# Forex
# Hedging With
# Machine Learning

Capstone project by Wei Hao

*Disclaimer*

# I'M NOT YOUR FINANCIAL ADVISOR. THIS IS NOT FINANCIAL ADVICE

## Do Your Own Due Diligence

# Table of contents

**01**

**Problem Statement**

**02**

**What is Forex Hedging ?**

explanation on what FX hedging is.

**03**

**Oanda API**

Broker's built in API

**04**

**Data exploration**

Data set explanation + Feature Engineering and transformation

**05**

**Models**

Logistic regression, XGB Classification, Keras Neural Network Model

**06**

**Implementation**

Implementation with Oanda

# Problem
## Statement

- Companies that conduct business internationally are exposed to foreign currency risk.

- Possibility that the value of a foreign currency will change in a way that is unfavorable to the company.
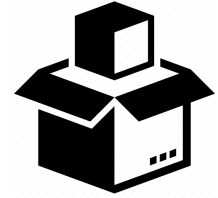
- Sales, Investments, Properties

## Original position

Goods sold worth USD 100K to Germany

→

As USD Strengthened against the EUR

→

Reducing it's value to USD 90k

## So you open a hedge

Open a short EUR/USD CFD contract

→

As EUR Declines against USD

→

Profit used to offset loss in value of the goods

# Problem Statement

- Companies typically use complex financial instruments
    - Forward Contracts
    - Futures Contracts
    - Options
    - CFDs
- Costs:
    - Upfront Premium cost
    - Ongoing cost for maintaining the hedge
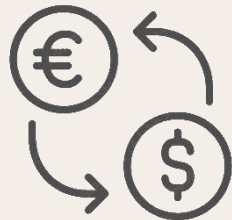    - Opportunity cost of not cashing in on favourable movements

# OANDA API

We'll be utilising OANDA's built in API to pull historical Data and also to implement our model into production

- They're simple
- Loads of tutorial online
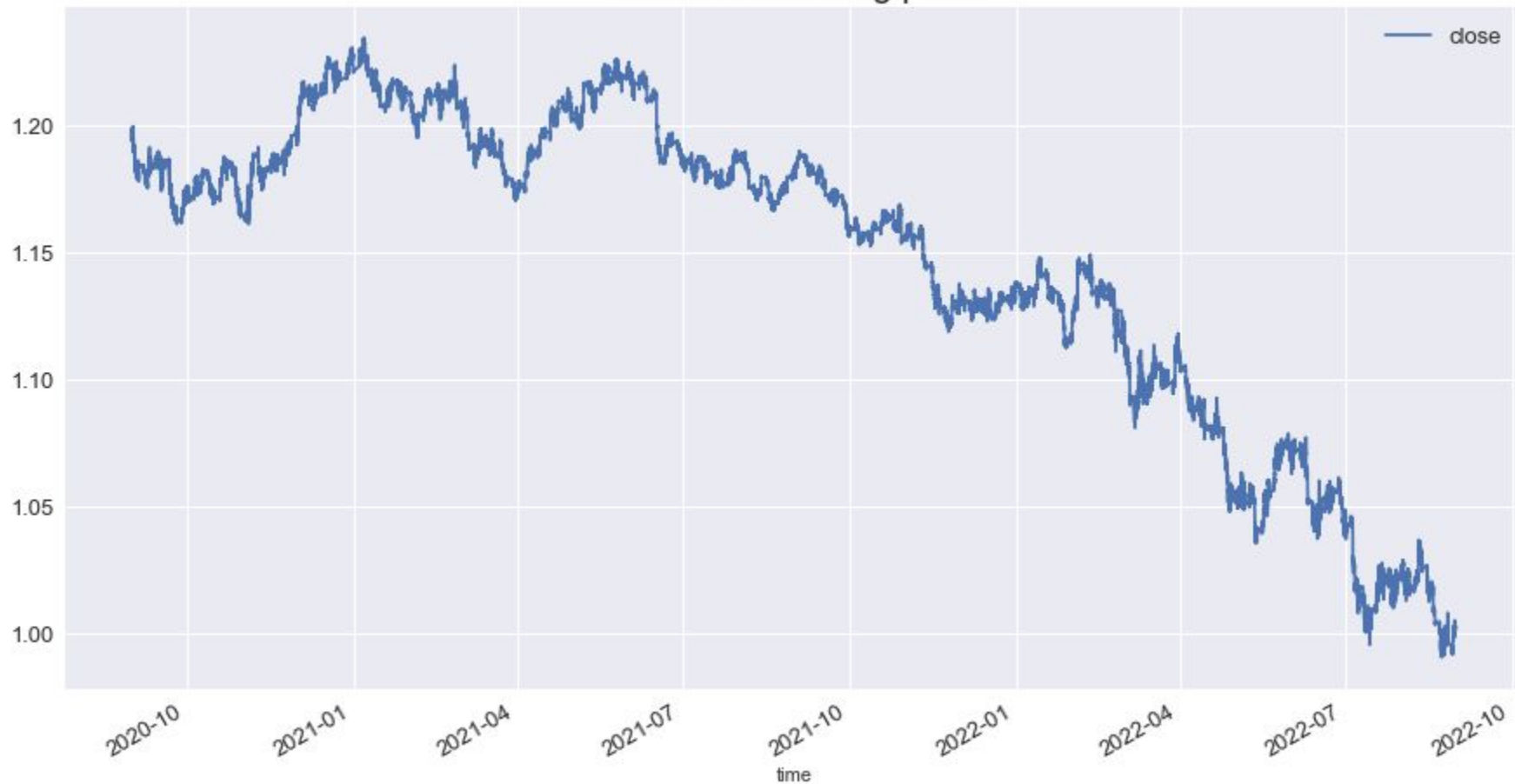- Libraries that compliment the API

# Dataset



**EUR/USD**

Period: Sep 20 - Aug 22 (24 mths)
Granularity: 5 mins
Datasets: 148,874

EUR/USD Closing price

# Defining **Target** Variable (Binary)

| time | close |
|------|-------|
| 2020-09-01 00:00:00 | 1.19436 |
| 2020-09-01 00:05:00 | 1.19451 |
| 2020-09-01 00:10:00 | 1.19472 |
| 2020-09-01 00:15:00 | 1.19468 |
| 2020-09-01 00:20:00 | 1.19448 |

| time | close | returns |
|------|-------|---------|
| 2020-09-01 00:00:00 | 1.19436 | NaN |
| 2020-09-01 00:05:00 | 1.19451 | 0.000126 |
| 2020-09-01 00:10:00 | 1.19472 | 0.000176 |
| 2020-09-01 00:15:00 | 1.19468 | -0.000033 |
| 2020-09-01 00:20:00 | 1.19448 | -0.000167 |

| time | close | returns | dir |
|------|-------|---------|-----|
| 2020-09-01 00:05:00 | 1.19451 | 0.000126 | 1 |
| 2020-09-01 00:10:00 | 1.19472 | 0.000176 | 1 |
| 2020-09-01 00:15:00 | 1.19468 | -0.000033 | 0 |
| 2020-09-01 00:20:00 | 1.19448 | -0.000167 | 0 |

We look at only the Close price

creating a column for the log returns between intervals

The Target variable indicates if the returns is positive or negative

# Defining  Target  Variable (Multiclass)

| | close |
|---|---|
| time | |
| 2020-09-01 00:00:00 | 1.19436 |
| 2020-09-01 00:05:00 | 1.19451 |
| 2020-09-01 00:10:00 | 1.19472 |
| 2020-09-01 00:15:00 | 1.19468 |
| 2020-09-01 00:20:00 | 1.19448 |

| | close | returns |
|---|---|---|
| time | | |
| 2020-09-01 00:00:00 | 1.19436 | NaN |
| 2020-09-01 00:05:00 | 1.19451 | 0.000126 |
| 2020-09-01 00:10:00 | 1.19472 | 0.000176 |
| 2020-09-01 00:15:00 | 1.19468 | -0.000033 |
| 2020-09-01 00:20:00 | 1.19448 | -0.000167 |

| vol_lag_4 | vol_lag_5 | dir_class |
|---|---|---|
| | | |
| 0.000408 | 0.000424 | 1 |
| 0.000407 | 0.000408 | -1 |
| 0.000407 | 0.000407 | 0 |
| 0.000409 | 0.000407 | -1 |

We look at only the Close price

creating a column for the log returns between intervals

-1 if Returns > 0.5 b.p. & dir ==-1
0 if returns <= 0.5 b.p.
1 if Returns > 0.5 b.p. & dir ==+1

# Class imbalance (Binary)

| Class | Count | % |
|-------|-------|-------|
| 0 | 76718 | 51.5% |
| 1 | 72155 | 48.5% |
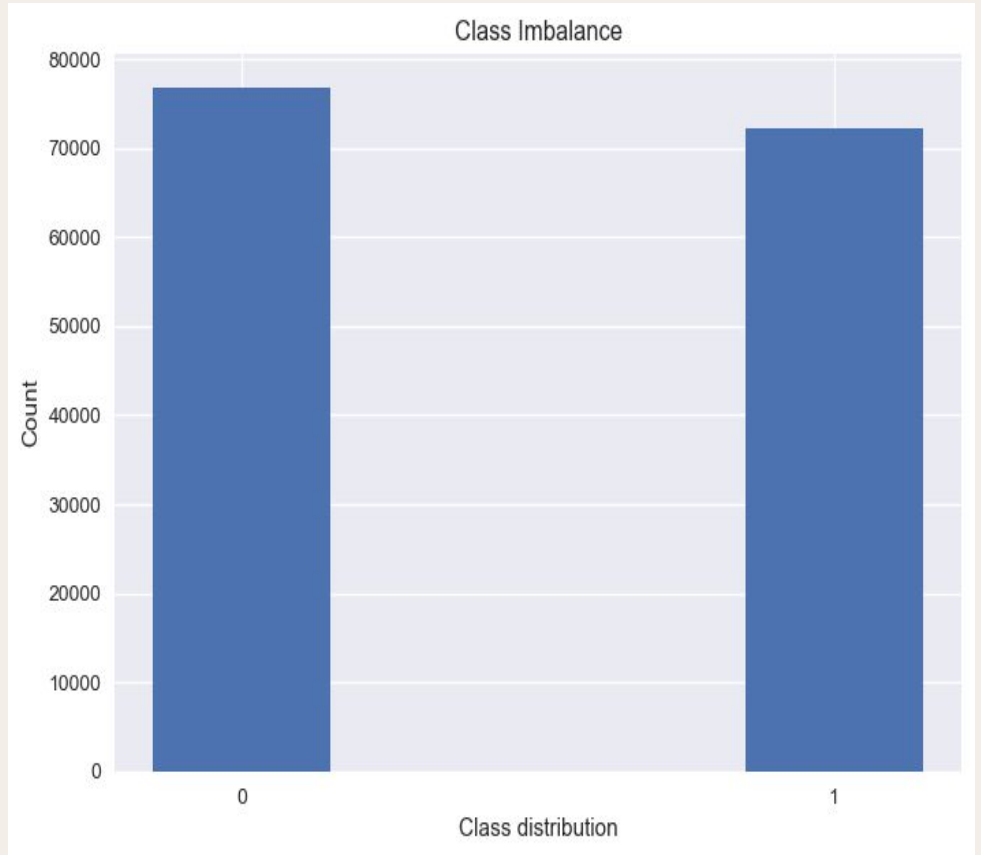
# Class imbalance (Multiclass)

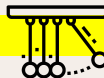| Class | Count | % |
|-------|-------|--------|
| -1 | 55942 | 37.59% |
| 0 | 55874 | 37.54% |
| 1 | 37004 | 24.86% |
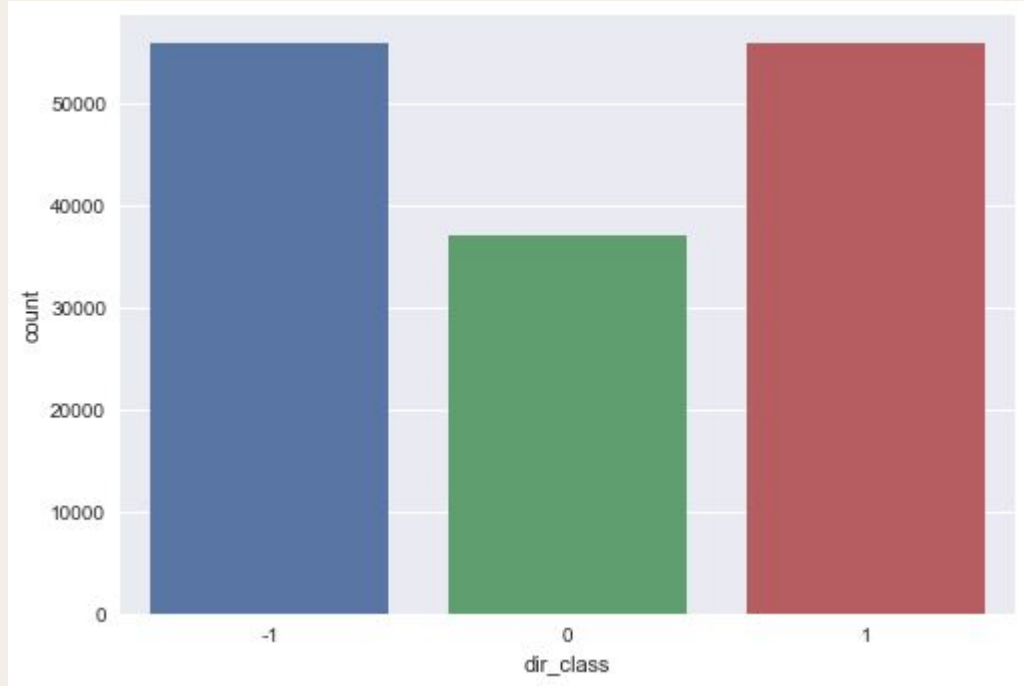
# Feature Transformation

## Simple Moving average

df['close'].rolling(25).mean() - df['close'].rolling(50).mean()

## Bollinger band

df['close'].rolling(25).mean() + df['close'].rolling(25).std()* 2
df['close'].rolling(25).mean() - df['close'].rolling(25).std()* 2

## Momentum

df["returns"].rolling(3).mean()

## Min/Max

df['close'].rolling(25).min() / df['close'] - 1
df['close'].rolling(25).max() / df['close'] - 1

## Volatility

df["returns"].rolling(25).std()

EUR/USD - SMA

EUR/USD - Bollinger

# Adding Lags

40 Columns

| time | bol1_lag_1 | bol1_lag_2 | bol1_lag_3 | bol1_lag_4 | bol1_lag_5 |
|---|---|---|---|---|---|
| 2020-09-01 04:30:00 | 1.200197 | 1.200351 | 1.200458 | 1.200514 | 1.200565 |
| 2020-09-01 04:35:00 | 1.200007 | 1.200197 | 1.200351 | 1.200458 | 1.200514 |
| 2020-09-01 04:40:00 | 1.199603 | 1.200007 | 1.200197 | 1.200351 | 1.200458 |
| 2020-09-01 04:45:00 | 1.199595 | 1.199603 | 1.200007 | 1.200197 | 1.200351 |
| 2020-09-01 04:50:00 | 1.199583 | 1.199595 | 1.199603 | 1.200007 | 1.200197 |

# Train Test Split

**Train 70%**
**Test 30%**

**With standard Scaling**



Training and Test Data
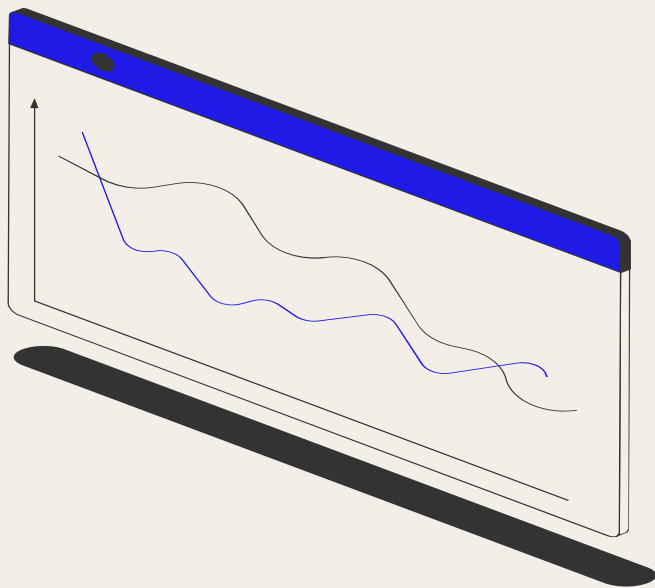
# Modeling

Log_reg

XGBoost

DNN

# Models

| Model | Classification | Train ACC | Test ACC | % |
|-------|----------------|-----------|----------|-----|
| **Log_reg** | MultiClass | 39.66% | 42.37% | -6.83% |
| **XGBoost** | Multiclass | 62.30% | 41.67% | 33.11% |
| **DNN** | Binary | 52.10% | 50.54% | 2.99% |

# Logistic Regression

In-Sample
Backtesting
With 1 b.p.

# DNN Model



Features ——— predict ———→ Market Direction

Direction
SMA
Bollinger upper
Bollinger Lower  } Lags 1-5
Min
Max
Momentum
Volatility

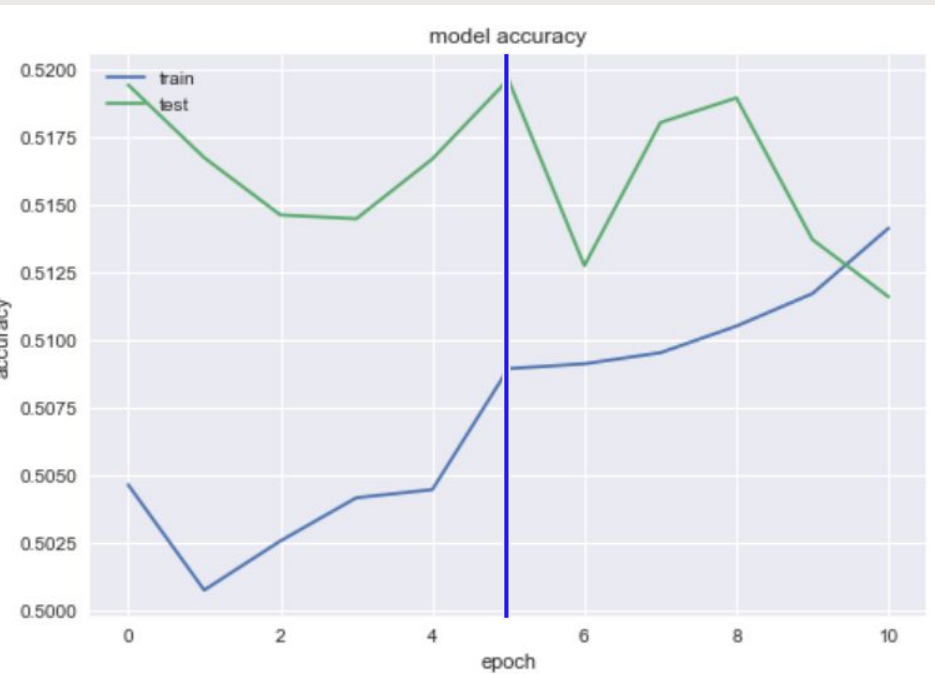0/1

Input Layer
40 Features

Hidden Layers
(3 Layers + 50 Neurons Relu)

Output layer
(Sigmoid)
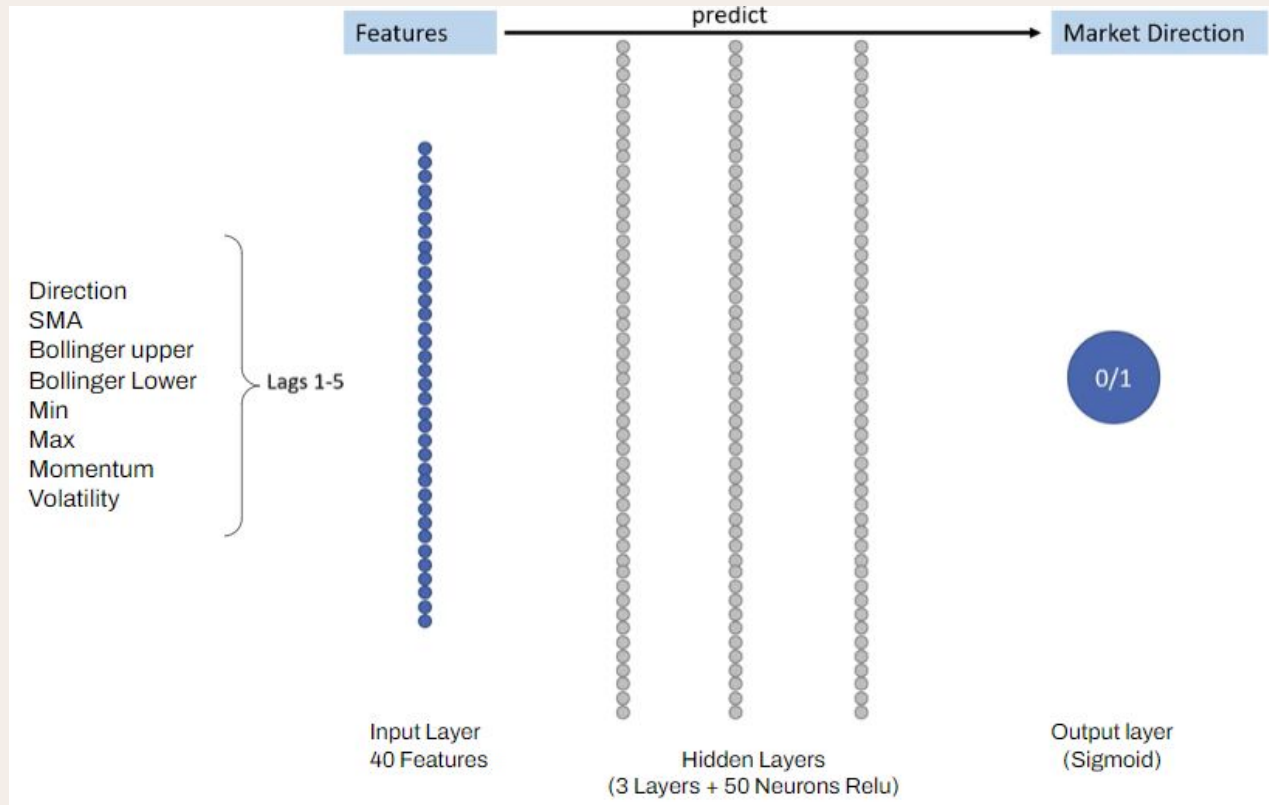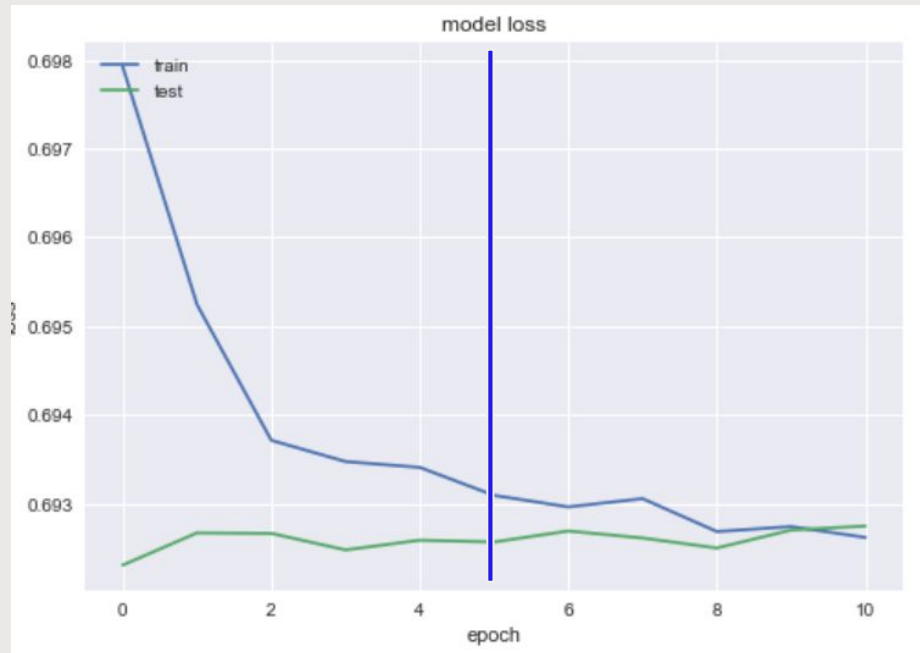
# DNN Model

# DNN Model

## In-sample Testing

For the DNN model
Used prediction probability to
Determine Buy, Sell or Hold

- Probability >=53 -> Buy
- Probability <= 47 -> Sell
- Probability between 47 & 53 -> Hold
                      ^         ^
                   (Exclusive)
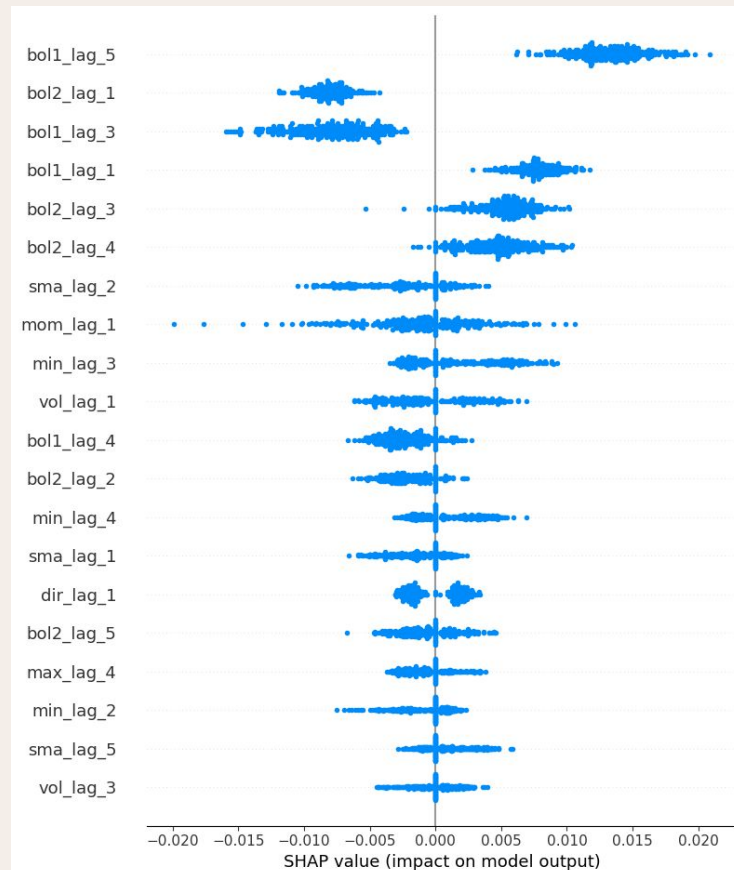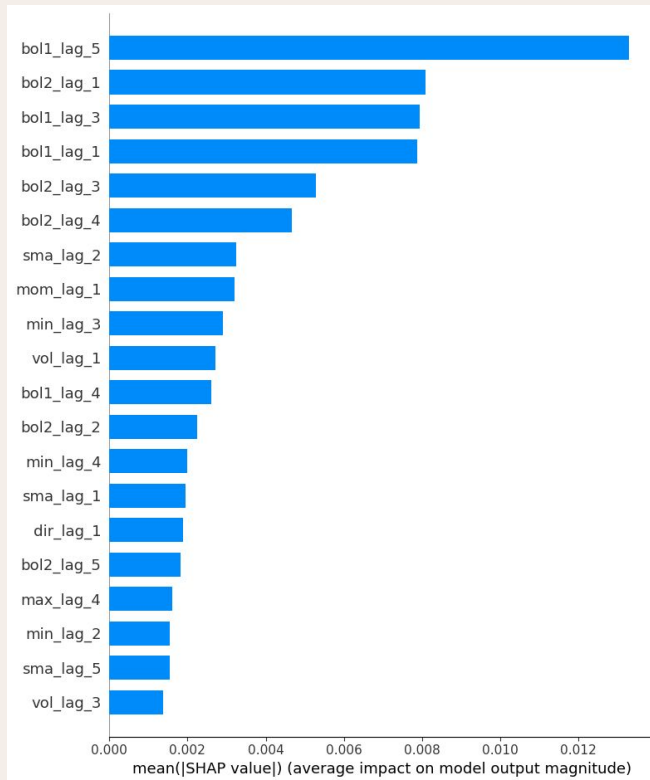
# DNN Model

Out-Sample
Forward Testing

# DNN Model

Out-Sample Forward Testing
With 1 b.p. spread

Strategy = 0.950971
Net strategy = 0.936157
Buy & hold = 0.888743
% diff = 5.065%

# SHAP

# Implementation

Implementing DNN model to OANDA

# DNN Model Implementation

622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653
654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685
686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717
718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749
750 751 75

----------------------------------------------------------------------------------------
2022-10-19T18:50:01.020118227Z | GOING LONG
2022-10-19T18:50:01.020118227Z | units = 100000.0 | price = 0.97679 | P&L = 0.0 | Cum P&L = 0.0
----------------------------------------------------------------------------------------

23/23 [==============================] - 0s 1ms/step766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784
785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816
817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848
849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880
881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912
91

# API test in 3 days

## $157
### Day 1
10 Trades
9 profits
1 Loss

## $40
### Day 2
2 Trades
2 Profits

## $53
### Day 3
2 Trades
2 Profits

Based on 100k USD per trade

# Conclusions

### Ease of use

Implementation is easy and needs minimal human intervention

### Performance

The model performs better as compared to buying and holding on to EUR.

### Cash in on Favourable movmt.

The model allows users to take advantage of favourable price movements

# Improvements

### Data Drift
Model Will need to be retrained frequently.

### More Technical indicators
We can experiment with more features to optimise the model for more profits.

### particle swarm optimization
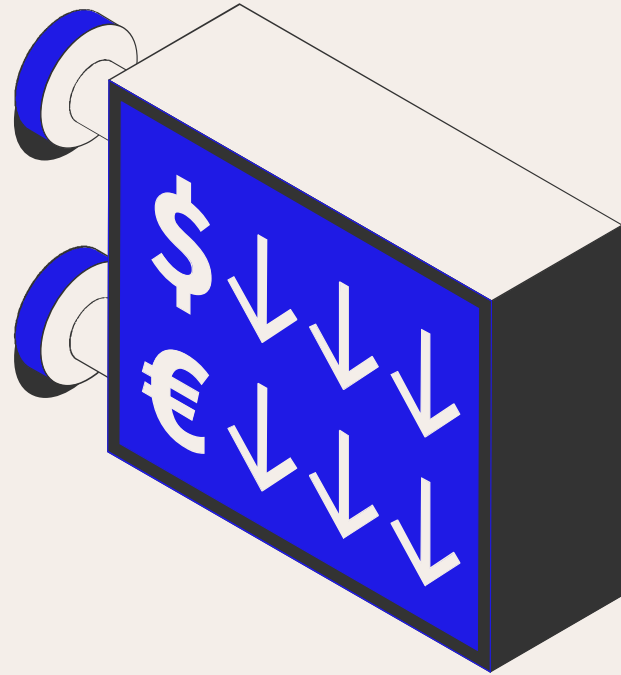We can try to utilise the PSO activator to see if the model performs better.

### Take Profit + Stop Loss
Add in a take profit/ stop loss via the OANDA API.

# Thanks!

**Do you have any questions?**

# Thanks!

Do you have any questions?

youremail@freepik.com
+91 620 421 838
yourwebsite.com