Chris Lin

Data Scientist / Analyst |Statistics Major and Economics |University of British Columbia
Phone: (778) 869-9665 | chris.chunyu.lin@gmail.com
Github: github.com/ChrisL088 |Linkedin: linkedin.com/in/chriscylin/

Detailed project description

Data Science:

1. Predict online advertisement budget (IBM) :

The goal is to help businesses predict how much they should spend on online Ad to reach their desired sales goal. The idea is when the manager inputs the sales goal, targeted customer demographics, and other crucial information such as duration of the Ad and geographic regions..etc, the algorithm can suggest the amount of money that should be spent on this Ad.

Technology used:
Python (Numpy, TensorFlow Keras, Pandas, Matplotlib)

Data Source:
The data is extracted from Google analytics & Facebook Business on our clients' end. The data consists of a summary of their ad campaign performance, targeted audience demographics, targeted region, click through rate, duration of the ad, revenue, ad budget...etc.

Preparing the dataset:
I first prepared the dataset, starting with data cleaning. There were only a few null values so after I made sure removing the rows containing null values doesn't impact the overall average and spread, I went ahead and dropped those rows. Then I examined the outliers. With the aid of box plots and scatter plots, IQR method, I found some noticeable outliers in revenue where they are more than 1.5 * IQR above the third quartile. After consulting with our clients, those outliers are confirmed to be coming from A/B testing, as well as loyal customers who buy their product in bulk every almost quarter. I removed the outliers from A/B testing since the clients abandoned that particular strategy, and I kept the ones from loyal customers because they may be valuable information.

I checked multicollinearity with VIF, random forest search, along with multi-variable data visualization. I first ran a random forest search to find the top predictors, after choosing those variables, I ran a VIF (Variance Inflation Factor) test, where values for VIF exceeding 10 were regarded as indicating multicollinearity. I treated those variables by either combining them or removing them if they don't add additional values to the dataset.

A deep understanding of every variable is key in selecting the appropriate variables, besides consulting with business owners I also did some research on my own. After selecting the variables, I one hot encoded the categorical variables and normalized the numerical data, then joined the two data sources. For the last step of data preparation, I split the data into a train set (70%) and a test set(30%).

Building Baseline Model:
For this particular project, I employed the Tensorflow library. I first built a baseline model with one input layer and one output layer. The baseline model is a simple model that has a single fully connected hidden layer with the same number of neurons as input attributes.
Performance Matrix:
Since this is a regression problem. I chose to use MSE as the main matrix to evaluate how well this model is performing.

Model Selection:
Then I started adding different layers, changing the number of input layer and hidden layer neurons and trying different activation functions. I finally settled with a keras sequential model with 3 dense layers, two with 'relu' activation and one linear output layer, and 1 dropout layer. The activation function is there to induce non-linearity into the layers to produce better results. I chose 'relu' as the activation function because it is popular for regression problems and it performs really well.

The model looks something like this (but not exactly):
```
model = Sequential()
model.add(Dense(512, activation= 'relu', input_shape= [
len(train_dataset.keys())]))
model.add(Dropout(0.4))
model.add(Dense(256, activation= 'relu'))
model.add(Dense(1), activation= 'linear' )
```

For the optimizer, I used an ADAM optimization algorithm with a learning rate of 0.001 and a MSE loss function is optimized. I chose ADAM as it combines the good properties of Adadelta and RMSprop and tends to do better for most of the problems. Adam also has the fastest learning rate and is the most stable among others. This is the same metric that I used to evaluate the performance of the model because by taking the square root it gives me an error value I can easily understand in the context of the problem.

Training the model:
I train the model with 100 epochs, where each epoch is a pass through all the training data. With the help of an EarlyStopping, it stops the model as soon as the validation loss stops improving within 5 epochs. By doing this I could save computational energy and time.

Making prediction:
Using the final model to make predictions. I got a mean absolute error of $17.632, this is a pretty good result as Ad budget usually ranges from hundreds to thousands.

Result:
The marketing director can now know how much to spend on the next few online advertisements. By knowing how much they're spending on online advertisement, the company can avoid over and under-spending on the online marketing, manage money effectively, and allocate appropriate resources to projects.

Improvement:
One of the improvements I want to make is automating choice of learning rate.
It's crucial for model training to have a good choice of optimizer and learning rate, although in this case automation is not required because this is a single use case. However, If I were to scale this project to a bigger level, the model might change depending on the type of company I am working with. In this case, the previous optimal hyper-parameters may no longer be ideal.
I would use Grid search to find the optimal params —for each hyper-parameter, create a list of possible values. Then for each combination of possible hyper-parameter values, train the network and measure how it performs. The best hyper-parameters are those that give the best observed performance.

2. Custom name entity recognition (IBM):

This is a project I picked up with another data scientist on another team. The goal of the project is to design an algorithm that recognizes the name entity of phases in a sentence and use the entity to extract information from the data and make visualization accordingly.

Technology used:
Python (Numpy, TensorFlow Keras, Pandas, SpaCy)

Data Source:
Gathered by Mohamed (co-worker) and I, the data is scraped from IBM internal resources as well as gathered from chatbot on Cognos Analytics and formatted into JSON file

Preparing the dataset:
The data cleaning and processing for this dataset is relatively easy. I removed special characters, extra white spaces, stop words, and performed stemming and tokenization. Then I encoded the labels and added padding to the maximum length of the sentence using pad_sequence.

Building baseline model:
SpaCy has a built-in model that does name entity recognition pretty well. I decided to use it as my baseline model.

Building model and Making prediction:
I used a Sequential model with an embedding layer and a bidirectional LSTM (Long Short Term Memory) and one dense layer. LSTM is well-suited to classify, process and predict time series given time lags of unknown duration. It trains the model by using back-propagation. Embedding layer is widely used for neural networks on text data and requires that the input data be integer encoded.
Choosing the default activation function softmax, and optimizer Adam with learning rate 0.001. I chose softmax because softmax activation function is great for building a multi-class classifier which solves the problem of assigning an instance to one class when the number of possible classes is larger than two.

The model looks something like this:
```
model = Sequential()
model.add(Masking(0, input_shape=(MAX_LENGTH,)))
model.add(InputLayer(input_shape=(MAX_LENGTH, )))
model.add(Embedding(len(pos2index.keys()), 16))
model.add(Bidirectional(LSTM(64, return_sequences=True)))
model.add(TimeDistributed(Dense(output_size)))
model.add(Activation('softmax'))
```

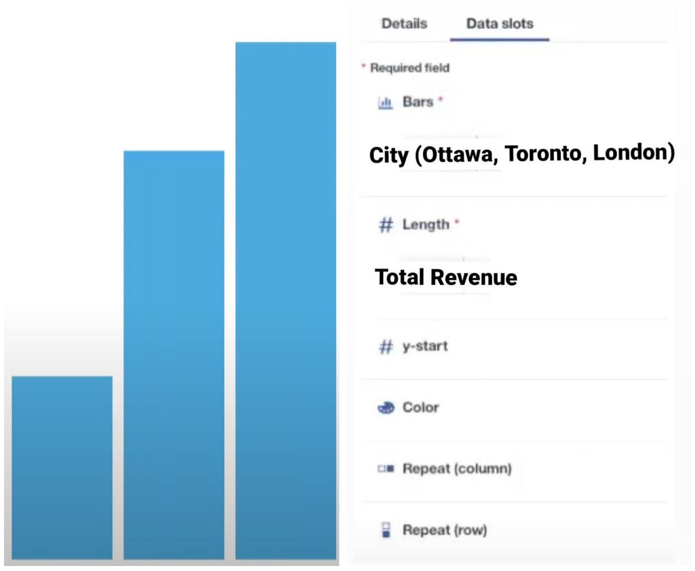
For the optimizer, I used an ADAM optimization algorithm with a learning rate of 0.001 and a categorical cross entropy loss function is optimized.

Training the model:
Then I trained the model with 50 epochs, where each epoch is a pass through all the training data. With the help of an EarlyStopping, it stops the model as soon as the validation loss stops improving within 2 epochs.

Improving the model:
Initial model has an error percentage of 18.24%. After implementing grid search to find the optimal optimizer and its learning rate, I adjusted the learning rate to 0.0015, and changed the number of hidden layer neurons from 128 to 256 and 64. By doing so the error percentage dropped to 13.6%.

Result:
Users who might have little experience in data analytics can still create dashboards and data visualization by speech (text). For example, if a customer typed into the chat bot: **show total revenue for Ottawa, London, and Toronto**. The algorithm will recognize `Total Revenue` as a column name and Ottawa, Toronto, and London as filters for column `City`, and make a plot/spreadsheet accordingly.

This is a visual example:



Personal projects (3, 4 & 5)

3. **Telco Company Customer Churn Analysis:**
   https://github.com/ChrisL088/customer_churn_analysis/blob/main/Telco_Customer_Churn_Analysis_Chris_Lin.pdf

4. **How Do Economic Indicators Impact the General Public's Interest in US Jobless Claims (Time Series Forecasting):**
   https://github.com/ChrisL088/Economic-indicators-impact-on-interest-in-jobless-claim/blob/master/Time_Series_Analysis_Chris_Lin.pdf

5. **Bias in Judicial System (research paper):**
   https://github.com/ChrisL088/Bias-In-Judicial-System

Data Analytics:

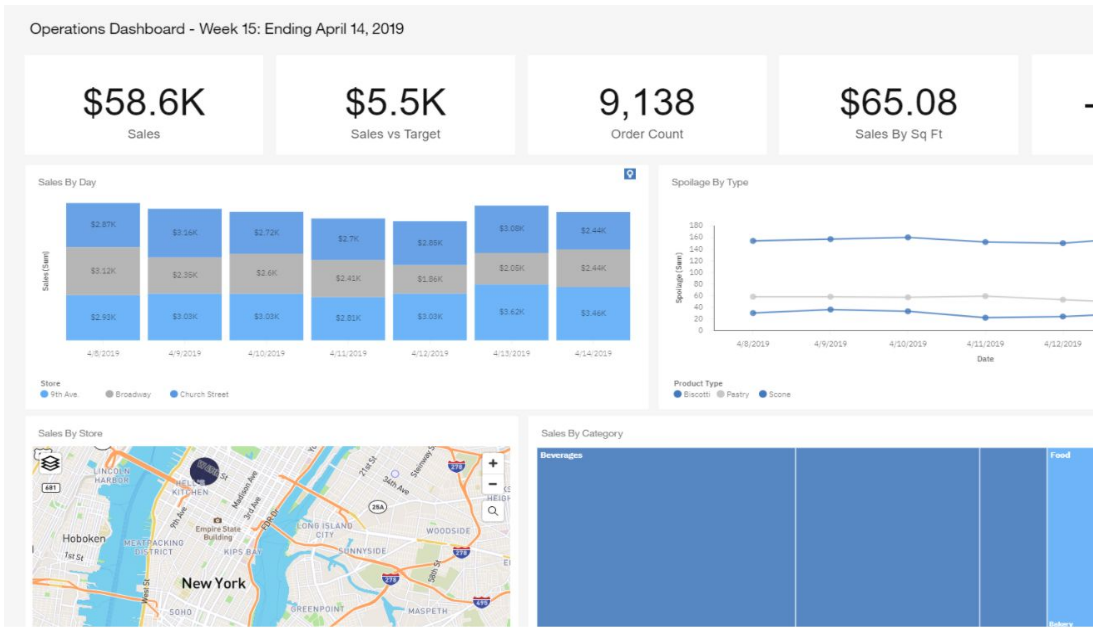
1. **Creating dashboards and reports (IBM):**
   Technology used: Cognos Analytics (IBM), Excel
   Here are some examples of the dashboards that my team and I created, tested, and maintained. However, the dashboards I created cannot be shared publicly as they are designed for specific customers and contain sensitive business information.

   Purpose:
   Design dashboards and reports for customers who want to keep track of their business performances but have little experience in data analysis and visualization. There were quite a few business owners who approached us with questions like how do I keep track of both my online and physical store performance? How do I know if one store is performing better or worse than the other in a simple, non time-consuming way?

Use case:
There's this one sports retail owner who owns two physical stores and an online store. He has all the data from both stores but hasn't found a way to effectively manage, view, and compare them. Before diving into the data, I first needed a good understanding of their KPIs. Throughout my research, indicators like sales, average order size, conversion rate, gross profit..etc are the major KPIs that tell how well a business is doing.

I designed a data pipeline that automatically converts the data into comparable format, and created a multi-page dashboard and report. The interactive dashboards keep track of both online and physical store Weekly Sales, Customer Conversion rates, Average transaction value and size, Monthly Expenses, and Gross Profit. The dashboard shows alerting messages when KPIs are underperforming or when business expenses exceed expectations.

This is an simplified example of how the dashboard looks like:


Operations Dashboard - Week 15: Ending April 14, 2019

$58.6K Sales | $5.5K Sales vs Target | 9,138 Order Count | $65.08 Sales By Sq Ft

In one case the dashboard I created showed the company is making money on both physical and online stores but physical stores require a lot more expenses to maintain. I also found that marketing budgets spent on promoting in store sales aren't as effective as online stores. It is very valuable to the business because now they can consider cutting down the marketing budget on physical stores and focus more on online store marketing.

Software Development:

1. **VirtualEdu** (link to demo video):

   Ever since the covid19 outbreak in march, the government has announced school and day care center closure to fight the spread of the virus. While university students are adapting to virtual lectures, younger kids who require interactive ways to learn and play have been putting their learning opportunity on pause. VirtualEdu is an online platform that provides younger kids a safe space to learn and have fun during covid lockdown. It keeps children entertained and engaged in activities through live classrooms, online playdates, interactive mini-games, and educational videos.

   Technology used:
   Frontend - React JS, CSS, HTML.
   Design - Figma

   I was responsible for designing and implementing front end features such as setting up the flow of the pages, arranging objects within a page, and adding fillable forms. I designed the layout and the flow using Figma; bright colours and cartoon graphics are used as they tend to draw attention from younger kids.

   Full link to the dashboard:
   https://community.ibm.com/accelerators/catalog/content/Monitor-loyalty-card-member-activity
   Full link to the VirtualEdu demo:
   https://www.youtube.com/watch?v=Vo-yl4oKH_8&t=4s