**Data Analysis and Visualization on real-world Dataset**

**GitHub -** [**https://github.com/daraghmcardle23/UCDPA\_daraghmcardle.git**](https://github.com/daraghmcardle23/UCDPA_daraghmcardle.git)

***Case study*** –

ETFs are an accessible alternative for retail investors compared to mutual funds which are generally only available to institutional investors. Do they represent a better investment opportunity overall? As part of the project we will be aiming to provide insight into the main points below -

* How do the returns of ETFs compare against the returns of actively managed Mutual funds?
* What ratio’s are important to consider when investing in ETFs and Mutual funds?
* Using machine learning we will try to predict future stocks prices based off previous days prices.

Additional information to help understand why we are comparing ETFs against Mutual funds –

*What is a mutual fund?* A mutual fund is a type of financial vehicle made up of a pool of money collected from many investors to invest in securities like stocks, bonds, money market instruments, and other assets. Mutual funds are operated by professional money managers, this allows the investments to be actively managed on a day to day basis.

*What is an exchange traded fund(ETF)?* An ETF is a type of security that tracks an index, sector, commodity, or other asset, but which can be purchased or sold on a stock exchange the same as a regular stock.

***Data Source –***

ETF csv file contains 1,680 ETFs and Mutual Fund file contains 24,821 mutual funds with columns ranging from top 10 holdings, portfolio holdings(stocks, bonds, sectors) to financial ratios(beta, Sharpe Ration and price/earnings). Files were sourced from Kaggle, [here](https://www.kaggle.com/stefanoleone992/mutual-funds-and-etfs), with all data being scraped from Yahoo Finance. The dataset contains information ranging from individual MF returns to categorical returns from 2011 to 2020.

***Project code and analysis of output -***

Reading in Data – contained in ‘Part 1 – final project’

Pandas package was used to read in the main ETF and Mutual funds data sets. ‘*Low\_memory = False*’ is included due to the large number of columns, I wanted my project to run as efficiently as possible and this allowed PyCharm to read in the files faster. Used *‘print(etf\_df.shape)’* as a check to ensure the CSV was being read in correctly from the windows folder on my laptop.

Main dataframes throughout the project are –

, etf\_df = contains all exchange traded fund information

, mf\_df = contains all mutual fund information

Text

Description automatically generated

Netflix is regarded as the highest returning stock from 2011-20, we us the Yfinance API to pull NFLX’s opening, adjusted and closing prices on a monthly basis across this date period. I had some issues when running the regular expression function on the dataframe so used an underscore in the columns to help with pulling the correct column. I also used dropna to remove any null values and included a print function as a check that outputs how many rows with at least one NA value have been removed

A computer screen capture

Description automatically generated with low confidence

On a separate script ‘Part3 – Neural Network’ we use a sub package called pandas\_datareader which allows you to create a similar Netflix dataframe from different internet sources. As part of this project we used the Yahoo Finance API with this package to read in the stock price of Netflix(NFLX) from 2011-2020. This dataframe will be used as the last step of the project to create a model that predicts the stock price on the next business day.

Text

Description automatically generated

Data Cleaning –

As part of our preliminary data cleaning, we remove all null and unused columns from our dataframes. We use a Clean\_by\_keyword method that allows us to break down the large data sets and sort based off their keywords in column names. The method is used across multiple columns on both datasets, standard deviation, ratios, sectors etc.

Text

Description automatically generated

Roughly half of the columns contain categorical data, that is incomplete or not relevant to our analysis so it makes sense to remove them from the main dataframe. We do use the YTD,3,5, and 10 year returns column later so these are re-added for analysis.

Text

Description automatically generated

A left join is used to merge the statistical columns that could be useful when looking deeper into each dataset later in the project.

Text

Description automatically generated

Additional functions used as part of initial data cleaning –

Text

Description automatically generated

Function that divides each column by 100 to convert it to a percentage rather than a whole number for later analysis.

Advanced Data Cleaning –

Treynor ratios are recorded as strings so we create a string to float conversion method which converts the specific columns from str to float. For every None-type in the specified columns they are assigned a value of -999.

Text

Description automatically generated

Method below, looks for all values above -990 and calculates the mean and replaces any values below -990 with the calculated mean.

Text

Description automatically generated

Regular Expression –

The etf\_df dataframe has a top10 holdings columns, which contains what % has been allocated to different companies for each ETF. As mentioned above, NFLX has the highest returns of stocks since 2010 so I thought it would be interesting to extract how many ETF’s hold NFLX as one of their top 10 investments and the % allocated.

Graphical user interface, application, table, Excel

Description automatically generated

Used re.finall to return all non-overlapping matches of the pattern in the specified column. Matches are returned in the order found. This method return a list of groups. Of the 1680 ETFs in the report only 18 of them hold NFLX as one of their top 10 holdings with 4.94% being the biggest allocation.

A picture containing text, screenshot, indoor

Description automatically generated

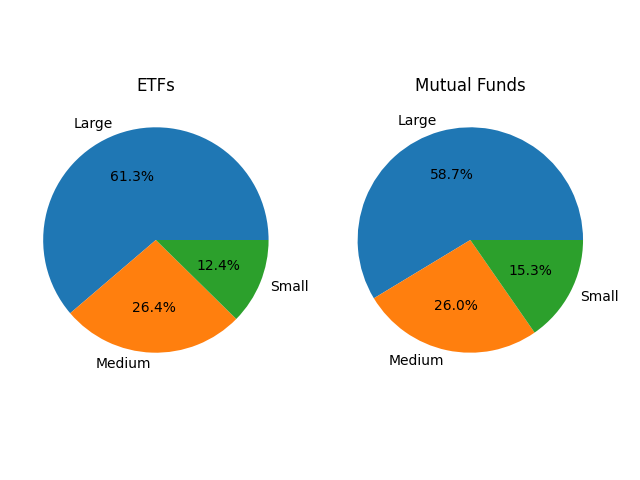
**Exploratory data analysis** –

To visualise how our data looks after cleaning and give some insight into what size and investment types across both our dataframes.

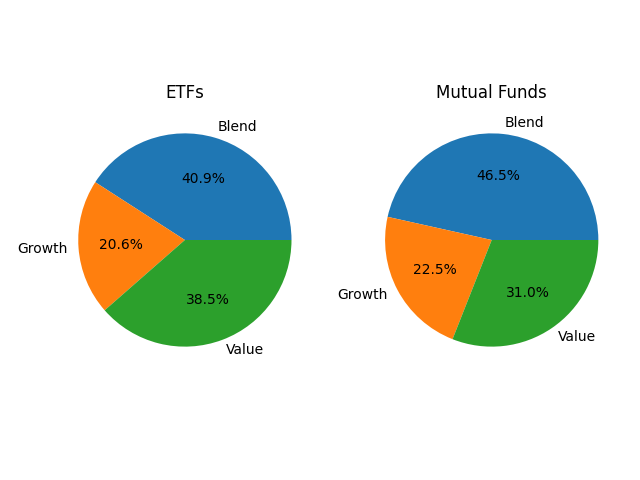
Pie charts – created a function that produces two pie charts –

Text

Description automatically generated



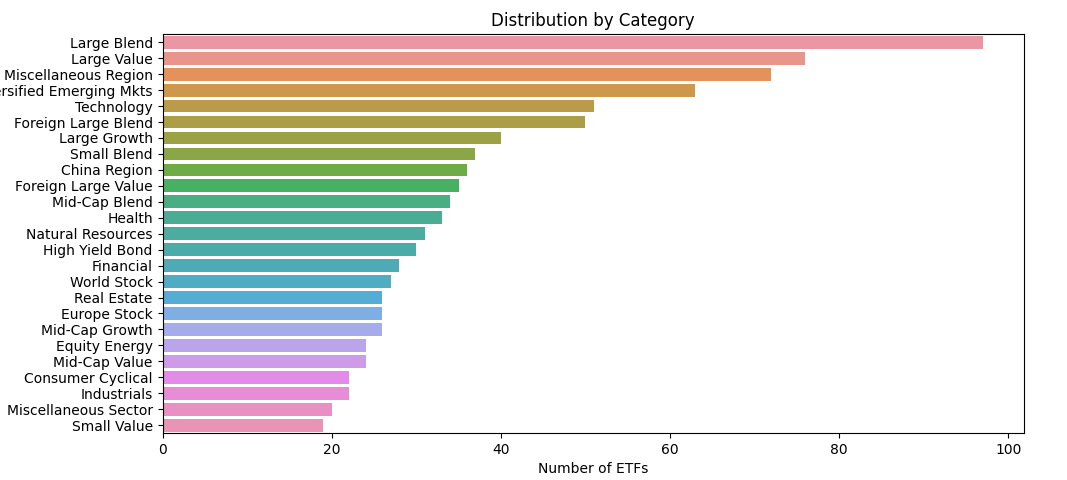
Large ETF and MFs make up over 50% of our datasets. With Blend being the investment type with the highest allocation across both MF and ETFs in the pie charts below.



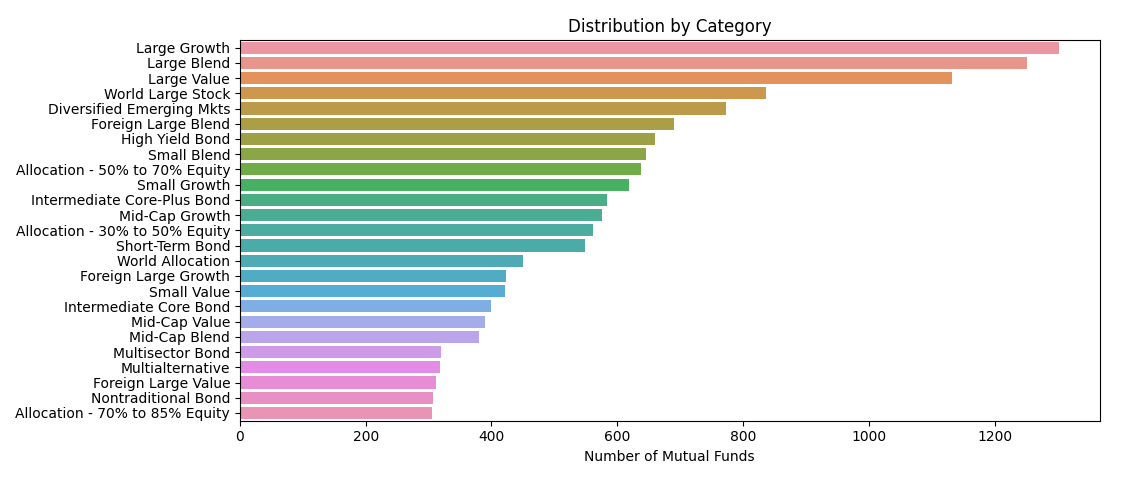
Generate a bar plot using seaborn as SNS.

Text

Description automatically generated



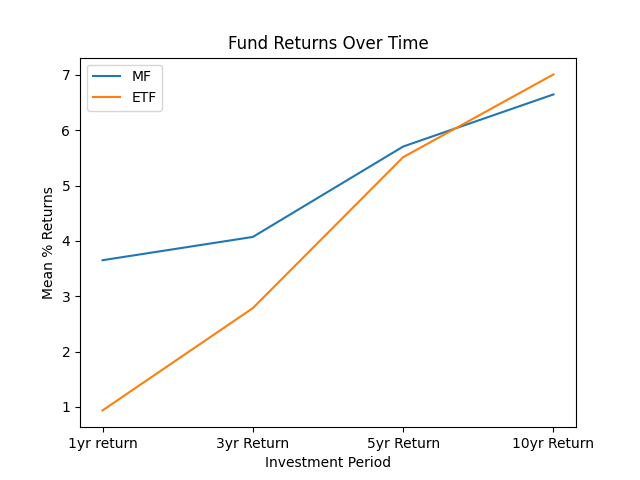
ETFs highest allocation is in large blend category with small value being the lowest allocation. MF lowest allocation is in 70-85% equity funds which is surprising because MFs would typically hold a large % of equity in their portfolio.



***Analysis, results and conclusion -***

**Time Frame Analysis -**

One of the toughest questions when investing in a fund/etf/stock is knowing how long to hold that specific position. In this section we will look at the ideal length of time to hold a position for each fund type.

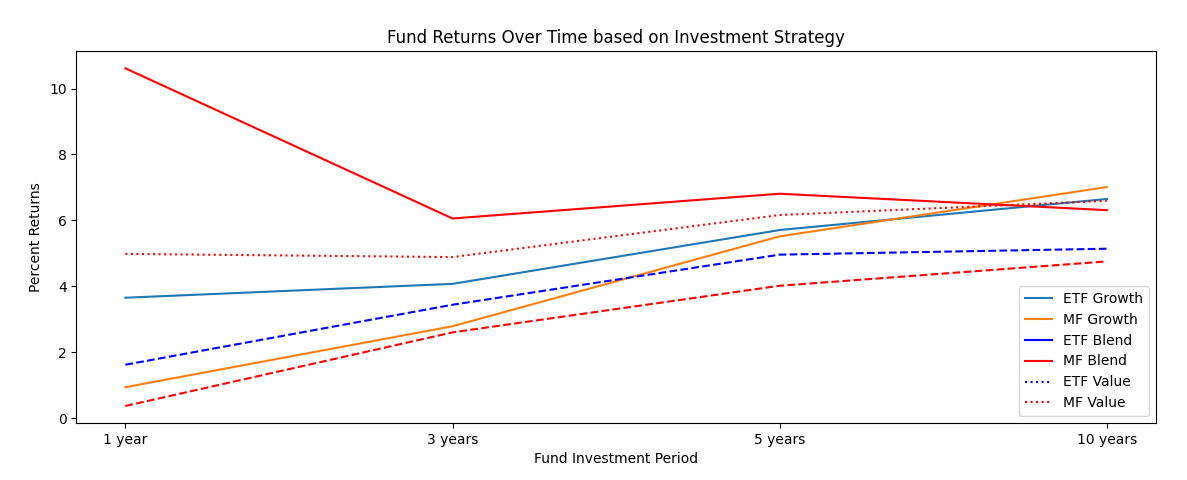


From the graph we can see that Mutual funds have a greater return for the first 3 years of holding a position. In my opinion this is due to the active management style of mutual funds which allows portfolio managers to react quickly to fortunate or unfortunate market conditions.

After 5 years, the ETF fund investments starts to eclipse Mutual fund returns and from 5 to 10 years the ETF returns are greater than that of mutual funds.

**Investment type analysis –**

The 3 main types of funds in our datasets are growth(high risk rating), value(generally cheaper) and blend(a mixture of both). How do returns over time compare between these fund investing styles?



From the graph, for a short term investor(<1 year) we can see that investing in a Mutual fund blended investment type would be highly recommended as yearly returns are on >10%. As a medium to long term investor(3-5 years) the recommendation would be to invest in MF Blend, MF Value and ETF Growth.

In comparison, value funds start off with a return of <2% and gradually increase after 1 year and cross above the 2% return threshold at about 2 years. Value funds have smaller returns because they pay their shareholders a higher dividend compared to blend funds, who pay a small dividend, and Growth funds who usually pay close to no dividends.

Script 2 – **‘part 2 – random Forest’**

**Random forest regressors(RFR) –**

We will use RFR to find what ratios are important to consider when investing in both MF and ETFs, using two different dependent variables, ‘fund return ytd’ and ‘fund return 10years’ to analyse which ratios are important on a short term and long term basis and comment on any differences.

We use Ordinary least squares regression model with multiple predictors, across the 4 models, setting our confidence level as 0.05. Earlier in the project we used our clean\_by\_keyword function to create a new dataframe that contains all of the ratio columns from our main ETF and MF dataframes. These will be used for the RFR.

Text

Description automatically generated

Model 1 results – Mutual fund ‘fund\_return\_ytd’

print(result1.summary())

A picture containing text

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

Model1 discussion – R-squared value of 0.790(79%) shows that our regression line approximates quite well to the real data points. Fund ‘sharpe\_ratio\_3\_years’ has a coefficient value of 28.8126 which would be expected because an increase of 1 in the Sharpe ratio would increase the returns YTD significantly.

From the table we can see that the ‘fund\_sharpe\_ratio\_3years’ is the ratio that effects fund return ytd the most and is highly positively correlated and has a p-value of 0 meaning our test is statistically significant. The t value for the fund\_sharpe\_ratio\_3years is also the largest (54.236) which indicates that the co-efficient is significant.

From our model we can conclude that the most significant ratio to consider before investing in a MF when analysing the fund return ytd is the Sharpe ratio over the previous 3 years.

Model 2 results – ETF ‘fund\_return\_ytd’

print(result2.summary())

Graphical user interface

Description automatically generated with low confidence

A picture containing text, computer

Description automatically generated

Model2 discussion – R-squared value of 0.582(58%) shows that our regression line approximates relatively well to the real data points but it is not as good a fit as the MF version. From our table above we can see that the 3year Treynor ratio is an important feature to consider and it positively correlated. With a p-value under our confidence level 0.05 we can conclude that the Treynor ratio has statistically significant relationship with fund return YTD for ETFs.

Price\_book\_ratio(p\_value=0.4894), Price\_sales\_ratio(p\_value=0.0629) and fund\_sharpe\_ratio\_3years(p\_value = 0.4578) do not have a statistically significant relationship with fund return YTD.

Using the same models we will now compare what ratios are significant for fund returns over a 10 year period while also comparing them against the fund return YTD results.

Model 3 – MF 10year return

Text

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with low confidence

R-Squared = 0.769

Fund Sharpe Ratio 10 years, t value = 82.006.

Fund Treynor Ratio years, t value = 21.520.

In comparison to the YTD return results, our R-squared value decreased marginally showing our model is not as accurate. The Sharpe Ratio for 10years has a large T-value, showing that the co-efficient is significant.

From our table we can see the 10 year Treynor ratio is the most important feature according to our model and is positively correlated with a p-value of 0. The 10 year Sharpe ratio is also statistically significant based off our model.

Model 4 – ETF 10 year return

Text

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

R-Squared = 0.878

Fund Sharpe Ratio 10 years, t value = 13.032

The R-squared value of 0.878 is the highest value across the four summaries showing this model is the most accurate of the 4. The 10 year Sharpe ratio has the highest coef and t value as well as having the highest correlation to the 10 year return dependent variable.

In conclusion, from our four models above we can see that the Sharpe and Treynor Ratios are statistically significant when investing in MF or ETFs on a short or long term basis. On a long term basis my recommendation from the models above but also the timeframe and investment type analysis graphs would be to invest in a blended ETF for at least 10 years as the returns are consistent and the risk of minimal returns is quite low.

Script 3 – ‘**Part 3 – Neural Networks**’

**Neural Networks Stock Predictor –**

For the last section of the project we will use a LSTM recurrent neural network model to predict future stock prices. LSTM networks use memory blocks that are connected through layers. LSTMS are sensitive to the scale of the input data so we rescale the data to the range of 0-1.

scaler = MinMaxScaler(feature\_range=(0,1))  
scaled\_data = scaler.fit\_transform(data['Close'].values.reshape(-1,1))

Can see from the graph that there is an upward trend for NFLX stocks prices from 2011-2020.

Chart, line chart

Description automatically generated

Run1 -

Initial parameters on the model, Epochs = 5, batch size =32.

Chart, line chart, histogram

Description automatically generated

Can see that our model is not accurate and is severely under predicting the NFLX stocks price for the duration of the test. The number of epochs can be increased to increase the amount of learning that the model does and in theory increase in accuracy. We will test this in our next run.

Run 2 – Epochs = 25, batch\_size=40

Chart, histogram

Description automatically generated

The increase in number of Epochs and batch size resulted in the red line being a lot closer to the actual NFLX price but like the previous run the predicted NFLX stock price is less than the actual NFLX price. We can see that our predicted price has significant increase like the actual price around the 150 to 200 day mark which shows that our model is learning the significant increases in prices. In my opinion it is better to have a model that is under predicting rather than over predicting the price.

Run 3 – Epochs 40, batch size = 40

Chart, histogram

Description automatically generated

15 additional epochs and same batch size compared to run2 but our model has not significantly increased in accuracy but had a long run time in python which is not overly efficient. Our predicted price trends along the actual price and is accurate by picking up the decrease in price(50 days) and increases from 120-200 days but I would prefer if the predicted price trended above the actual price as this would show that the model is not under estimating the share price for the duration of the run. Overall I feel that the model is accurate and with some additional tuning the predicted price could trend closer to the actual price.

Using the LSTM model, we tried to predict a stock price for the next business day based off what the model has learnt from the previous runs.

Text

Description automatically generated

Predicted price for 4th of January 2021

Predicted price: 493.67874