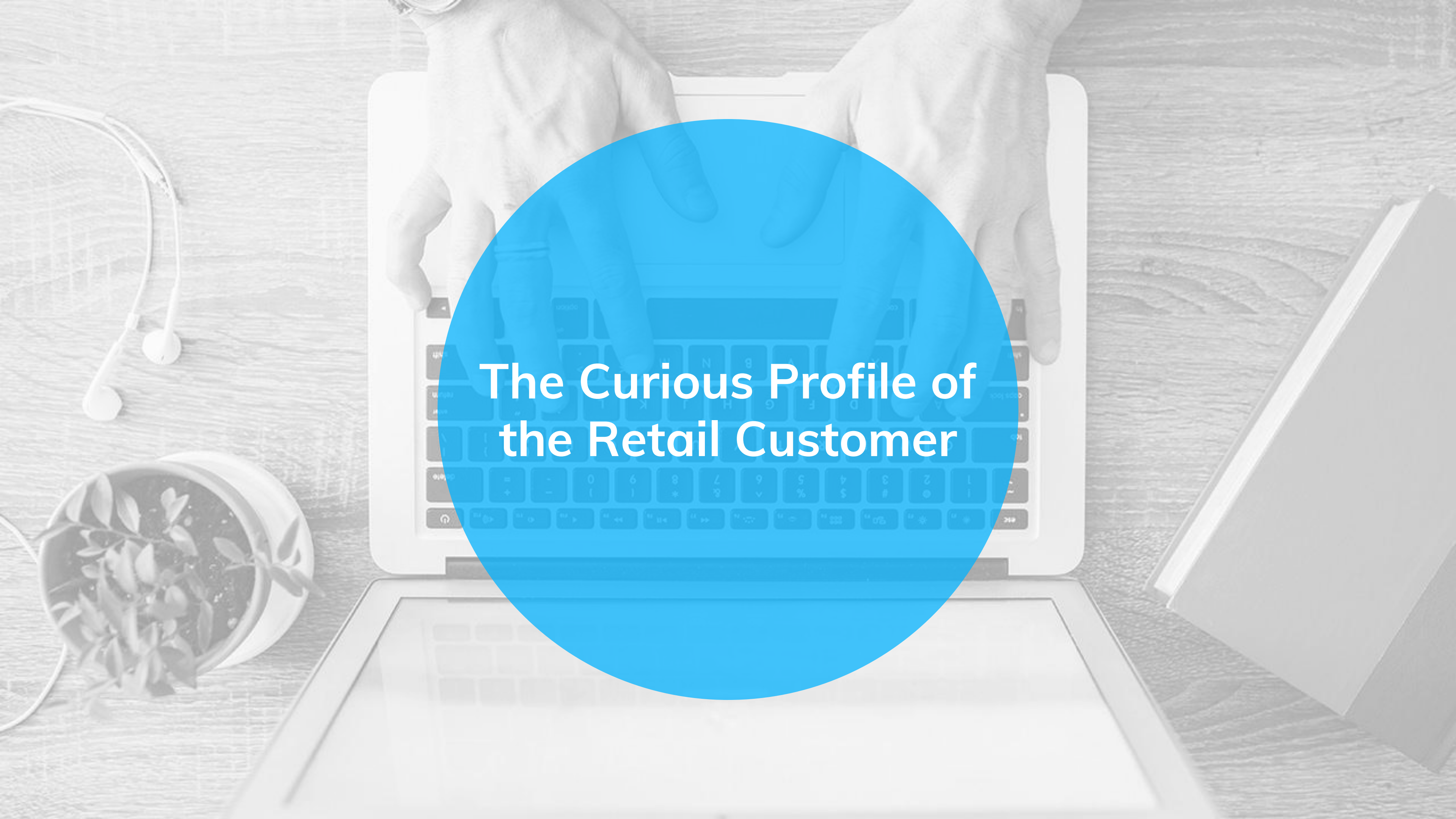
The Curious Profile of the Retail Customer

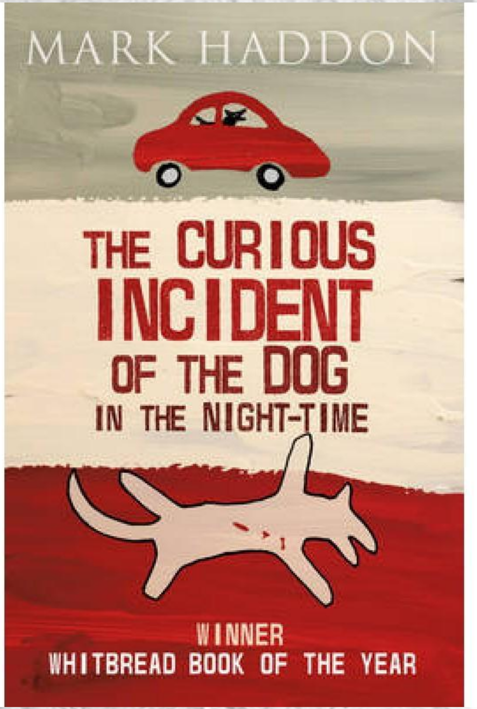


Hello!

I am Thomas

I am here because I love to share the little I know and also learn from you

You can reach me at:
jmjtomslabs@gmail.com



“Curious Profile of the Retail Customer”

Some things we want know about customers

- Anticipate customer behaviour
  - Product propensity
- How to entice the customer to buy more ?
  - Show associated products
- How to find similar customer groups to cross sell ?
  - Segment customers and recommend products
- Predict revenue by a particular customer
- Suggest product prices for a customer
- Predict customer churn
- Predict customer sentiment

Next Product To Buy

Different approaches and demonstration of sequence to sequence prediction techniques

Some approaches for Product Propensity

Considerations

- Predict in sequence
- Incorporate nuances like day ordered, what period did the order happen etc
- With each sequence get associated products

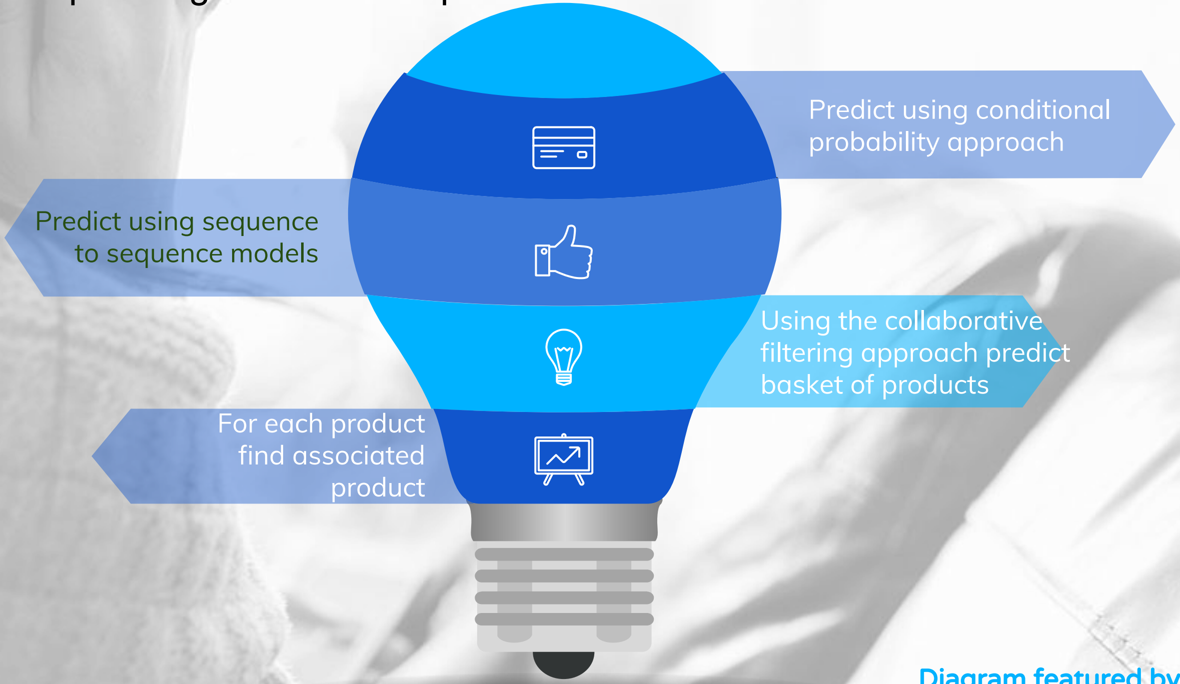


Diagram featured by http://slidemodel.com

The Data Set

- The data set used for this demo is taken from a Kaggle competition (Instacart Product Recommendation)
- Two key data sets and some ancillary data sets involved

Orders data set

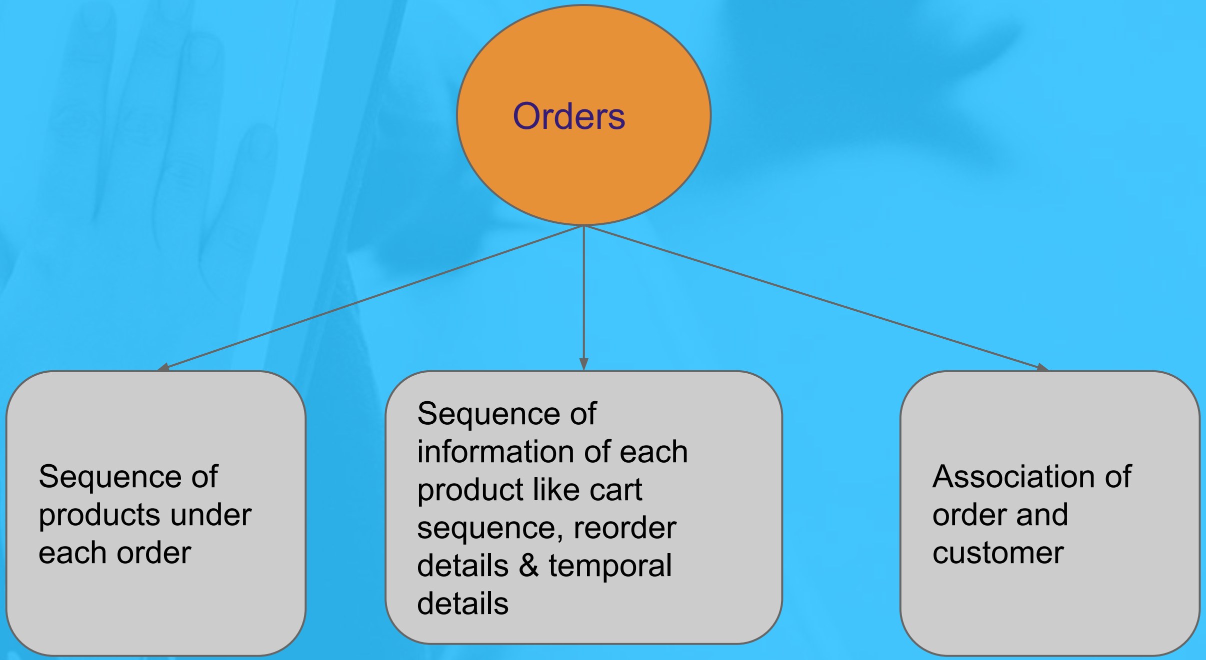
order_id	user_id	eval_set	order_number	order_dow	order_hour_of_day
<int>	<int>	<chr>	<int>	<int>	<chr>
2539329	1	prior	1	2	08
2398795	1	prior	2	3	07
473747	1	prior	3	3	12
2254736	1	prior	4	4	07
431534	1	prior	5	4	15
3367565	1	prior	6	2	07

Orders + Products data set

order_id	product_id	add_to_cart_order	reordered
<int>	<int>	<int>	<int>
2	33120	1	1
2	28985	2	1
2	9327	3	0
2	45918	4	1
2	30035	5	0
2	17794	6	1

Orders data set

Connecting the Data Sets together



Code for consolidating data sets : Order wise

```
# Connecting the Data Sets Together

```{r}
# Joining the prior order with Order details to get details of Users, DOW, HOD etc along with the Order detail
conUser <- order_products_prior %>% left_join(orders,by = "order_id")

# Adding the product list also
conUser <- conUser %>% left_join(products,by = "product_id")

# Summarizing the order details so as to get each record order wise
expOrdProd <- conUser %>% group_by(order_id) %>% summarise(Products = as.vector(list(product_id)),OrderToCart =
as.vector(list(add_to_cart_order)),Reorder = as.vector(list(reordered)),OrderNo = mean(order_number),DOW = mean(order_dow),HOD =
mean(as.numeric(order_hour_of_day)),DSPO = mean(days_since_prior_order),User = mean(user_id))
```
```

Order wise consolidated dataset

order_id <int>	Products <list>	OrderToCart <list>	Reorder <list>	OrderNo <dbl>	DOW <dbl>	HOD <dbl>	DSPO <dbl>	User <dbl>
2	<int [9]>	<int [9]>	<int [9]>	3	5	9	8	202279
3	<int [8]>	<int [8]>	<int [8]>	16	5	17	12	205970
4	<int [13]>	<int [13]>	<int [13]>	36	1	9	7	178520
5	<int [26]>	<int [26]>	<int [26]>	42	6	16	9	156122
6	<int [3]>	<int [3]>	<int [3]>	4	1	12	30	22352
7	<int [2]>	<int [2]>	<int [2]>	11	2	14	30	142903

Sequence data for each order, consolidated as list

```
[[1]]
[1] 33120 28985  9327 45918 30035 17794 40141  1819 43668

[[1]]
[1] 1 2 3 4 5 6 7 8 9
```

Preparing data for Sequence Prediction

Model Type

Multiple time steps and multiple products to be predicted

Use LSTM as a many to many model.

Time Steps

Each order to be broken down into a sequence of products acting as time steps

Features

Features for each time step to be the temporal data like order to cart, HOD, DOW, RO etc.

All categorical features to be one hot encoded and then stacked one below the other.

Code for creating dictionary by tokenization (text_tokenizer())

```r
library(keras)

# Creating the dictionary

otcList <- unique(unlist(expOrdProd$OrderToCart)) # Total 145 order to list items

otcList <- lapply(otcList, function(x) paste0('otc',x)) # To make the product ids unique, append it with the character"pd"

# Creating the tokenizer using text_tokenizer function

otcTok <- text_tokenizer(num_words = 146,char_level = FALSE) %>% fit_text_tokenizer(otcList)
```

Using tokenizer converting the data into one hot encoded format

```r
# Take the list of products
tempProd <- expOrdProd$OrderToCart[1] %>% lapply(function(x) paste0('otc',x))
# Create the list of products as sequence of integerst
seq2 <- otcTok$texts_to_sequences(tempProd)
# Padding the sequences to make the input lengths standard
pseq2 <- pad_sequences(seq2,maxlen = 15,padding = 'post')
# Creating one hot encoding with to_categorical() function
otcMat2 <- to_categorical(pseq2)
```

```r
# Stack different examples to create a three dimensional array of the form ( examples, Timesteps, Features )
dim(xTrain)
```

```
[1] 12 21 62
```

Dimensions, for input layer of sequence to sequence models

```
# Stack different examples to create a three dimensional array of the form ( examples, Timesteps, Features )
dim(xTrain)
```

```
[1] 12 21 62
```

Training the model using LSTM

```{r}
library(keras)

model <- keras_model_sequential()

model %>% layer_lstm(units = 32,input_shape = c(21,62),return_sequences = TRUE) %>% layer_activation('relu') %>% layer_dense(units
= 89) %>% layer_activation('softmax')

summary(model)

model %>% compile(optimizer = 'rmsprop',loss = 'categorical_crossentropy',metrics = c('accuracy'))

history <- model %>% fit(input_train,y_train,epochs = 100,batch_size = 125,validation_split = 0.2)
```

```
________________________________________________________________________________
Layer (type)                         Output Shape                     Param
#
================================================================================
lstm_9 (LSTM)                        (None, 21, 32)                   12160
________________________________________________________________________________
activation_17 (Activation)           (None, 21, 32)                   0
________________________________________________________________________________
dense_9 (Dense)                      (None, 21, 89)                   2937
________________________________________________________________________________
activation_18 (Activation)           (None, 21, 89)                   0
================================================================================
Total params: 15,097
Trainable params: 15,097
Non-trainable params: 0
```

```
# Prediction with LSTM network

```{r}
y_hat <- model %>% predict_classes(X_test)
```
```

Dynamics of the prediction process

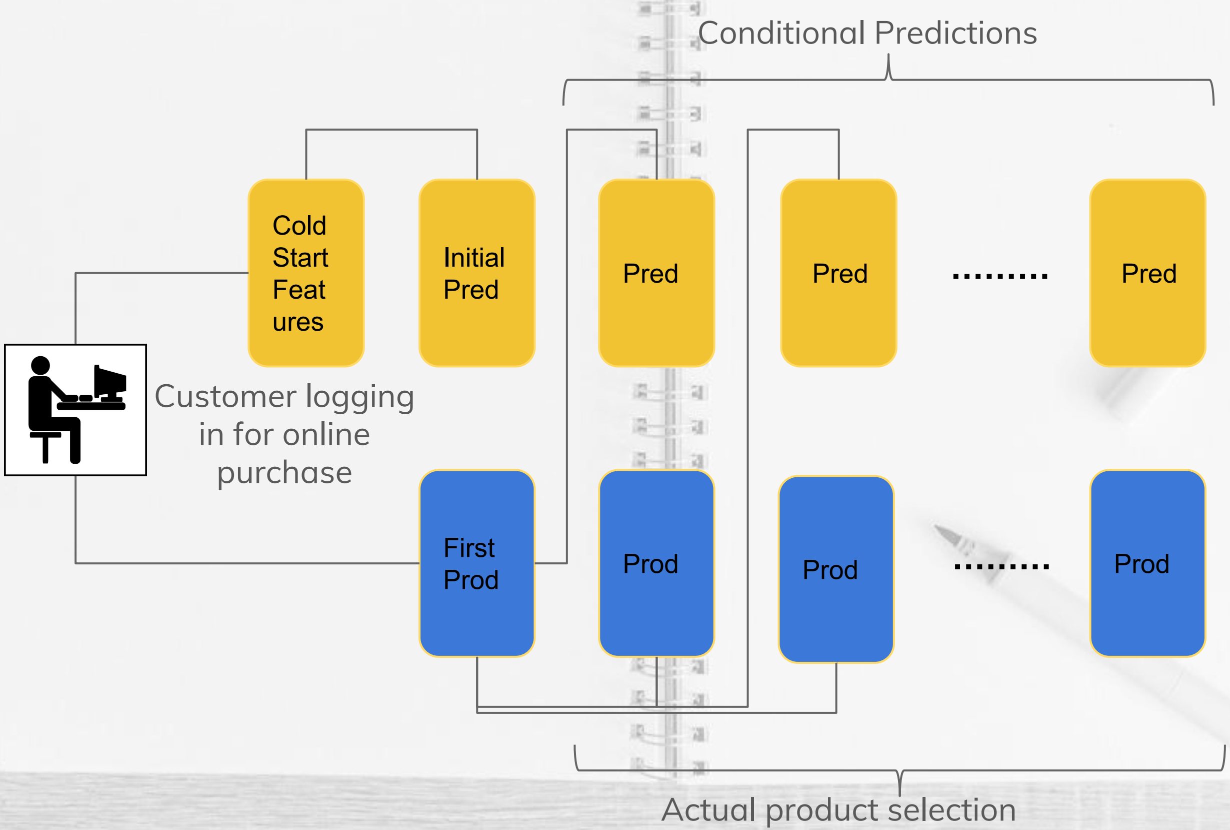


Diagram featured by

Approaches for other business outcomes related to customers

LET'S Summarize
Business Outcomes and approaches

Product Propensity
- ➜ Conditional probability
- ➜ Sequence to sequence models

Entice Customers
- ➜ Product associations using affinity analysis
- ➜ Bundling / product promotions using apriori algorithm

Customer Segmentation
- ➜ Collaborative filtering algorithms
- ➜ Clustering techniques.

Predicting Revenue
- ➜ Regression techniques
- ➜ Time series techniques
- ➜ Sequence to sequence models

Predicting Churn
- ➜ Normal classification models
- ➜ Deep learning models

Sentiment Analysis
- ➜ Word embeddings + Sequence to sequence models
- ➜ Classification models

Thanks!

Any questions?

You can find me at:
jmjtomslabs@gmail.com
Phone : 07926241557