Predicting Azure Churn with Deep Learning and Explaining Predictions with LIME

Feng Zhu (Speaker)
Chao Zhong
Shijing Fang
Ryan Bouchard
Val Fontama (Speaker)

Xinying Song
PD Singh
Jianfeng Gao
Li Deng

Microsoft Research
Agenda

- Azure customers and churn
- Customer data and features
- Current ML pipelines and challenges
- Apply deep learning to improve model performance
- Explain model predictions using LIME
Azure Customers and Churn
Customer attrition is expensive!

**Acquiring customers is expensive**

- Industry growth leaders (>35% YoY) spend ~40% of revenue on acquiring new customers via sales and marketing.
- For renewal customers, they incentivize their sales team with 7%+ commission.

**Not all customers are acquired equally**

- The fastest growing companies generate up to 40% of their revenue from upsells (avg. is 16%) with goals of <3% monthly attrition.

A Predictive Model for Churn is Needed

To identify customers at high risk of churning
- The model outputs a ranked list of customers with the likelihood of churning
- Help prioritize a list of customers for churn intervention
- Help optimize resource allocation and ROI

Business impact
- Increase customer retention and Azure market share
- Increase profitability of Azure business
- Increase customer satisfaction and loyalty
Customer Features and ML Pipelines
Customer and Churn Definitions

- Each customer can have multiple subscriptions across multiple cloud services.
- Most usage and billing data are generated at subscription level and then aggregated to customer level.
- Two types of churns:
  - System churn: No active usage in 28 days
  - Usage churn: Cancel all subscriptions
Customer Data and Features

- Customer profiles consist of two types of data.
  - Snapshot data (billing info)
  - Time series data (detailed usage info)
- Use 8 weeks of snapshot and time series data to build customer feature sets.
- Label customers as churn or not churn in next 4 weeks

- Offer Type
- Tenure Age
- Segmentation

- Overall Usage
- Usage of Different Services
- Free Usage

Churn Labels:
- Usage Churn Label
- System Churn Label
Feature Extraction from Time Series Data

- Time Series (20+)
  - FreeUsage
  - Usage of Cloud Services
  - ...

- Extraction Functions (20+)
  - Min Max Mean SD Median
  - Number of Zero Values
  - Slope
  - Sum of Last Interval Values
  - ...

- Extracted Features (400+)
  - FreeUsage.min
  - TotalUnits.max
  - ...

400+ Features
1. **“Feature explosion?”** – As we increase # of time series, features extracted from time series will grow exponentially.

2. **“Squeeze more juice?”** – Can we further improve the model performance?

3. **“Should I trust model/scores?”** - If end users do not trust a model or a prediction, they will not use it.

---

**Churn V1 ML Pipeline**

**Customer Data**
- Time series usage
- Snapshot billing status

**Feature Engineering**
- Extract features from time series
- Combine all features together

**Random Forest Model**
- Classification model
- 400+ features

**Churn Risk Scores**
- Weekly scores for all active customers

**End Users**
- Take actions based on prediction scores
Churn V2 ML Pipeline

1. Deep learning model – 1) “automated” feature engineering 2) boosts prediction accuracy

2. LIME – An interface to explain model and predictions to end users
Deep Learning Models
What is Deep Learning (DL)?

- Deep learning is a class of neural network based ML algorithms
  - Transform raw inputs through multiple layers of neural networks
  - “unreasonably effective” in speech/image recognition and NLP.
  - DL excels with unstructured data

- Deep learning can be applied to many business problems.
  - Churn prediction
  - Recommender

Overview of DL Model Architecture

- The model is a “hybrid” model consisting of two types of neural networks.
  - Deep neural networks (DNN) layers
    - Takes static features as input
    - Fully connected MLP with add-ons
  - Recurrent neural networks (RNN) layers
    - Takes time series as input
    - Long Short Term Memory (LSTM) networks

“In Deep Learning, Architecture Engineering is the New Feature Engineering.”
Five identical DNN layers are stacked to transform static features.

- Input dimension of DNN layer 1 equals input dimension of static feature input
- DNN 2-5 have the same input and output dimensions
- **Batch normalization** layer is for faster convergence and “gradient vanishing” mitigation
- **Highway-like** structure is to mitigate “gradient vanishing” and improve accuracy
- The output of DNN layers is the merged output from each layer
Two identical RNN (LSTM) layers are stacked to transform time series data.

- The input dimension is 18*56. (18 time series and 56 time steps).
- Each LSTM layer has 128*56 hidden states. (128 hidden nodes and 56 time steps)
- Highway-like structure is also added to mitigate “gradient vanishing” and improve accuracy
- The final output of RNN layers is the merged states from the last time step.
## Model Performance Improved

<table>
<thead>
<tr>
<th>Model</th>
<th>AUC</th>
<th>Precision-Recall AUC</th>
<th>Recall (@Precision =0.562)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Forest (V1 model)</td>
<td>0.836</td>
<td>0.325</td>
<td>0.283</td>
</tr>
<tr>
<td>DNN Layers only</td>
<td>0.854 (+1.8%)</td>
<td>0.352 (+2.7%)</td>
<td>0.32 (+3.7%)</td>
</tr>
<tr>
<td>DNN + RNN Layers</td>
<td>0.863 (+2.7%)</td>
<td>0.383 (+5.8%)</td>
<td>0.343 (+6%)</td>
</tr>
</tbody>
</table>
The payout curve shows significant potential margin from DL model.

- Select top X percent of customer ranked by the scores
- Apply churn intervention to these customers
- Assume revenue per churning customer is $m, cost of intervention per customer is $n
- \( Y = \text{Payout} = \text{total revenue saved} - \text{total cost of intervention} \)
- Payout of new model - Payout of old model
Explaining Model Predictions Using LIME
Customer Usage Data
- Time series usage
- Snapshot billing status

Deep Learning Model
- Deep neural networks
- Recurrent neural networks

Risk Scores
- Weekly scores for all active customers

Model Explaining
- LIME algorithms
- Explain the scores at customer level

End Users
- Trust the model more
- Easy to start the conversion
- Take actions based on churn scores

Making Sense of Predictions
LIME is an interface between scores and end users.

- **Local Interpretable Model-agnostic Explanations**
- Use a simple model to approximate the scoring logic of a black-box classifier
- Fit the original classifier locally using sparse linear models through sampling for local exploration
An Example Using LIME

- LIME takes a scoring function and a test instance as input.
  - Assume a model is trained and its scoring function (classifier.predict_proba) is obtained.
  - Choose a test instance (test_example) which needs to explain.
  - A sparse linear model is built from the perturbations of the test instance.
  - Show the coefficients as supporting evidences of predictions for this test instance.

```
import lime
exp = explainer.explain_instance(test_example, classifier.predict_proba, num_features=6)
```
Explaining Churn Score of A Customer

- LIME can provide the evidences of churning and reduce “information overwhelm”
  - Pick on customer for explaining
  - Supporting features are shown along with the score
  - Visualize only the related time series (out of many time series)
  - End users can make informed decisions on if they should reach out to the customer.
Summary
Lessons Learned and Future Plans

- **Lessons learned**
  - “In Deep Learning, Architecture Engineering is the New Feature Engineering.”
  - Quantify the business value to motivate stakeholders
  - Making sense of predictions helps operationalize the model.

- **Future Plans**
  - Explore other network architectures for faster training and higher accuracy
  - Incorporate reinforcement learning to recommend the best churn intervention action
Thanks! Questions?