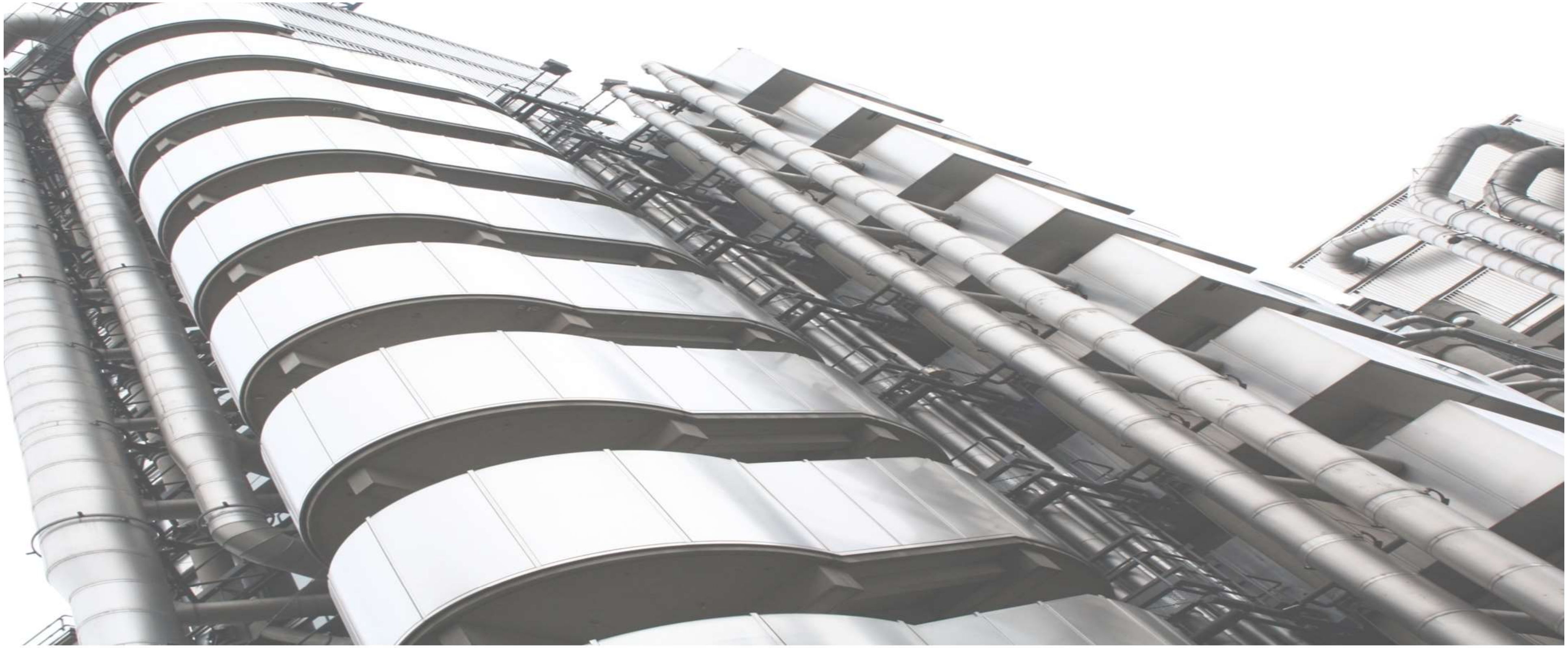


Feature Synthesis Using t-SNE and Clustering

Mark Lee

Insight Risk Consulting

The question

- Powerful supervised learning algorithms can improve the predictive power of pricing models, but predictive power is not all:
  - Implementation issues from legacy systems.
  - Difficulties with transparency – may not easily be explained or adjusted.
  - Convincing stakeholders to move from familiar models.
- How can I use newer machine learning methods in pricing, while avoiding these issues?
  - Work with legacy systems
  - “Transparent” final model

A solution

- Rather than looking at the latest and greatest supervised learning algorithm – try to use unsupervised algorithms to enhance existing model.
- Here I use t-distributed Stochastic Neighbour Embedding (t-SNE) and hierarchical clustering.
- Applied to real data – here conversion data for a personal lines motor insurance – looking for features which were not adequately modelled in the pricing GLM.

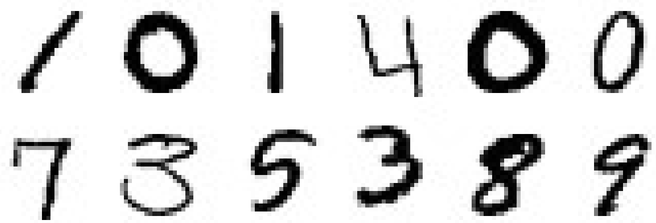
t-SNE in a nutshell

- A dimensionality reduction technique.
  - Measures the similarity between data points in high dimensional space.
  - Build a map in low dimensional space (typically 2D or 3D) such that points that were similar are close together.
  - Tries to preserve local similarities, at the cost of large scale similarities.
- L.J.P. van der Maaten and G.E. Hinton, *Journal of Machine Learning Research* 9, 2579 (2008).

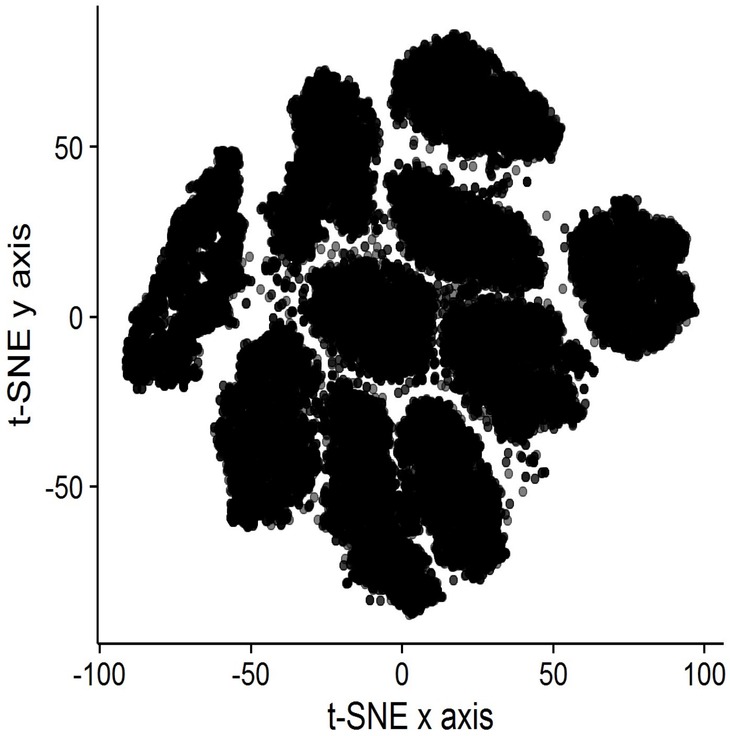
t-SNE example - MNIST

Insight Risk Consulting

- MNIST - handwritten digits. 28x28 pixels = 784 dimensional space.

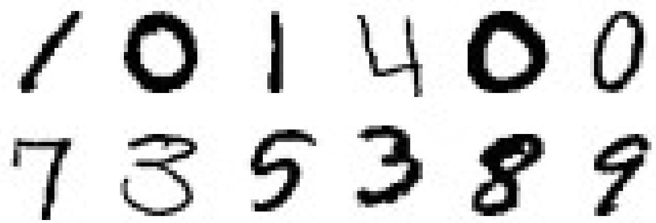


- t- SNE 2D:

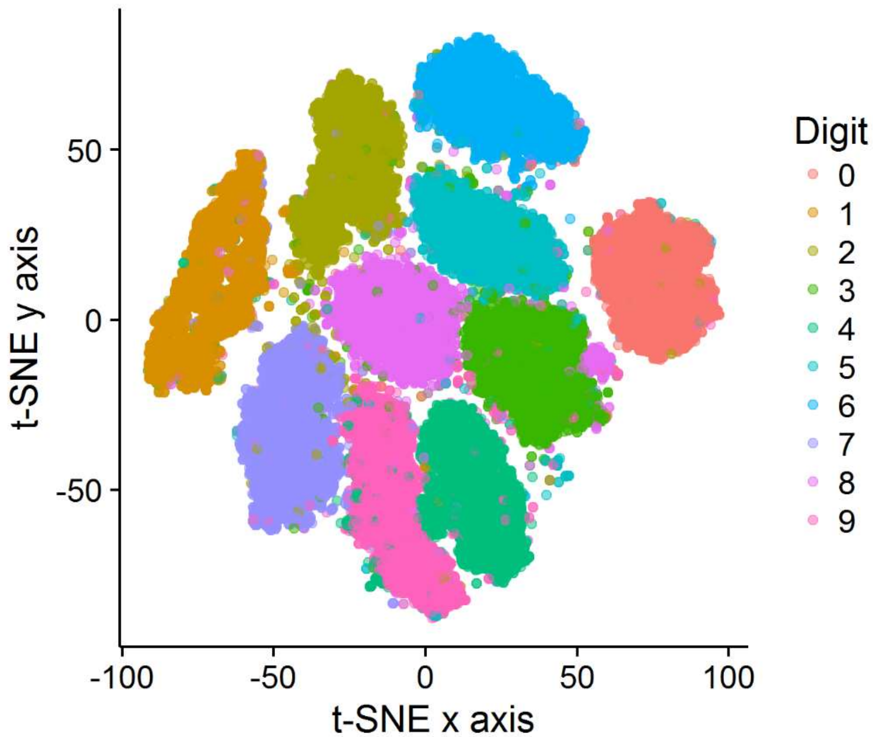


t-SNE example - MNIST

- MNIST - handwritten digits. 28x28 pixels = 784 dimensional space.



- t- SNE 2D:



Using t-SNE

- Tune the hyperparameters – particularly the “perplexity” and whether the algorithm has converged (number of iterations and learning rate).
- Cluster sizes are normally not meaningful.
- Distances between clusters might not be meaningful.
- In general, look at results with different perplexities to ensure you are not just looking at noise.
- See Wattenberg, et al., "How to Use t-SNE Effectively", Distill, 2016. http://doi.org/10.23915/distill.00002

Conversion analysis

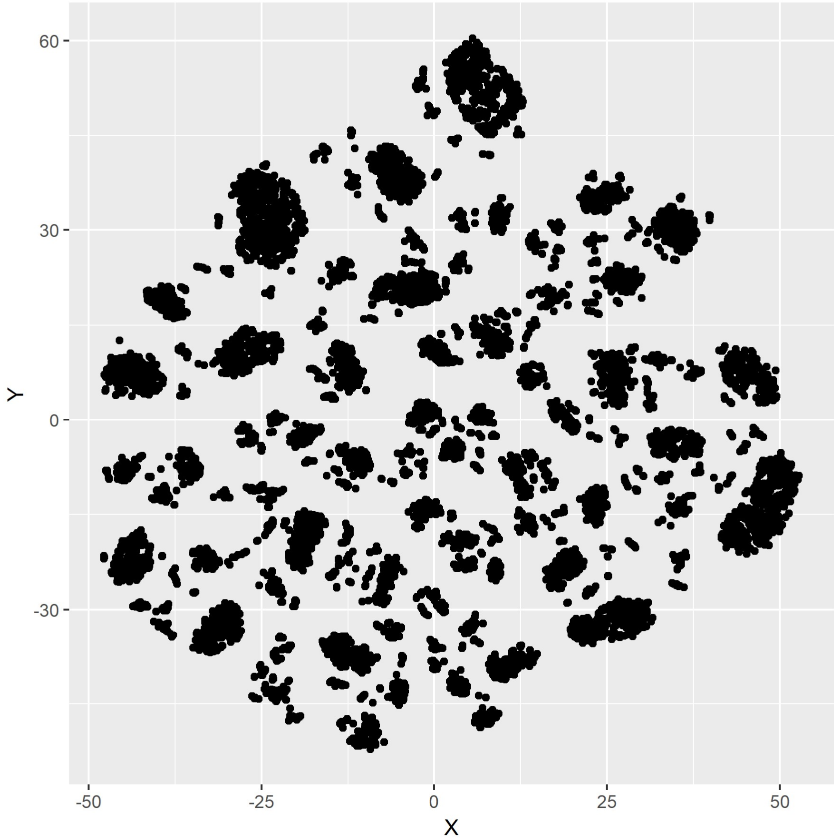
- Apply to (anonymised, adjusted) conversion data - take up of personal lines motor insurance quote (similar analysis applies to severity/freq modelling).

WRISK Things change. Embrace Wrisk.

- We use 16 of the most important exposure variables – some of these are categorical – 29 dimensional space.
- Need a similarity measure for mixed variable types - use Gower distance:
  - standardises numerical variables
  - categorical variables – 1 if identical, 0 otherwise
  - binary variables - uses Dice coefficient
  - maps distances so that measure is always between 0 and 1 for each variable.

t-SNE for conversion data

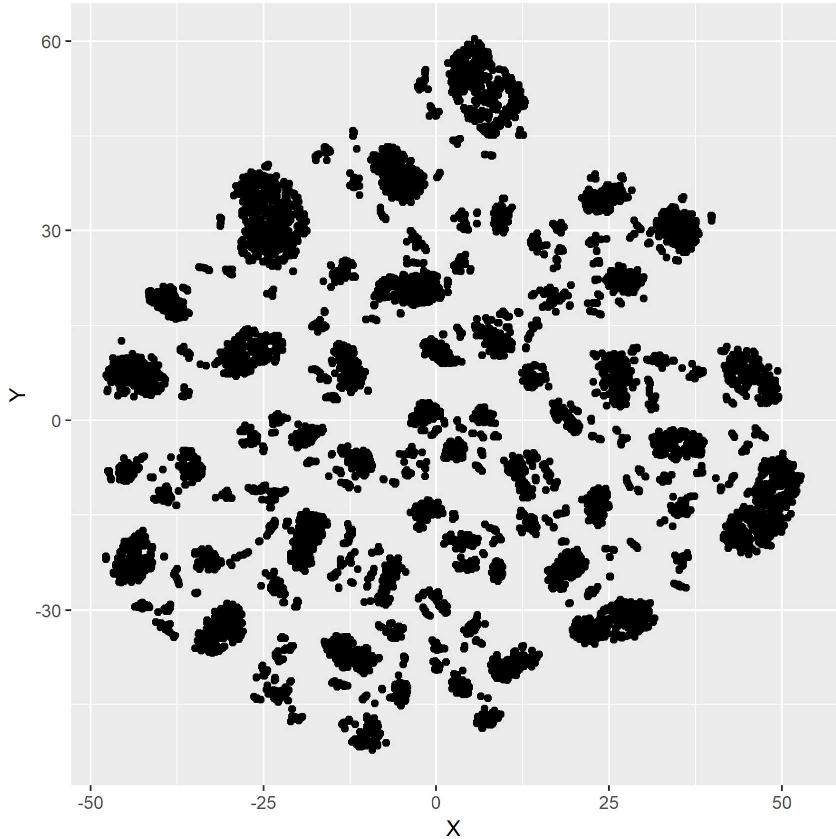
- 2D t-SNE - there seem to be some groups:




```
R:
> library(cluster)
> gower_dist = daisy(df,
        metric = "gower")
> library(Rtsne)
> tsne = Rtsne(gower_dist,
        is_distance = TRUE,
        dims = 2,
        perplexity=50)
> tsne_data = tsne$Y %>%
    data.frame() %>%
    setNames(c("X", "Y"))
> library(ggplot)
> ggplot(aes(x = X, y = Y),
         data = tsne_data)
        + geom_point()
```

t-SNE for conversion data

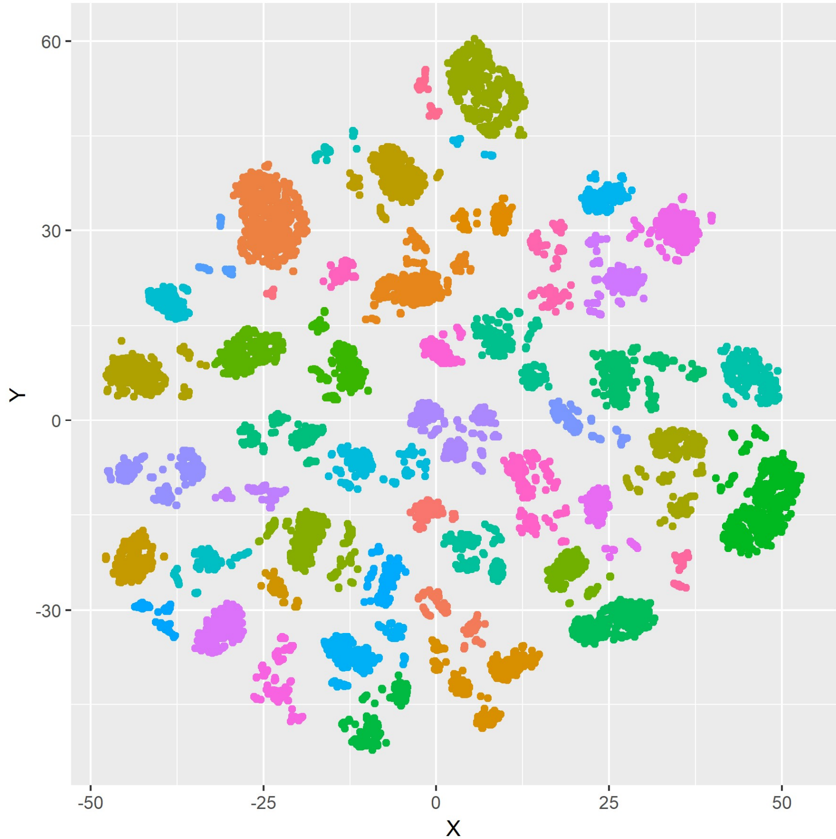
- Group using hierarchical clustering:




```
> cluster_model = hclust(dist(tsne_data%>%select(X,Y)),
        method = "average")
> df$cluster = cutree(cluster_model,50)
```

t-SNE for conversion data

- Group using hierarchical clustering:

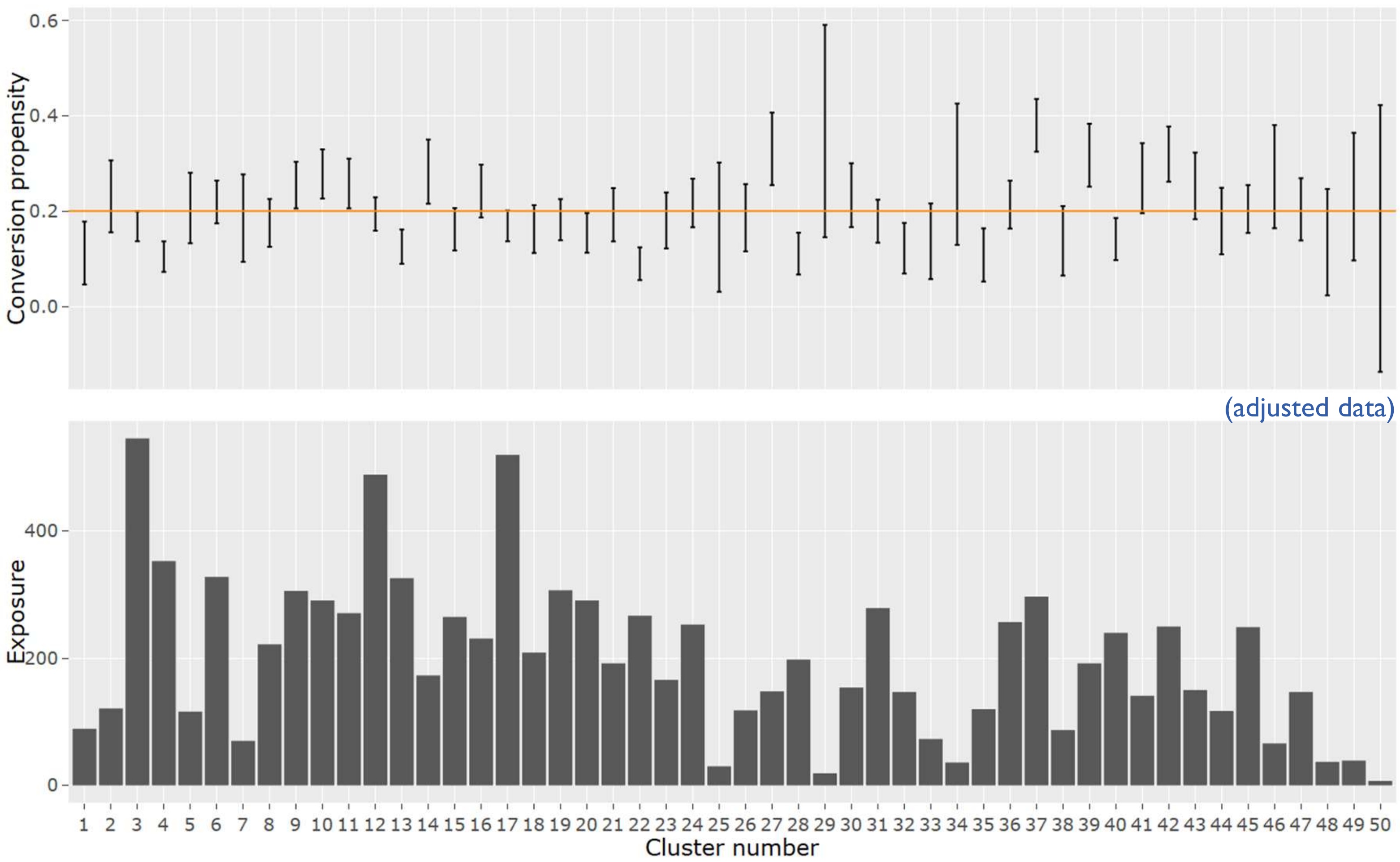



```
> cluster_model = hclust(dist(tsne_data%>%select(X,Y)),
        method = "average")
> df$cluster = cutree(cluster_model,50)
```

Are those clusters predictive?

Insight Risk Consulting

- Average conversion by cluster:



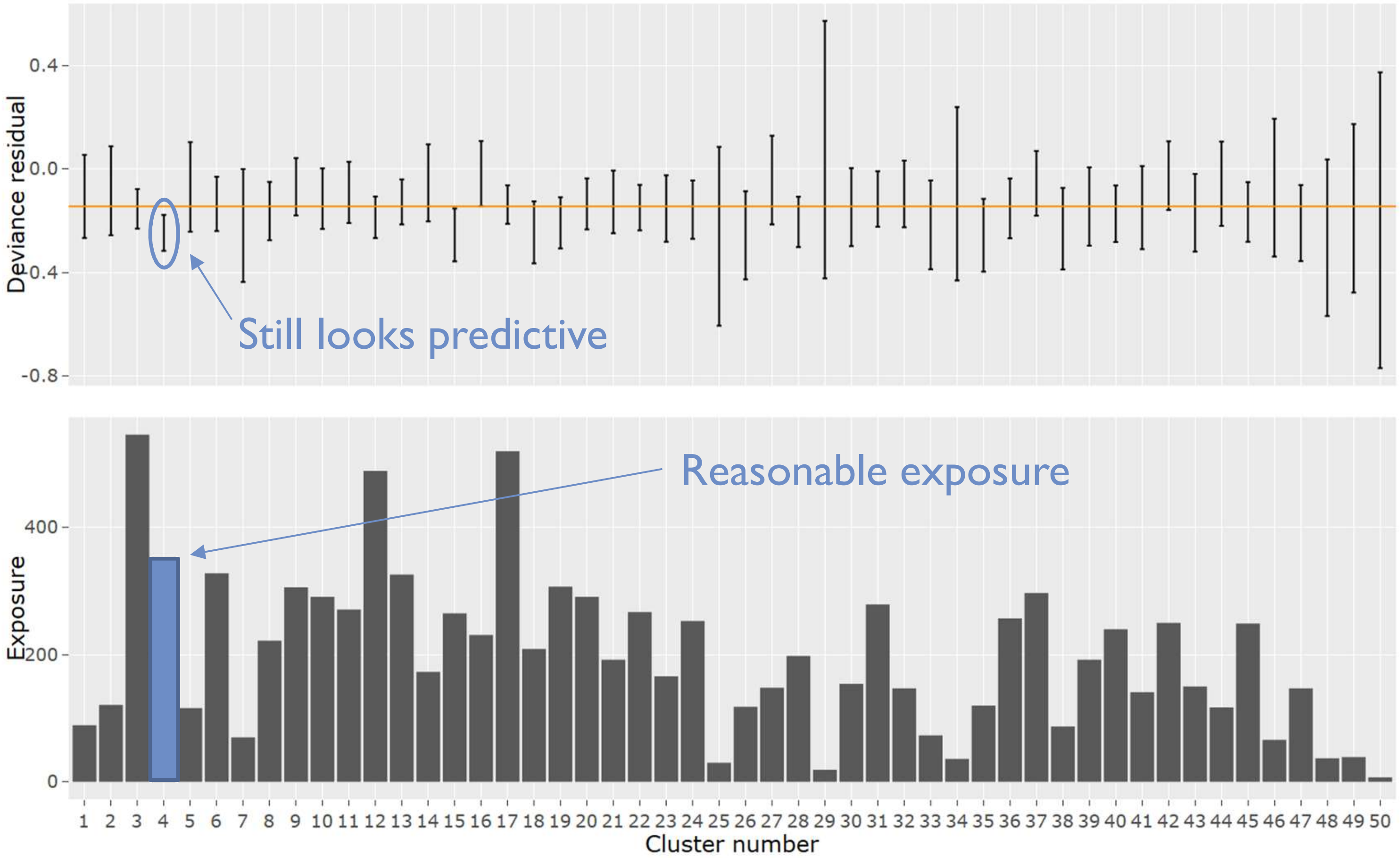
Are the clusters already modelled?

- The clusters by themselves seem predictive - BUT:
- Much of the explanation of the clusters different conversion might already be accounted for in your model structure - e.g. might just be due to the Age curve.
- To check, use logistic regression with the same model structure, rating factors etc as used to generate the quote premium.

Fit logistic model for conversion → Calculate residuals from model → Assess residuals by cluster

Residuals from logistic regression

- Most dependence with cluster disappears when assessed against residuals of logistic model:

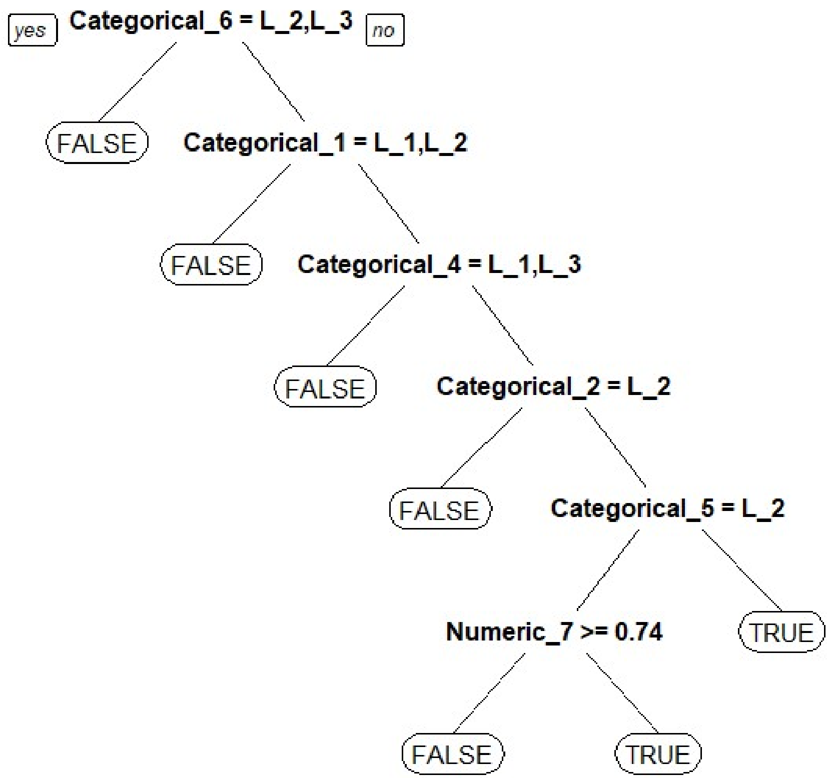


Cluster 4

Insight Risk Consulting

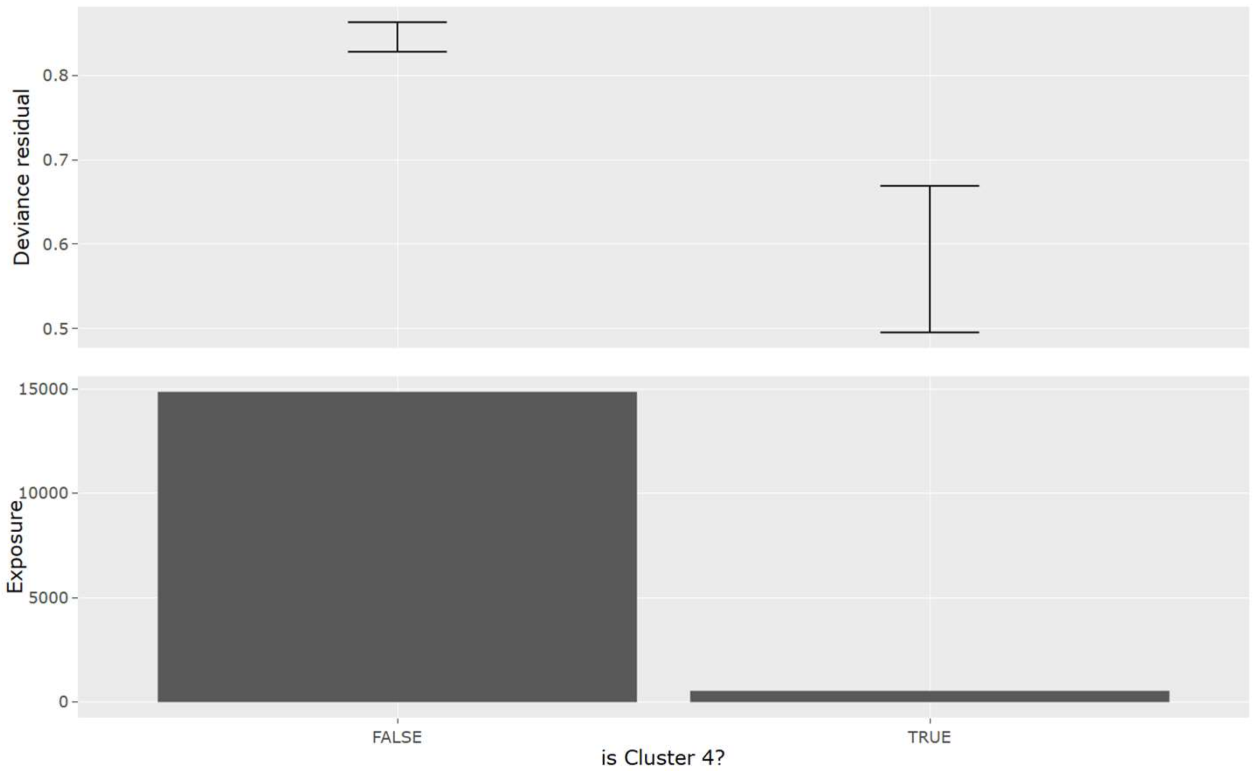
- An explanation of "it's in cluster 4" is not transparent!
- Understand what makes up cluster 4:
  - e.g. CART tree model
  - So here explained by 5 categorical variables – looking at data volumes, can whittle down to mostly a 4 way interaction on vehicle attributes and vehicle usage.
- Now we have a variable which we can take to the underwriter.

```
> library(rpart)
> rpart(isCluster4~.,data = df%>%
        mutate(isCluster4 =
        as.factor(cluster==4))
```



Test performance

- Test on held out data:
  - Classify as “cluster 4” or not based on interaction rule.
  - Assess residuals from logistic model of conversion against this classification.



Conclusions

- Feature synthesis – a new predictive variable/interaction was found, that could be relatively easily communicated, and implemented in a traditional rating system.
- Found using a combination of (mostly) unsupervised learning methods:
  - t-SNE
  - hierarchical clustering
  - logistic regression modelling
  - CART models
- The same procedure *can* work on any predictive data – claims freq., claim severity, etc.