Data Design and Analysis Report

FINS3645 20T3 Option 2: Neobank Project

Christopher Shi

z5165244

# Product Design

While the banking industry has traditionally been tied to personal and commercial lending, the wide availability of public data on global markets and the accessibility of the tools used to analyse this data will allow neobanks to develop in-house products with broader value propositions outside of traditional banking, meeting the demands and improving the experience of customers from a range of backgrounds and industries.

For this specific example, we will be looking at how technology will enable neobanks to develop new products for customers in the agricultural sector that will include:

Information centric products:

* Commodity price dashboards that provide real time predictions of future prices with respect to volume and market sentiment
* Industry relevant dashboards that provide real time predictions on data such as weather, environment, operating conditions, etc.

Risk centric products:

* The ability to purchase commodities as assets, as well as their financial instruments (options, futures, etc) based on predictive modelling.
* Smart insurance and loan products, using deep learning of market and meteorological trends to manage and mitigate risk.

As key with all neobanks currently on the market, these products will have a heavy focus on a seamless online user experience, and as such a high percentage of investment is recommended to be dedicated to platform engineering and customer accessibility, especially given that agricultural customers may not be in an area of high internet serviceability.

Therefore, these products should be:

* Engineered mobile first with a focus on storage and computation
  + Data such as customer cash flows are already on the neobanks system
  + Other data such as weather, commodity prices can be directly requested via API’s
* Built with a heavy automation focus, highly reusable, lightweight and fault tolerant
  + Feature engineering and predictive modelling computations should be automated and self-improving where possible, exploring the use of unsupervised learning.

For customers, the end product will a seamless experience with high quality relevant information and accurate predictions at their fingertips in the form of dashboards, as well as a singular place where they can get access to a variety of financial assets, smart insurance and loan products directly integrated with their already existing bank account.

The core aim of this project is to produce a proof-of-concept for the core backend data computations required for the above products, including:

* Core algorithms and models behind these neobank products, including further ETL, feature engineering and predictive modelling of a small subset of real market data.
* Lightweight, modularised and reusable code that can be easily scaled up for production scenarios. Using design patterns to produce DRY and loosely coupled code.

# Input Features / Feature Engineering

Feature engineering is the process of exploring data provided and preparing it before it’s used in models and computations. For this project, it involved the following steps (reflected in the code):

* File I/O and more ETL/data cleaning.
* Feature consideration and selection with further engineering where applicable. EG, deriving average temperature, wind direction as x/y components.
* Indexing, plotting and exporting for use in models.

For this proof of concept, we define four input sources (features):

Commodity news (headlines, articles) – fetched from news sources such as Bloomberg and Yahoo Finance via API’s. Data here will typically originate in a JSON format. In a production environment, there will be a need for a live downlink from news and media sources

Commodity market history (prices, volumes) – fetched from providers of historical market data such as the Yahoo Finance API, FactSet or otherwise. Historical data here will typically originate from a .xlsx format, which should be removed of graphs and converted to .csv format. In a production environment, there will be a need for a downlink directly from the market/exchange for live data.

Sample customer cash flow data (cash balance) – should already be within the neobank’s databases in the form of existing customer data.

Weather data (temperature, rainfall, wind speed and direction, humidity) – fetched from providers of meteorological data, such as BoM or otherwise. Data here typically originates as a .csv file. In a production environment, there will be need a downlink directly from source for live data.

In all of the above, the data will need to be preprocessed into a pandas Dataframe before further analysis, including more detailed feature engineering.

# Model Design

Model Design refers to the construction of architectures and models that are used to extract information from feature engineered data. For this project, it involved the following steps:

* Importing feature selections
* RNN preprocessing and design, hyperparameter tuning
* Model prediction, plotting model loss and performance
* Model export

For this proof of concept, we define the following models:

Sentiment analysis of commodity news using NLTK Vader – headlines were fed into a the NLTK Vader Sentiment Intensity Analyzer (SIA) to produce negative, neutral and positive sentiments from the given news headline. While NLTK Vader provides a basic valid model for sentiment analysis there are also cases where words in the headline do not exist in its lexicon and as such lower the accuracy. (See appendix 1.1) This can be rectified by modifying the NLTK lexicon to add support for industry specific keywords.

Deep learning of corn and wheat prices through LSTM RNN – prices and volumes were fed into a single layer LSTM RNN architecture with average 10 units followed by average 0.6 dropout. The motivation for this architecture was driven from the relatively small datasets provided, the small numbers of features that were fed in, and a response to overfitting. These models predict the price of the commodity 10 days into the future based on the last 60 days with mostly minimal loss (< 0.8) over 5 epochs. (See appendix 1.2) In a production environment, it is recommended to feed more data, such as market sentiment to more accurately model market supply and demand, the main forces of commodity prices.

Deep learning of customer cash flows through LSTM RNN – cash flows of clients were fed through a basic single layer LSTM RNN architecture similar to that of corn and wheat price modelling, with a set of tuned hyperparameters for this specific use case. The main limitation in this case was the lack of multivariate data, resulting in excessive overfitting. (See appendix 1.3) In a production environment, this could be improved by feeding data generated from sentiment analysis, commodity prices and weather to more accurately predict customer cash flows.

Deep learning of average temperature trends through LSTM RNN – weather attributes such as average temperature, rainfall, wind direction and humidity were fed into a single layer LSTM RNN architecture similarly motivated to the above, albeit with a higher number of units and less dropout to compensate for the more complicated multivariate dataset which resulted in a very close training/validation loss. (See appendix 1.4)

In all of the above, model predictions and loss were plotted to visualise performance, and LSTM RNN data were preprocessed before being fed into their respective architectures. For production environments, cloud compute platforms are recommended for faster architecture training.

# Model Implementation

Model Implementation refers to the use of trained models on live data, through which insights are then used to power the various financial products in question – dashboards, financial instrument and smart insurance pricing, etc. For this project, it involved the following steps:

* Model export and prediction on live data
* Export of predictions to be used on neobank products

For this proof of concept, model implementation is as follows:

Sentiment analysis of news headlines – NLTK Vader is a ready-to-go module capable of sentiment analysis from live news feeds and it is recommended to address concerns outlined in Model Design. Headline sentiment can be used to improve loss and accuracy on real time predictions of future commodity prices. It is also recommended to build a dashboard containing real time sentiment analysis of news from multiple sources, including mainstream and social media.

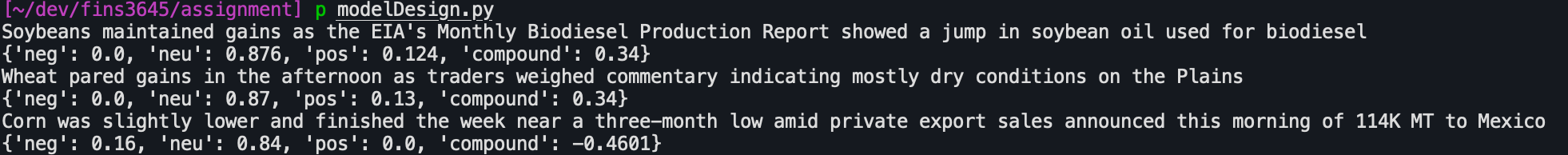
Predictive modelling of commodity prices – the addition of more data and sentiment representative of market demand will allow the LSTM RNN architecture to grow in complexity and more accurately predict future prices over a longer outlook. This can directly be uploaded to commodity price dashboards, used to improve the pricing of financial instruments such as options and futures for the neobank and its customers or even used to better price insurance products for agricultural clients.

Predictive modelling of customer cash flows – feeding more data streams into this LSTM RNN will allow the model to better learn the revenue/cost trends in the customer’s account, which can in turn be used to more accurately price insurance products for agricultural clients.

Predictive modelling of weather patterns – this architecture is relatively mature compared to the others in terms of its loss and accuracy. Average temperature predictions can be used on industry relevant dashboards, as well as used to more accurately price insurance products for agricultural clients.

In general, these exported models can be expanded to model more features, such as commodity trading volume, more weather patterns, debt trends in customer accounts, etc. in order to improve the granularity and sophistication of products offered by neobanks.

Appendix 1.1



Appendix 1.2 Commodity LSTM RNN Training and Validation loss

Corn:

A screenshot of a cell phone

Description automatically generated

Wheat:

A close up of a map

Description automatically generated

Appendix 1.3 Cash flows LSTM RNN Training and Validation Loss (Univariate)

A close up of a map

Description automatically generated

Appendix 1.4 Weather LSTM RNN Training and Validation Loss (Univariate)

A picture containing screenshot

Description automatically generated