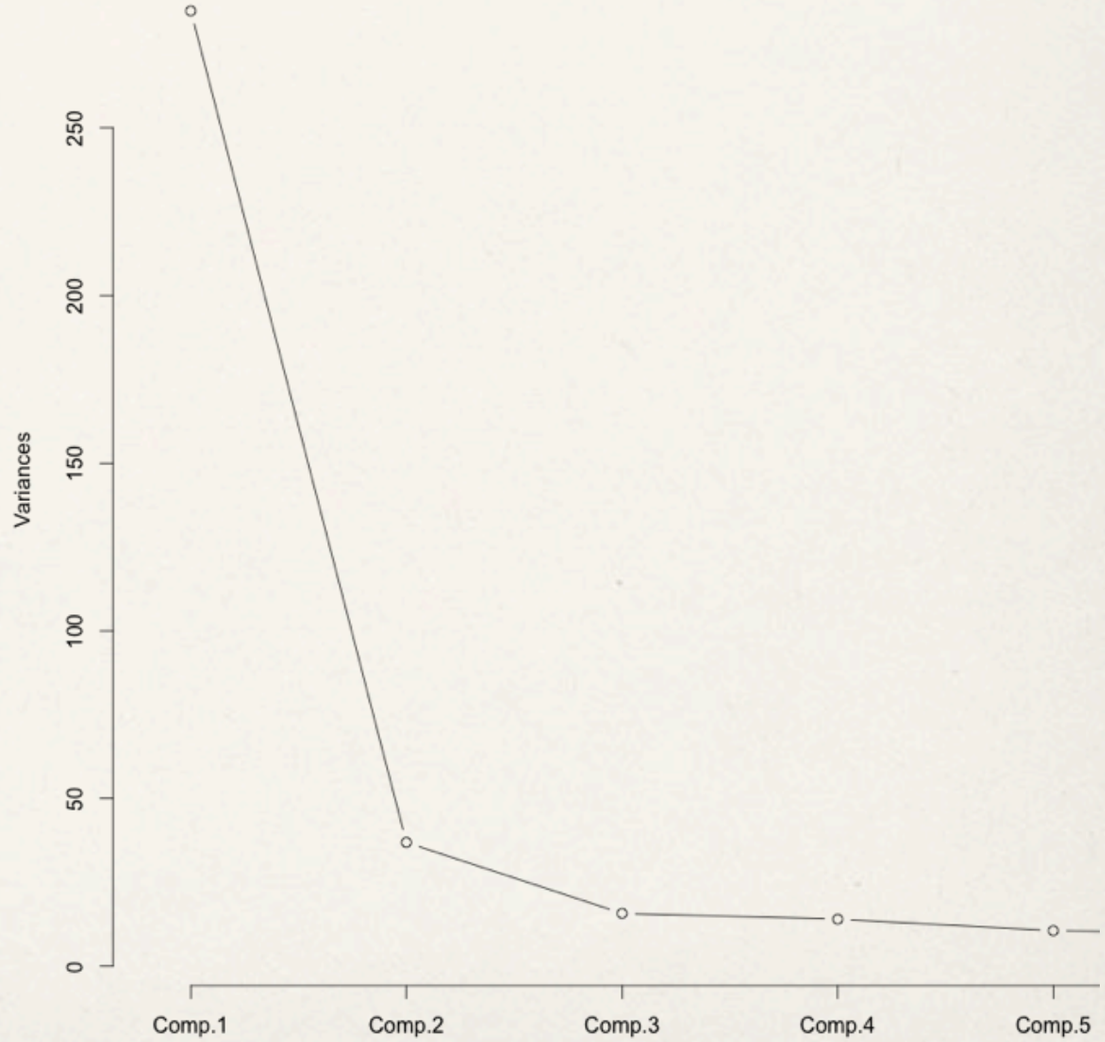
- Principal Component Analysis:



  - Standardize the training dataset
  - Apply Principal Component Analysis to find the best low-dimensional representation of the variation.

PCA Cntd...

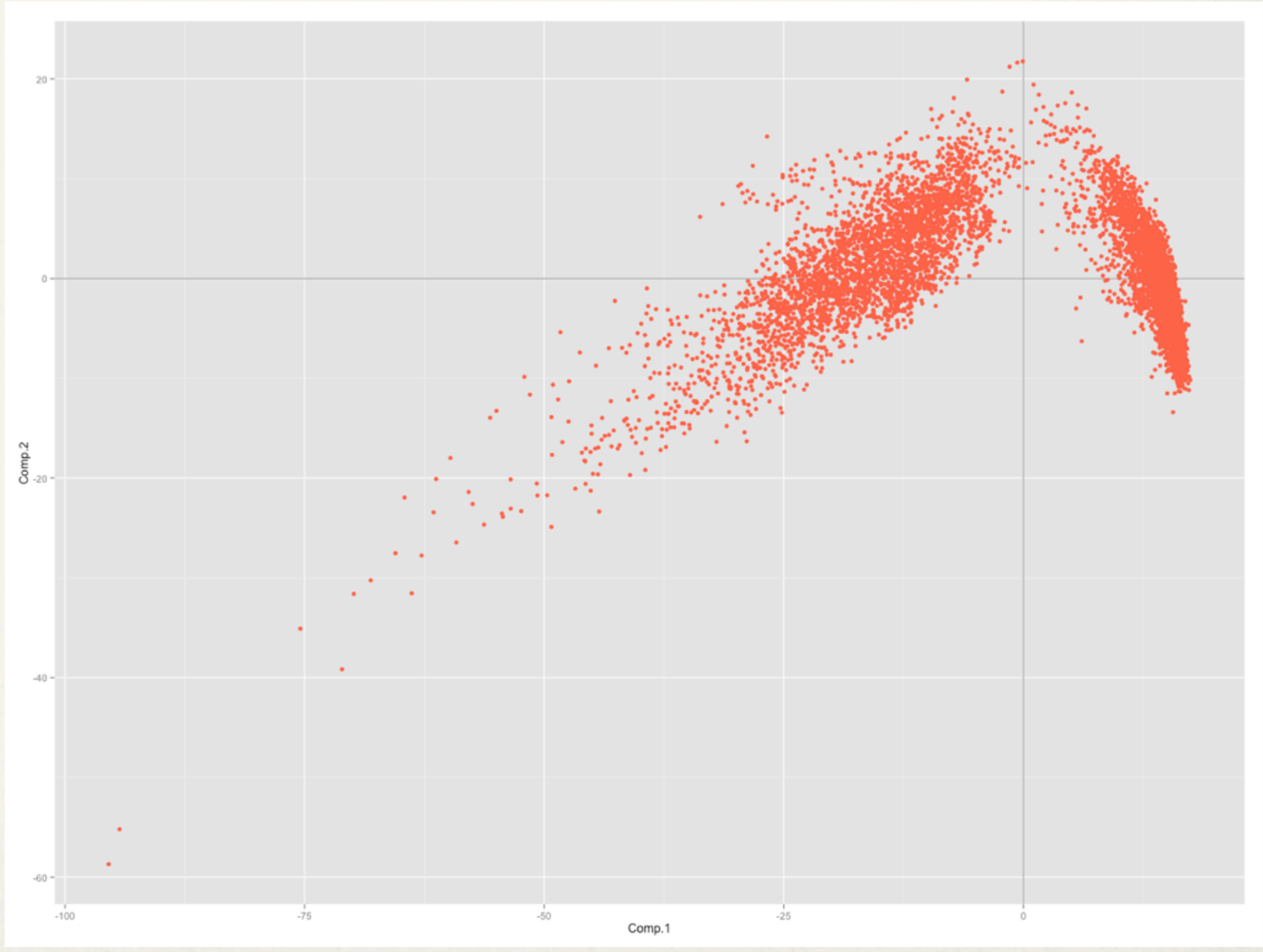
- After applying PCA there are three ways of deciding the number of principal components that are to be retained:
  - Based on the screeplot.
  - Retain components based on Kaiser's criterion wherein you keep all the components whose proportion of variance is greater than 1.
  - Pick the components such that the cumulative percentage of variance explained is closer to your desired percentage.



PCA Cntd...

- Post dimensionality reduction, the number of principal components retained were 61 and the cumulative percentage of variance explained by them was ~ 90%

- This was done so as to avoid overfitting the model by including many more principal components.



Oblique Decision Trees

- A new dataset consisting of the principal components was created and the same was fed to an Oblique Decision Tree model for training.
- Initially, we have considered only oblique splits as it is faster compared to doing both oblique and axis parallel splits.
- The model summary post training is as shown below:
  - *oblique.tree(formula = V1 ~ ., data = predset, oblique.splits = "only")*
  - *Number of terminal nodes: 17*
  - *Residual mean deviance: 0.1614 = 1184 / 7335*
  - *Misclassification error rate: 0 = 0 / 7352*

Model Summary

- The confusion matrix for the Oblique Tree training model is as shown.

	1	2	3	4	5	6
1	1226					
2		1073				
3			986			
4				1199	87	
5				103	1271	
6						1407

- The accuracy of the model was calculated as *97.41%*

- The confusion matrix for the Oblique Tree model when applied to hold-out dataset is as shown.

	1	2	3	4	5	6
1	444	22	30			
2	21	409	41			
3	14	19	387			
4				448	42	1
5				137	386	9
6				2		535

- The accuracy of the model on the hold-out dataset was calculated as *88.53%*

Thank You