Sequence Embedding for Prediction in Spark

- Pramod Singh
- Akshay Kulkarni
Clickstream Data
User
Web Journey
Digital Footprints
Clickstream Data
User
Multiple Users

Multiple Journeys

I got the product I was looking for, really happy!

Explicit Feedback

- Each visitor journey is different
- Each visitor journey has different length
- Difficult to compare journey’s of different visitors
- Difficult to predict satisfaction level from visitor journey
Business Use Cases : Clickstream Data

Applications:

• Predicting Visitor Conversion
• Predicting (visitor’s Interested Product) and recommending relevant content only
• Predicting visitor’s next best action on website
• Next Best personalized offer
• Visitor’s satisfaction level prediction
Use Case: Automobile Industry

Visitors → Website

- Booked Test Drive
- Not Booked Test Drive

Machine Learning Model → New Visitors
Approach

- Traditional Machine Learning
- Deep Learning
- Unsupervised Techniques
Traditional Approach

Sample processed dataset

<table>
<thead>
<tr>
<th>Visitor ID</th>
<th>Home Page</th>
<th>Reviews Page</th>
<th>...</th>
<th>Finance Page</th>
<th>Test Drive Booked</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>4</td>
<td>..</td>
<td>3</td>
<td>Yes</td>
</tr>
<tr>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>N</td>
<td>3</td>
<td>1</td>
<td>..</td>
<td>2</td>
<td>No</td>
</tr>
</tbody>
</table>

- **Input**: Each row of the dataset will contain list of count values (corresponding to each page by each visitor. Each row also contain the output label as True or False if a visitor has booked the test drive)
- **Output**: Probability of conversion (0 to 100 %)
## Traditional Approach: Feature Engineering

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Feature</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Page Category Count</td>
<td>Aggregated</td>
<td>Types of pages visitors are exploring</td>
</tr>
<tr>
<td>02</td>
<td>Time Spent Per Page Category</td>
<td>Aggregated</td>
<td>Time spent by visitors on different page types</td>
</tr>
<tr>
<td>03</td>
<td>Number of Activities</td>
<td>Aggregated</td>
<td>Total number of activities</td>
</tr>
<tr>
<td>04</td>
<td>Number of Sessions</td>
<td>Aggregated</td>
<td>Total number of session</td>
</tr>
</tbody>
</table>
Machine Learning Model: Random Forest

**Features**: Page Category Count, Time Spent, Total Session Count, Total Activity Count

**Model Results**:

<table>
<thead>
<tr>
<th></th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visitors</td>
<td>0.92</td>
</tr>
<tr>
<td>Converted Visitors</td>
<td>0.83</td>
</tr>
</tbody>
</table>

Out of **100 converted** visitors, ML Model is able to predict **83 of them as Converted**.
Word Embeddings
What is Embedding

“Embedding” is a process to encode objects (text, images, sequences) into continuous space (set of numeric values)

Embedding have changed several domains:
- Information Retrieval (including visual search)
- Recommendation Systems
## Word Embedding: One-Hot Encoding

<table>
<thead>
<tr>
<th>shoe</th>
<th>sneaker</th>
<th>tree</th>
<th>radio</th>
<th>boat</th>
<th>...</th>
<th>..</th>
<th>..</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Number of columns = Number of unique words in the vocabulary

### Issues:
- Sparse (all values are zero except one)
- Large embedding dimension (equal to vocab size)
- Semantic meaning not captured
  - “shoe” is at same distance as “tree”
Page-Category Embedding using Word2Vec

Input

\[ x_1 \quad 0 \quad x_2 \quad 0 \quad \cdots \quad x_i \quad 1 \quad \cdots \quad x_V \quad 0 \]

Embedding matrix

\[ \text{Vector of word } i \]

Matrix \( W \)

Hidden

\[ \begin{align*}
N &= \begin{bmatrix}
N \\
N \\
N \\
N
\end{bmatrix} \\
V &= \begin{bmatrix}
h_1 \\
h_2 \\
h_3 \\
\vdots \\
h_N
\end{bmatrix} \\
X &= \begin{bmatrix}
X \\
X \\
X \\
\vdots \\
X
\end{bmatrix}
\end{align*} \]

Output

softmax

\[ y_1 \quad 0 \quad y_2 \quad 0 \quad \cdots \quad y_j \quad 1 \quad \cdots \quad y_V \quad 0 \]

Matrix \( W' \)

\[ \text{Vector of word } j \]

Context matrix

\[ N' = \begin{bmatrix}
N' \\
N' \\
N' \\
\vdots \\
N'
\end{bmatrix} \]
Word Embedding: Prediction-Based Encoding

Benefits:

- Dense representation
- Smaller embedding dimension (equal to embedding size: $d$)
- Semantic meaning captured
  - "shoe" is at smaller distance than "tree"

[ 0.322 0.122 0.231 0.111 0.222 \ldots 0.445 ]

Word2Vec Model
(Google, 2013)
Sequence Embeddings
Journey Sequence Embedding

To represent the clickstream sequence in form of numerical vector of fixed size

**Benefits**:
- Easy to compare journeys with different length
- Find similar journeys
- Identify divergence in journeys
- Visualize journeys easily
# Journey Sequence Embedding

## Page-Category Embedding

<table>
<thead>
<tr>
<th>Category</th>
<th>Brochure</th>
<th>Specification</th>
<th>Finance</th>
<th>Reviews</th>
<th>Test Drive</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Column Length</strong> : <strong>Embedding Size</strong> : 100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Brochure</td>
<td>0.13</td>
<td>0.45</td>
<td>..</td>
<td>0.21</td>
<td>0.67</td>
</tr>
<tr>
<td>Specification</td>
<td>0.25</td>
<td>0.23</td>
<td>..</td>
<td>0.53</td>
<td>0.98</td>
</tr>
<tr>
<td>Finance</td>
<td>0.98</td>
<td>0.12</td>
<td>..</td>
<td>0.34</td>
<td>0.76</td>
</tr>
<tr>
<td>Reviews</td>
<td>0.21</td>
<td>0.53</td>
<td>..</td>
<td>0.23</td>
<td>0.87</td>
</tr>
<tr>
<td>Test Drive</td>
<td>0.87</td>
<td>0.24</td>
<td>..</td>
<td>0.63</td>
<td>0.25</td>
</tr>
</tbody>
</table>

## User Journey

<table>
<thead>
<tr>
<th>Category</th>
<th>Brochure</th>
<th>Specification</th>
<th>Finance</th>
<th>Reviews</th>
<th>Test Drive</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Time Spent</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Brochure</td>
<td>0.13</td>
<td>0.45</td>
<td>..</td>
<td>0.21</td>
<td>0.67</td>
</tr>
<tr>
<td>Specification</td>
<td>0.25</td>
<td>0.23</td>
<td>..</td>
<td>0.53</td>
<td>0.98</td>
</tr>
<tr>
<td>Finance</td>
<td>0.98</td>
<td>0.12</td>
<td>..</td>
<td>0.34</td>
<td>0.76</td>
</tr>
<tr>
<td>Reviews</td>
<td>0.21</td>
<td>0.53</td>
<td>..</td>
<td>0.23</td>
<td>0.87</td>
</tr>
<tr>
<td>Test Drive</td>
<td>0.87</td>
<td>0.24</td>
<td>..</td>
<td>0.63</td>
<td>0.25</td>
</tr>
</tbody>
</table>

## Time Spent

<table>
<thead>
<tr>
<th>Category</th>
<th>Brochure</th>
<th>Specification</th>
<th>Finance</th>
<th>Reviews</th>
<th>Test Drive</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Time Spent</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Brochure</td>
<td>0.43</td>
<td>0.75</td>
<td>0.98</td>
<td>0.55</td>
<td>0.87</td>
</tr>
<tr>
<td>Specification</td>
<td>0.75</td>
<td>0.23</td>
<td>..</td>
<td>0.53</td>
<td>0.98</td>
</tr>
<tr>
<td>Finance</td>
<td>0.34</td>
<td>0.76</td>
<td>..</td>
<td>0.63</td>
<td>0.25</td>
</tr>
<tr>
<td>Reviews</td>
<td>0.63</td>
<td>0.76</td>
<td>0.35</td>
<td>0.65</td>
<td>0.89</td>
</tr>
<tr>
<td>Test Drive</td>
<td>0.87</td>
<td>0.24</td>
<td>..</td>
<td>0.63</td>
<td>0.25</td>
</tr>
</tbody>
</table>

## Sequence Embedding

<table>
<thead>
<tr>
<th>Category</th>
<th>Brochure</th>
<th>Specification</th>
<th>Finance</th>
<th>Reviews</th>
<th>Test Drive</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sequence Embedding</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Brochure</td>
<td>0.53</td>
<td>0.76</td>
<td>0.35</td>
<td>0.65</td>
<td>0.89</td>
</tr>
<tr>
<td>Specification</td>
<td>0.76</td>
<td>0.23</td>
<td>..</td>
<td>0.53</td>
<td>0.98</td>
</tr>
<tr>
<td>Finance</td>
<td>0.35</td>
<td>0.76</td>
<td>..</td>
<td>0.63</td>
<td>0.25</td>
</tr>
<tr>
<td>Reviews</td>
<td>0.65</td>
<td>0.76</td>
<td>0.35</td>
<td>0.65</td>
<td>0.89</td>
</tr>
<tr>
<td>Test Drive</td>
<td>0.89</td>
<td>0.24</td>
<td>..</td>
<td>0.63</td>
<td>0.25</td>
</tr>
</tbody>
</table>
Page Category Embedding

**Modeling Approach**

*0.5 Million Sequence* used to create Page Category Embedding

**Word2Vec** model trained to represent each page-category to a vector of length **100**

**Validation**

2 dimensional representation created for the purpose of visualization (using **PCA** – Principle Component Analysis technique)

**Similar page-categories** are nearer in 2-dimensional space
Random Forest

Modeling Approach

Journey sequence embedding created by using page-categories embedding and time spent by each visitor on the respective page-categories.

Validation

Built propensity model (used journey sequence embedding as input and conversion as output).

Propensity model built using journey sequence embedding was able to identify 92% of converted visitors.

<table>
<thead>
<tr>
<th></th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Converted Visitors</td>
<td>0.91</td>
</tr>
<tr>
<td>Converted Visitors</td>
<td>0.92</td>
</tr>
</tbody>
</table>

Out of 100 converted visitors ML Model is able to predict 91 of them as Converted.

Converter Classification Results

<table>
<thead>
<tr>
<th></th>
<th>Correct Prediction</th>
<th>Misclassification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data was actually Converter</td>
<td>91%</td>
<td>9%</td>
</tr>
<tr>
<td>Non-converter Converter</td>
<td>8%</td>
<td>92%</td>
</tr>
</tbody>
</table>

Model Predicted ...
Thank You

In case of more questions:

Pramod Singh: prasingh137@sapient.com
Akshay Kulkarni: akulkarni7@sapient.com
GREAT INDIAN DEVELOPER SUMMIT™

Conference: April 23-26, Bangalore

Register early and get the best discounts!

www.developersummit.com  @greatindiandev  bit.ly/gidslinkedin