PGDMLAI - Capstone Assignment

Nuveen Sales Data Analysis

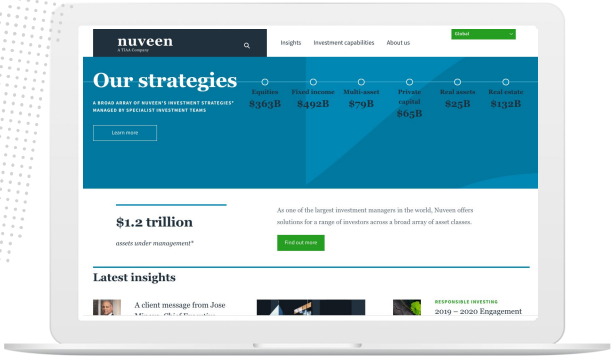
Submitted by: Abhinav Verma
CPE Registration No.: 201510637C
Student Reference No.: PGDMLAI/B-9591/163193
Email ID: candid.abhinav@gmail.com
LinkedIn: https://www.linkedin.com/in/candidabhinav/

AGENDA

Capstone Assignment - Help Nuveen sell better

BACKGROUND - SITUATIONAL ANALYSIS

NUVEEN - AN INTRODUCTION



ABOUT:

Nuveen(A TIAA Company) is a mutual fund company which markets and sells mutual funds to investors through investment professionals such as brokers, financial planners, and financial advisors.

KEY CHALLENGES:

Since the market is highly competitive and there is high sales cost, Nuveen needs to understand & analyze their sales data using Data Science to:

- ❑ acquire new clients cost-effectively
- ❑ sell more to existing clients
- ❑ reduce redemptions (ADR – acquire, develop, retain)

PROJECT OBJECTIVES

OBJECTIVE & EXPECTED OUTCOMES

OBJECTIVE: Assist Sales and Marketing by improving their targeting

Outcome 01

Predicting the following year's sales using data of previous year using a Regression Model.

Outcome 02

Estimate the probability of an Advisor adding a new fund in the following year, using a Classification Model

Outcome 03

Predict potentially profitable financial advisors for sales and marketing efforts by combining Classification & Regression Models

APPROACH & OVERALL METHODOLOGY

APPROACH - STEP OUTLINE

STEP 1

Objective identification -

1. What is the business objective?
2. What do we want to achieve from the project?
3. What are the possible suggestion outcomes?

STEP 2

Acquire Data - Collect data provided by Nuveen :

1. Transaction Data
2. Information on Investment firms and Financial Advisors

STEP 3

a) Data Pre-Processing : Cleaning the data

b) Explore the data - access data quality:

- Accuracy
- Completeness
- Reliability
- Relevance

c) Prepare the data : Merging the datasets, Data Munging

STEP 4

Feature/Variable selection -

1. Select the variables or features used for Regression
2. Select the variables or features used for Classification

STEP 5

Refine - Building processing pipelines for data processing

STEP 6

Model Evaluation & Selection - Combine Classification & Regression Models to give best Advisor prediction.

STEP 7

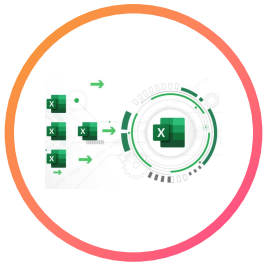
Recommendation - Provide recommendation based on the model predictions

DATA ANALYSIS

DATA - FROM WHERE & HOW USED

DATA SOURCES

- Nuveen Transaction Data (2018 & 2019) containing:
  a. Sales information
  b. Redemption information
  c. Information on Assets under Management.
- Nuveen Firm Information Data containing:
  a. Different sales Channels and subchannels
  b. firms information.



DATA PROCESSING & CLEANING ACTIONS

- Merging Transaction and Firm information data to have a single comprehensive data source
- Fill missing values with 0 as these indicate no sales or no new fund added
- Replace negative sales values with 0 as again they simply indicate no sales.
- Remove columns which are highly correlated as they unnecessarily complicate the prediction capability of the models.
- Normalizing and Powertransforming the data as the data is highly skewed and sparse.

FEATURE/VARIABLE SELECTION

FEATURES USED FOR REGRESSION

FEATURES

- ❑ Number of sales in last 12 M having value >=1$
- ❑ Number of redemption in last 12 M having value >=1$
- ❑ Number of funds sold to FA in last 12 months having value >=1$
- ❑ Number of funds redeemed by FA in last 12 months having value >=1$
- ❑ Number of asset class sold to FA in last 12 months having value >=1$
- ❑ Number of asset class redeemed by FA in last 12 months value >=1$
- ❑ Number of funds currently held
- ❑ Current Assets Under Management
- ❑ Total sales in current month
- ❑ Total sales in last 12 months (excluding current month)
- ❑ Total redemption in current month

Methodology

- ★ Regression is used to forecast the sales for next year based on sales figures of the previous year.
- ★ Used SelectKbest to select the best features for Regression

FEATURES USED FOR CLASSIFICATION

FEATURES

- ❑ Number of sales in last 12 M having value >=1$
- ❑ Number of redemption in last 12 M having value >=1$
- ❑ Number of sales in last 12 M having value >=10K$
- ❑ Number of funds sold to FA in last 12 months having value >=1$
- ❑ Number of funds redeemed by FA in last 12 months having value >=1$
- ❑ Number of funds currently held
- ❑ Asset under Management
- ❑ Total sales in current month
- ❑ Total sales in last 12 months (excluding current month)
- ❑ Total redemption in current month
- ❑ Total redemption in last 12 months (excluding current month)
- ❑ New funds added in the last 12 Months excluding current month
- ❑ AUM for different Asset class
- ❑ Sales channels

Methodology

- ★ Classification is used to classify whether an advisor will add a new fund(i.e. make a sales) or not.
- ★ Used RFE to select the best features for Classification

ANALYSIS SUMMARY

MODEL SELECTION

STEP 1

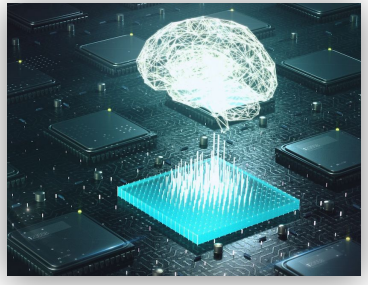
Select the Model which performs best Classification

Gradient Boosting Classifier

STEP 2

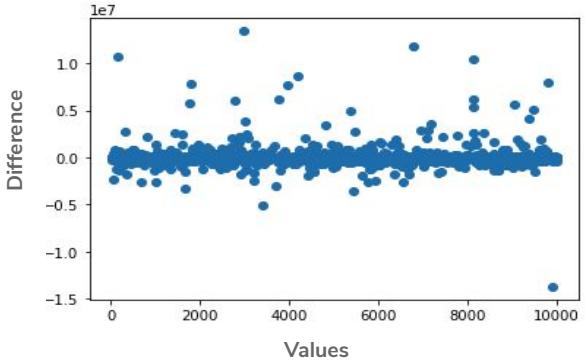
Select the Model which performs best Regression

Random Forest Regressor



INPUT	OUTPUT	INPUT	OUTPUT	FINAL ADVISOR LIST
Cleaned and Processed Data 70:30 as Train:Test split i.e. 70% of the data is used for Training the models & 30% for Test	List of Advisors who are expected to sell anything(i.e. Their expected sales > 0)	List of Advisors from the Classification Model	List of Advisors and their respective expected sales, which can be organized into Deciles(as per the sales amount)	Bucket and sort the Final Advisor List in Deciles as per the sales prediction for the respective advisor

MODEL PERFORMANCE



Regression Model Performance(for Forecasting sales)using RandomForestRegressor:

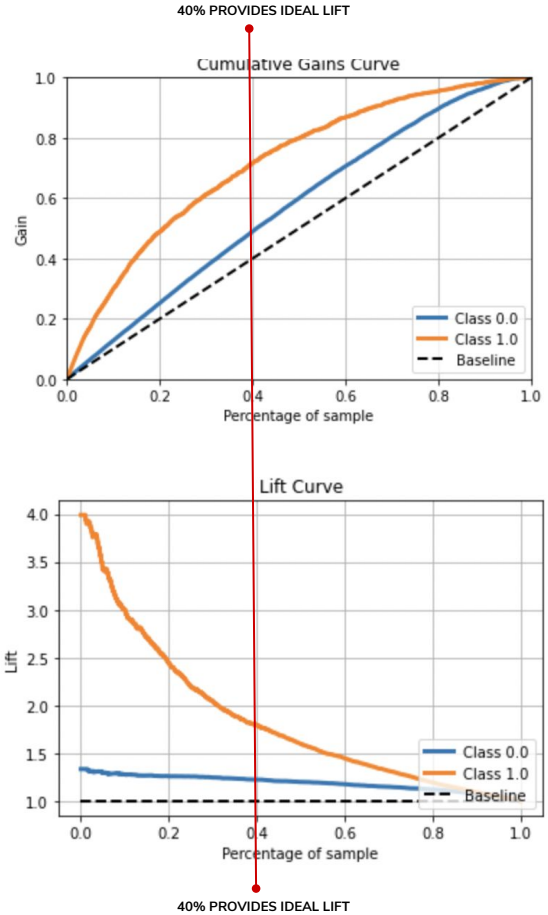
This a graph depicting the difference between the Predicted Values of sales and the Actual Sales. The Fact that the points are close to 0 in the X-axis indicates that the difference between prediction and actual values is nearly zero. This indicates that the regression model used to forecast next year's sales is quite accurate.

	precision	recall	f1-score	support
0.0	0.78	0.98	0.87	7484
1.0	0.76	0.19	0.30	2521
accuracy			0.78	10005
macro avg	0.77	0.58	0.59	10005
weighted avg	0.78	0.78	0.73	10005

Classification Model Performance(for Forecasting Advisors who will sell anything in the next year) using GradientBoostingClassifier:

This is a performance of the Final Classification Model which predicts which of the Advisors are most likely to add a new fund in the coming year. An overall weighted average f1 score of 0.73 indicates a robust model in classifying Advisors.

MODEL PERFORMANCE: Lift Curve



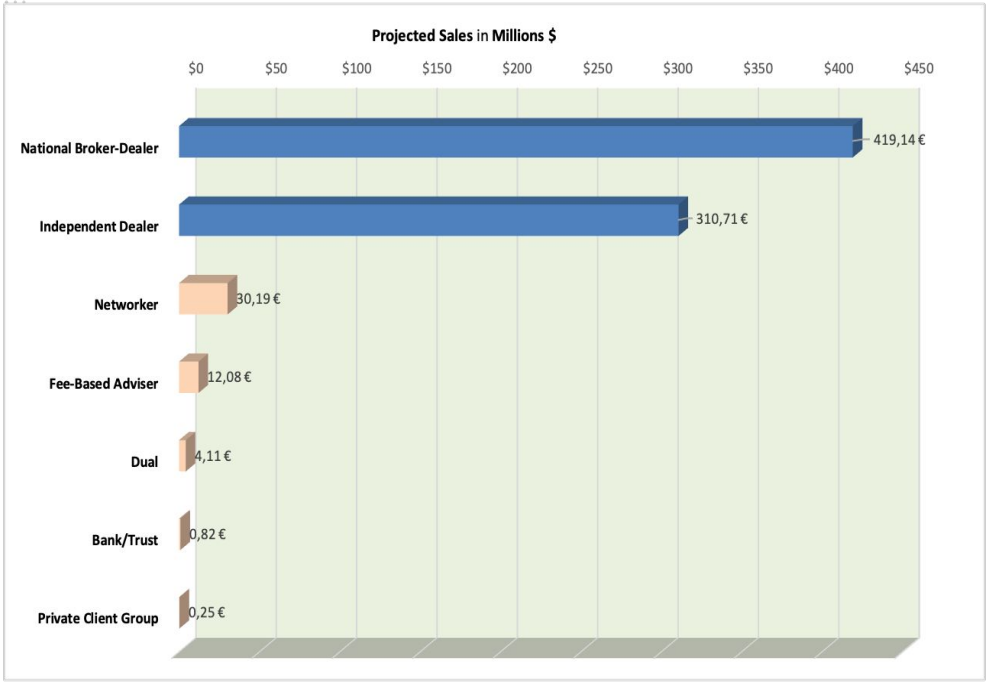
CUMULATIVE GAINS CURVE

This a graph depicting Cumulative Gains Curve, thus indicating the 'gain' in targeting a given percentage of the total number of Advisors using the highest modelled probabilities of performing(adding new fund), rather than targeting the Advisors at random. It can be observed that targeting the top 40% of the Advisors selected by the model brings maximum Gains(covers 45% of the projected sales).

LIFT CURVE

This is a graph depicting Lift Curve which indicates that we have a large lift associated with contacting a small proportion of Advisors. Here it can be observed that contacting as few as 40% of the total Advisors can bring about maximum Lift, hence can save marketing spend by not spending money on Marketing activities for the remaining 60% of the Advisors/Firms.

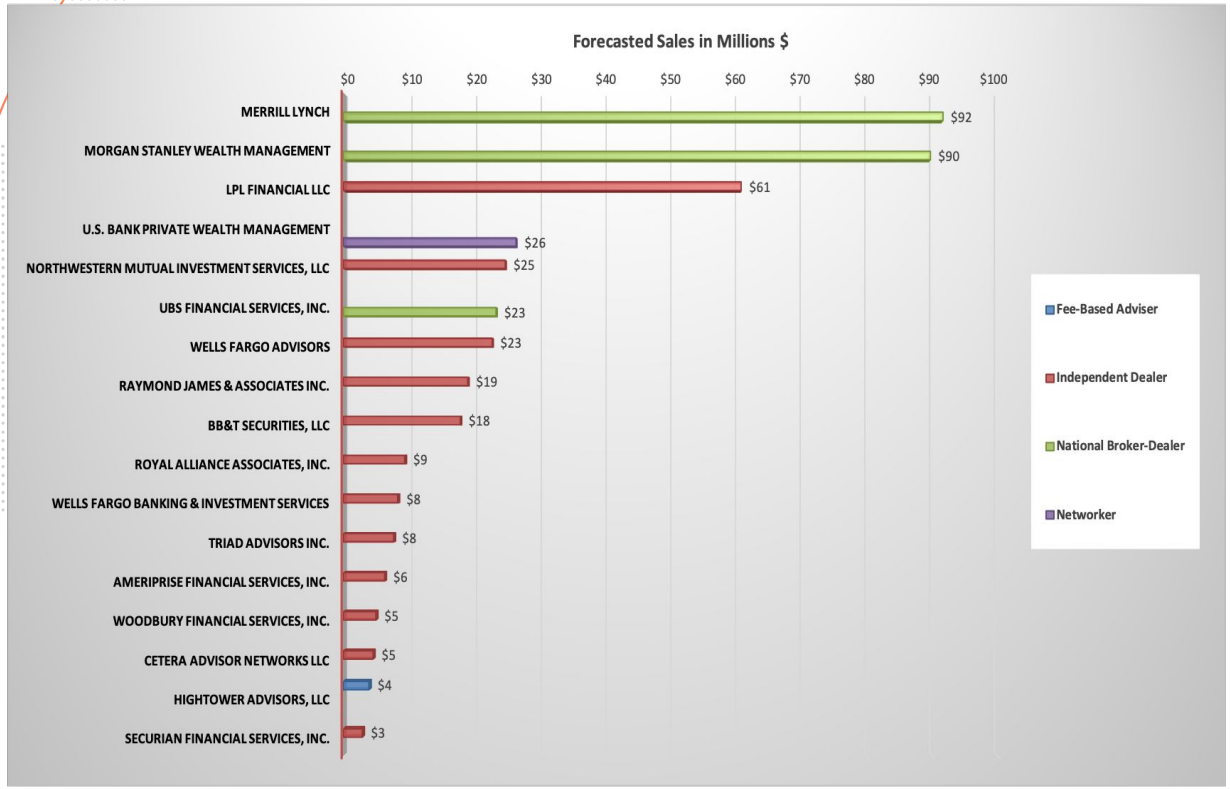
PROJECTED SALES PER CHANNEL



Sales Per Channel

- This Graph shows the Channel wise break-up of the projected sales.
- National Broker-Dealer & Independent Dealers account for around 93% of the projected sales

PROJECTED SALES PER ADVISOR



Sales Per Advisor

- This graph shows the projected sales for different advisors.
- Merrylynch, Morgan Stanley Wealth Management and UPL Financial LLC top the list and together account for 49% of the projected sales.

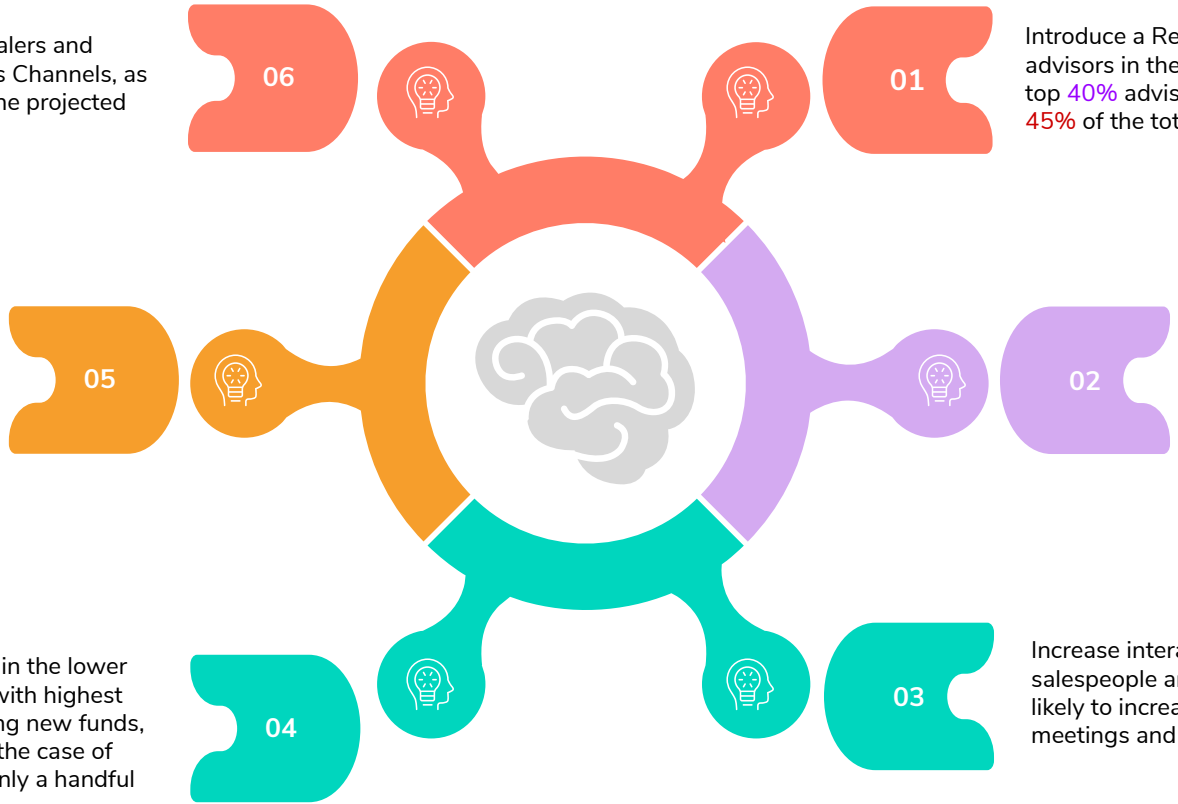
RECOMMENDATIONS

RECOMMENDATIONS

06 Focus on Independent Dealers and National Broker Dealers as Channels, as they account for 93% of the projected sales.

01 Introduce a Retention Strategy for advisors in the top deciles (1-4) i.e. the top 40% advisors , as they account for 45% of the total predicted sales.

05 Understand the factors of success of sales for Top decile Advisors and percolate these factors to the lower decile Advisors for more decentralized sales



02 Have a long term development strategy for the lower performing deciles 7, 8 and 9 to make the revenue streams more diverse.i.e. Decentralized sales

04 Focus on Advisors in the lower Deciles 7,8 and 9 with highest probability of adding new funds, as this will reduce the case of majority sales by only a handful of Advisors.

03 Increase interaction between Nuveen salespeople and advisors that are most likely to increase their sales. I.e. Set-up meetings and Workshops.

THANK YOU!

APPENDIX

CLASSIFICATION MODEL PERFORMANCE (Model Score)


```
In [70]: from sklearn.ensemble import GradientBoostingClassifier
```

click to scroll output; double click to hide

Select columns from RFE

```
In [71]: gbc = GradientBoostingClassifier()
         gbc.fit(X_train_prepared_cl, y_train_cl)

Out[71]: GradientBoostingClassifier()

In [72]: #look at the model score

In [73]: gbc.score(X_test_prepared_cl, y_test_cl)

Out[73]: 0.7531645569620253
```

FEATURES USED FOR REGRESSION

In [90]:
```
# Create and fit SelectKBest selector
selector = SelectKBest(f_regression, k=11)
selector.fit(X_train, y_train_reg)
# Get columns to keep and create new dataframe with those columns
cols = selector.get_support(indices=True)
features_df_new = X_train.iloc[:,cols]
```

In [91]:
```
features_df_new.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7003 entries, 1666 to 899
Data columns (total 11 columns):
 #   Column                                Non-Null Count  Dtype  
---  ------                                --------------  -----  
 0   no_of_sales_12M_1                     7003 non-null   float64
 1   no_of_funds_sold_12M_1                7003 non-null   float64
 2   no_of_funds_redeemed_12M_1            7003 non-null   float64
 3   no_of_funds_Redemption_12M_10K        7003 non-null   float64
 4   no_of_assetclass_sold_12M_1           7003 non-null   float64
 5   no_of_assetclass_Redemption_12M_10K   7003 non-null   float64
 6   No_of_fund_curr                       7003 non-null   float64
 7   AUM                                   7003 non-null   float64
 8   sales_curr                            7003 non-null   float64
 9   sales_12M                             7003 non-null   float64
 10  redemption_curr                       7003 non-null   float64
dtypes: float64(11)
memory usage: 656.5 KB
```

REGRESSION MODEL PERFORMANCE
Linear Regression

```
In [123]: lr_pipeline.fit(df_X_train_selectKbest, y_train_reg)

Out[123]: Pipeline(steps=[('pca', PCA(n_components=0.8)),
                          ('linearregression', LinearRegression())])

In [124]: lr_pipeline.score(df_X_train_selectKbest, y_train_reg)

Out[124]: 0.5147333090226286

In [125]: lr_pipeline.score(df_X_test_selectKbest, y_test_reg)

Out[125]: 0.4030098356041324
```

REGRESSION MODEL PERFORMANCE
Random Forest Regressor

```
In [128]: rf2 = RandomForestRegressor()

In [129]: rf2.fit(df_X_train_selectKbest, y_train_reg)

Out[129]: RandomForestRegressor()

In [131]: rf2.score(df_X_train_selectKbest, y_train_reg)

Out[131]: 0.8267920606771533

In [130]: rf2.score(df_X_test_selectKbest, y_test_reg)

Out[130]: 0.36279801301425807
```

Final Combined MODEL PERFORMANCE

In [91]:
```
from sklearn.metrics import classification_report
```

In [92]:
```
def final_classification(cl_model, reg_model, X_cl, X_reg, y_cl, y_reg):
    results = pd.DataFrame(index=y_cl.index)
    results['cl_actuals'] = y_cl
    results['reg_actuals'] = y_reg

    # Create predictions for classification on X_cl
    results['cl_preds'] = cl_model.predict(X_cl)
    results.loc[results['cl_preds'] == 0, 'reg_preds'] = 0

    # create predictions for advisors that classifier predicts as 1
    results.loc[results['reg_preds'].isnull(), 'reg_preds'] = reg_model.predict(X_reg.loc[results['reg_preds'].isnull()

    return results
```

In [93]:
```
first_model = final_classification(gbc, rf2, X_train_prepared_cl, X_train_prepared, y_train_cl, y_train_reg)
```

In [94]:
```
second_model = final_classification(gbc, rf2, X_test_prepared_cl, X_test_prepared, y_test_cl, y_test_reg)
```

In [95]:
```
final_model_dataframe = pd.concat([first_model, second_model])
```

In [96]:
```
print(classification_report(final_model_dataframe['cl_actuals'], final_model_dataframe['cl_preds']))
```

```
              precision    recall  f1-score   support

         0.0       0.78      0.98      0.87      7484
         1.0       0.76      0.19      0.30      2521

    accuracy                           0.78     10005
   macro avg       0.77      0.58      0.59     10005
weighted avg       0.78      0.78      0.73     10005
```

THANK YOU!
...Finally!!

-Lorem Ipsum-