AI/ML TRANSLATOR AND INTERPRETER

Azure • Pre-Sales • Evangelism

Haddy El-Haggan

www.linkedin.com/in/hhaggan

hhaggan@outlook.com

https://github.com/hhaggan

About Me

Haddy has a great passion for entrepreneurship trying to apply the latest technology to satisfy customers' needs. He has developed some autonomous vehicles demo and AI drone with his team. Haddy has joined the MMAI program to uplift his advanced analytics skills. He wants to integrate Artificial Intelligence and Machine Learning on Azure with particular focus on autonomous systems, Finance and Strategy domains.

Haddy, a black belt master in Tae Kwon Do, enjoys physical challenges. You will find him in 100+ km bike ride, or 80-walk events. Haddy also enjoys languages, he is fluent in English, French and Arabic.

EDUCATION

Master of Management in Artificial Intelligence (MMAI) Queen's University — 2019 – 2020

Computer Engineering Bachelor Arab Academy for Sciences and Technology — 2008 – 2013

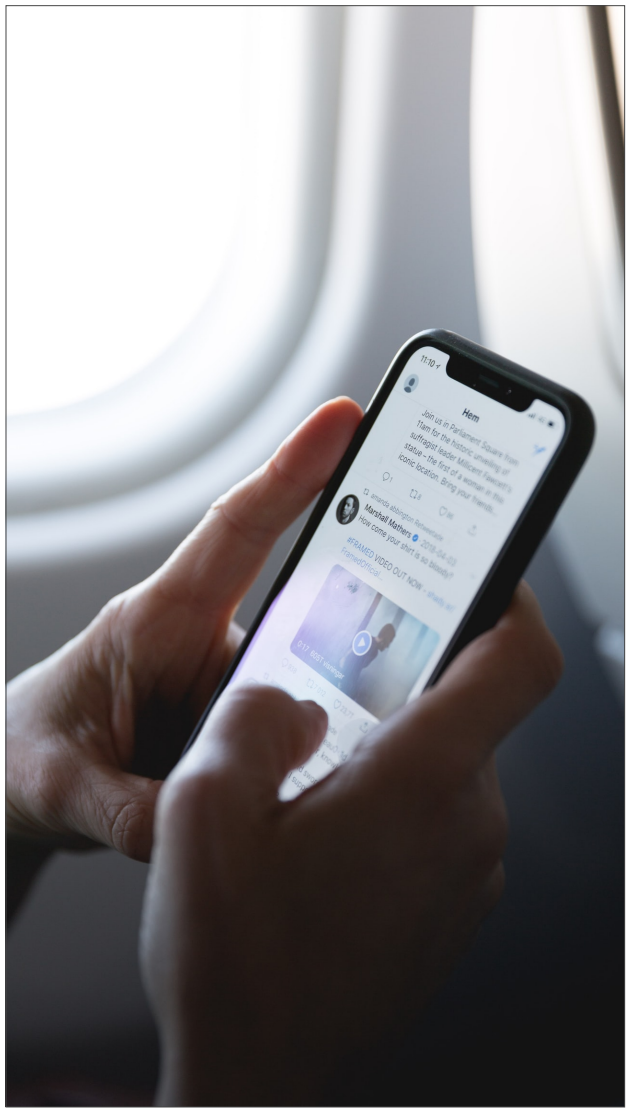
Portfolio Index

Team-Based Projects

Individual Projects

Sentiment Extraction

Team-based



Project Goal

500 million tweets are produced daily with information that could potentially impact organizations, companies, brands and people. ML-based techniques such as sentiment analysis have primarily focused on determining the polarity of a text, bringing relevant insights for decision-making purposes.

However, how could companies and organizations identify **which part of the text best explains that sentiment**? This is known as sentiment extraction

Solution Design

We participated in a Kaggle competition that required us to create a model that could look at tweets and identify which part best supports a specific sentiment. To address this challenge...

- We employed SimpleTransformers, a library that simplifies the use of transformers (a state-of-the-art deep learning architecture). Additionally, we based our approach on a question-answering schema, allowing us to structure data based on a **context** (the entire tweet), a **question** (positive, negative or neutral), and an **answer** (the extraction that best supports that sentiment)

- We used a pre-trained transformer in TensorFlow (Roberta), which gave us more flexibility when designing its architecture

Jaccard score, an index used for similarity, was the selected metric for this challenge.

Outcomes

Our two best models obtained a Jaccard score of 0.70 (SimpleTransformer-based) and 0.71 (TensorFlow-based).

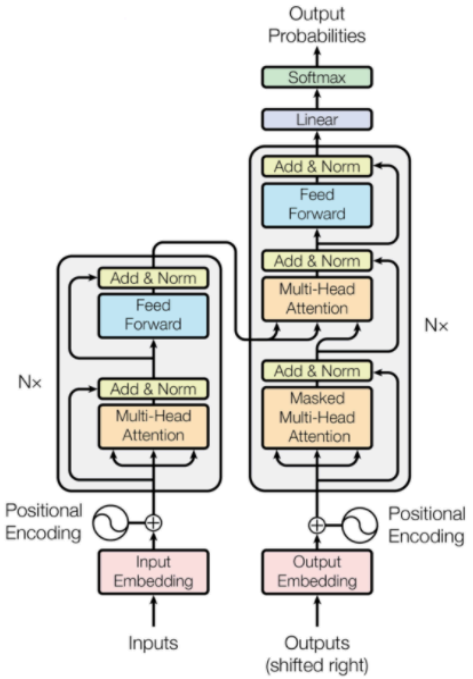
Our team also created setup and tutorial videos, a GitHub repository and a project wiki page with additional resources.

Next Steps

We could improve our results by:

- Increasing the number of models per ensemble and exploring more hyper-parameters (multi-sample dropout, Jaccard score custom loss, k-fold, optimizers, etc.)

- Employing more data pre-processing (removing extra spaces, custom labeling, etc.)



Source: **What is a Transformer?**

Stock Market Portfolio Management

Team-based

Project Goal

There were three main goals:

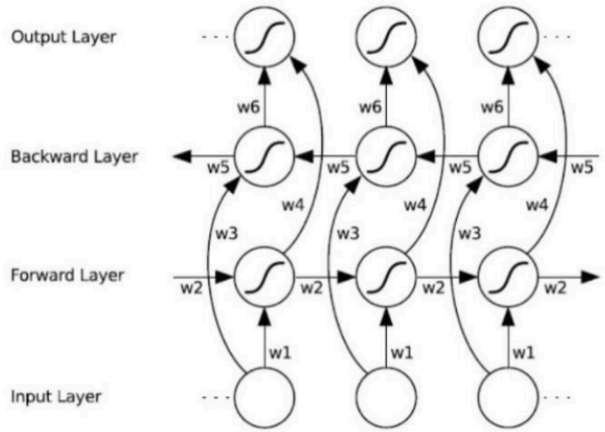
First, to use state-of-the-art machine learning (deep learning) techniques that would **predict the behaviour of stock returns** in the short-run.

Secondly, to determine **what stocks to buy, hold and sell** based on stock predictions.

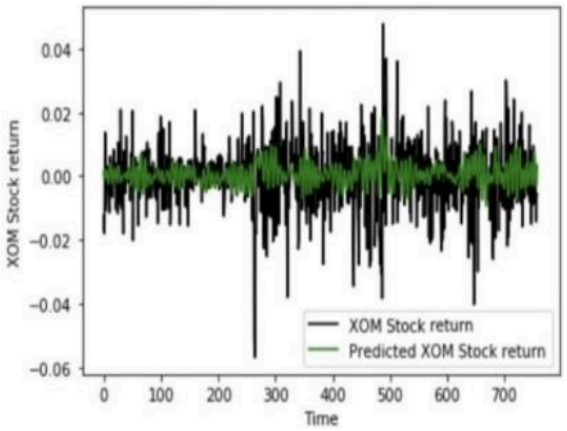
Finally, to **construct a portfolio** that would outperform the S&P 500 buy-and-hold benchmark without incurring in additional risk.

Solution Design

For the analysis and development of our Machine learning model, we selected the top 50 Stocks from the S&P 500 stock list. Yahoo Finance was tapped for historical data. After pre-processing, we used a deep learning neural network called Bi-LSTM. Some of the hyper-parameters used were: Number of units, dropout rate, type of optimizer, etc.



Source: **How to implement a different version of BiLSTM**



Outcomes

A set of back-tests based on prediction results were conducted for bull and bear market conditions. For the bull market, a long-short strategy based on equal-weighted portfolios outperformed the benchmark return with similar systematic risk as the market. For the bear market scenario, the equal-weighted portfolio achieved over 15% excess return with marginally higher systematic risk than the market.

Next Steps

To speed up model training without sacrificing performance, GRU (a variation of LSTM and RNN) could be considered.

There are opportunities to explore alternatives for portfolio construction. One method is the Candlestick analysis using GANs (Generative Adversarial Networks).

Finally, considerations for the long-run include exploring other market indexes with small-cap stocks and assets of diverse industries.

Customer Loyalty

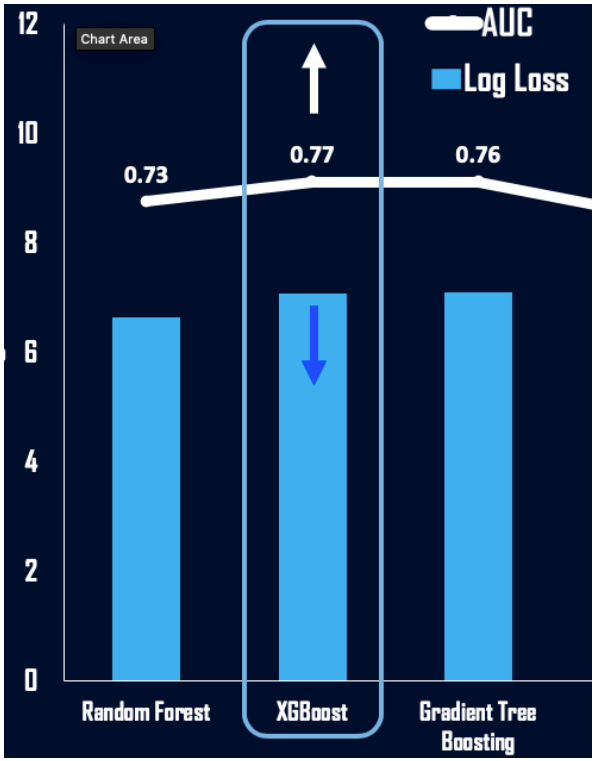
Team-based

Project Goal

The goal of the project was to boost our client's growth by predicting customer loyalty. Based on historical sales data, we used machine learning to identify patterns for diverse users and identified features are the most impactful.

Solution Design

We pre-processed the data and trained multiple models such as XGBoost, support vector machine, random forest, and logistic regression. Based on the metrics AUC-ROC and log loss, XGBoost was selected as the model that best identified the features that strengthen customer's loyalty to the brand.



Outcomes

From a binary classification (loyal or not loyal), we switched into a multi-class approach that better reflects customers' degree of loyalty. Finally, we provided quantified potential gains of investment (7% to 14% increase) if managers follow proposed suggestions on the features we identified.

Next Steps

We could develop and deploy an end-to-end solution in a production environment. This includes an integration of the dataset and model outputs into a dashboard, generating faster insights targeting multiple personas.

Project Goal

In 2011, the Mental Health Commission of Canada estimated that 1 in 5 Canadians experiences a mental health condition, representing $50 billion invested annually. With the rise of **COVID-19**, symptoms such as depression, anxiety and a sense of loneliness have impacted a larger population.

To address this, we have created an **AI-based framework** and a product concept that follow **design thinking principles**. Some of the pillars we included in our project were:

- Stakeholder assessment
- Privacy by design
- Risk assessment

Solution Design

SkAI is a **video diary application** that runs on mobile phones. Mental health practitioners will ask patients to record themselves answering guided questions or to share their thoughts and experiences via freestyle mode.

A second major feature is an alert system that is based on metrics and thresholds. This will determine whether the patient should be connected to a hotline.

This application is aimed to patients going through the therapy stage, who experience mild depression, anxiety or stress.

Research and Outcomes

Our interviews and surveys revealed many benefits that this solution could bring: form automation, ability to record more relevant patient insights, improved therapy and mental health outcomes. We also identified potential risks in the product such as reduced face-to-face interactions, an increased amount of data that can overwhelm the system, privacy concerns and false alerts.

In response to these, we assessed different scenarios that included the use of 3rd party cloud providers, data encryption and masking, anonymization, consent & ownership, and liability.

Next Steps

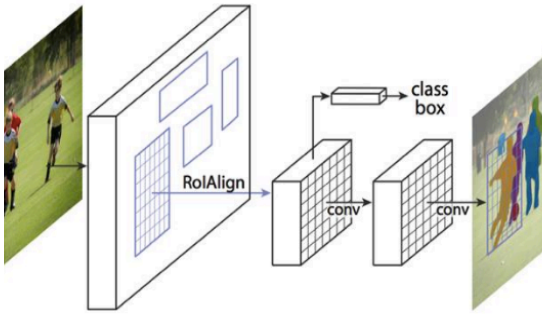
To execute this solution, we recommended a partnership with a research institution, bringing this way subject matter expertise required to develop better models. An in-house team will still be needed, consisting of data scientists, UX designers, software developers, and mental health experts.

Mental Health Assistance App

Team-based

Fare Evasion in Public Transit

Team-based



Source: **Mask R-CNN framework**

Project Goal

According to TTC's Audit, Risk and Compliance report (2019), fare evasion on streetcars costs $25M annually.

Our team completed a project using deep learning computer vision techniques to count the number of people present in a video feed. As a result of this solution, the TTC would be able to compare this estimate to the fare collected for the day, prioritize resources and control costs.

Solution Design

Object detection is a task that focuses on locating and classifying objects in an image. Unlike other traditional machine learning techniques, deep learning is better suited for this task, as it is able to identify complex patterns in this type of unstructured data.

We compared three object identification algorithms, including a sliding window image classifier, YOLOv3, and mask R-CNN. Finally, we determined the benefits and trade-offs of each approach by using performance metrics such as mean-average-precision (mAP).

Outcomes

As a result of the analysis, we recommended the use of mask R-CNN with transfer learning for this application. We were able to achieve a mAP of 77%.

The reason we want to use Mask R-CNN over YOLO is that the former has the added benefit that could be used for density mapping, a method used for counting or estimating the number of people on the streetcar.

Next Steps

The implementation of an AI model will be the first step in transforming the TTC's operational architecture into a data-centric organization. Additional revenue opportunities can be achieved by improving:

- Identification of fare evaders
- Automation of the ticketing process to reduce long processing times and increase revenue

Implementing both AI and automation will allow the TTC to reach 27% increase in profit in the long term.

Autonomous Drone

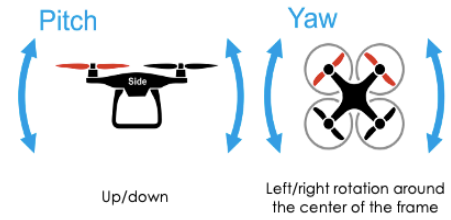
Team-based

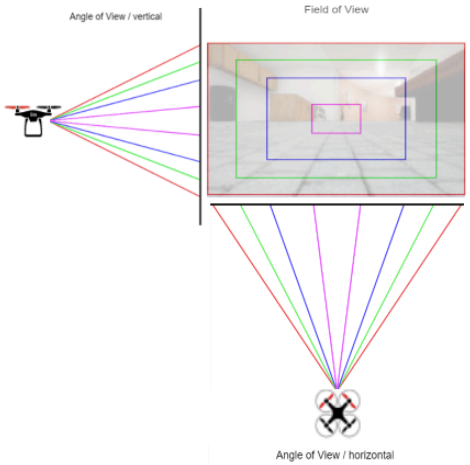
Project Goal

Our team used **reinforcement learning** frameworks and techniques to train a quadcopter to fly in a simulated indoor environment. This type of machine learning approach enables the agent to learn by trial and error and act under various surrounding complexities. We used Unreal Engine and Microsoft's AirSim, which acts as an interface between our python code and the simulator.

Solution Design

The drone had multiple front cameras that helped our agent make decisions based on the image depth (how far the drone/agent is from an object). This information was then fed into an algorithm called Q-learning, which provides positive reward if the drone does not collide, and a negative reward if it gets really close or clashes with an object/wall.





Outcomes

After 100 hours of training, our agent was able to navigate itself in an indoor environment without colliding for a few minutes.

Next Steps

- Make the drone reach a specific target in a space
- Build a more dynamic reward system that will help the agent navigate the environment and reach the target more effectively
- Run the solution into multiple environments
- Build an environment where multiple agents can work together to reach a target

Self-Driving Car

Individual Project

Outcomes

I was able to run the behaviour cloning at the end as a major milestone. Using the Udacity Simulator, I was able to drive the car teaching it to understand its environment by feeding it the images, speed and rotation, apply data augmentation and at the end the car was able to drive itself in the environment.

The solution in general is based on the Apollo program by Baidu where most of the technology providers and car manufacturers are part of. https://github.com/hhaggan/Self-Driving-Car-Behavior-Cloning/blob/master/video.mp4

Next Steps

- Work with a simulation for the HDMaps to help the car drive itself in an environment as a next step rather than using Simulators
- The self driving car has multiple sensor like LiDar and Radar, that I didn't leverage yet in my models.
- Apply reinforcement learning agent that can help the car navigate itself better given different circumstances. Not only relying on the behaviour cloning.

Projects Goal

I built multiple solutions for the self driving car as part of the Udacity program of Self Driving Car. The different projects are based on machine learning and trigonometry. The projects are mainly based on Computer Vision. I built applications to help the machine understand what is in front of it, imitate a driver behaviour using my own Deep Learning models or by transfer learning of other models.

Solution Design

The Self Driving Car has front cameras that helped ingesting the applications with video feeds. Next, I applied some data transformation or augmentation for the incoming data feed. Such data augmentation are like Computing the camera calibration matrix or Computing the distortion coefficients given a set of chessboard images or working with different picture layers.