Leveraging Analytical Tools for Barratt Development PLC's Strategic Investment Decision

Chuks Chiazor

April 2022

Abstract

Barratt Development Plc, established in 1958, is one of the world's largest residential property development firms. Since 1968, the company has been listed on the London Stock Exchange, and it is now a major component of the FTSE 100 index. The aim of this report is to provide sufficient advice on Barratt as a good company for investment based on existing data and to provide prospective investors with an understanding of the company's overall financial performance. To accomplish this, descriptive, time series, and fundamental analytics approach were carried out. From the analysis, Barratt has a higher mean return and standard deviation. This however implies that Barratt has both the higher return and volatility/risk. The regression line had a good fit between Barratt and Berkeley Returns. The time series forecasting analysis suggests Simple Exponential Smoothing method as the most suitable forecasting model due to its lower Moving Average Error and highest (four) forecast line points matching with the test price line (ground truth or expectation). From 2016 to 2020, Barratt's financial report indicates a progressive rise in revenue, gross profit, total assets, equity, and decreased liability. The Covid-19 pandemic had a negative impact on the company's financial results, but it recovered quickly and remains a market leader in the UK.

1 Introduction

This report is an analysis of Barratt Development Plc, one of the world's largest residential property development companies in the UK founded in 1958. The organisation has been listed on the LSE since 1968 and today is an important constituent of FTSE 100 index. The organisation has two national brands Barratt Homes and David Wilson Homes aimed at house building. In London, the organisation operates as Barratt London. Furthermore, the company also has a commercial property development business Wilson Bowden Development whose focus in on retail, industrial and office purposes (Barratt Development, 2020). The aim of this report is to:

- Compare Barratt development Plc's stock output to that of its competitors and to understand market trends.
- Promote financial market analysis using the Pearson correlation technique to calculate the linear relationship between Barratt development Plc, the competitor, and the market.
- Apply time series analysis, forecasting, and exponential smoothing techniques (Single Exponential Smoothing, Double Exponential Smoothing, and Winter's method) to make investment recommendations to investors for Barratt Development PLC based on a comprehensive and data-driven evaluation.

1.1 Contribution of this Paper

This paper focuses on the use of several analytical techniques to compare Barratt's financial performance with competitors, utilizing descriptive analysis, Pearson correlation for market analysis, and applying time series and exponential smoothing forecasting techniques for investment insights and fundamental analysis.

1.2 Outline of this Paper

This paper provides a comprehensive exploration encompassing several analytical methodologies. Firstly, it delves into descriptive analysis, providing a foundational understanding of Barratt's sales. Following this, the paper explores correlation and regression analysis, offering insights into the relationships and predictive dynamics between key financial markets. Subsequently, time series analysis is conducted to examine trends and patterns over time, enhancing the depth of the analysis. Finally, the paper culminates with a fundamental analysis, leading to informed and data-driven recommendations.

2 Preliminaries on Python Ecosystem

No prior knowledge of Python is required to use the material in this paper. However, we assume that the reader/instructor who wants to use the tools presented here has Python up and running on their device (desktop, laptop, etc.) The codes and corresponding results are based the use of Python under Anaconda 3 with Jupyter as the editor, all running on Windows 10 Enterprise (processor: Intel(R) Core(TM) i5-6300U CPU @ 2.40 GHz). The advantage of using Anaconda is that it installs Python with many important packages that are useful for time series analysis of the type covered in this paper. This therefore helps in part to reduce dependency issues between various packages used, and hence ensure that key packages are set to work nicely together. Nevertheless, all the codes presented here should be able to work smoothly on most platforms running version 3 of Python (see https://www.python.org/). The main packages needed are as follows:

- Numpy;
- SciPy;
- Matplotlib;
- Pandas
- Statsmodels

SciPy, an ecosystem of open-source software for mathematics, science, and engineering (available at `https://scipy.org/`), incorporates the NumPy library (`https://numpy.org/`) for operations on multi-dimensional arrays, and the Matplotlib library (`https://matplotlib.org/`), tailored for plotting functions.

The paper also extensively utilizes Pandas (`https://pandas.pydata.org/`), an open-source tool for data analysis and manipulation. Furthermore, the statsmodels library (`https://www.statsmodels.org/stable/index.html`), forms the backbone of our data analysis and forecasting methodologies, offering packages for exponential smoothing, ARIMA, and regression-based methods. In certain instances, the scikit-learn library may be employed for specific tasks, such as calculating error measures.

It is crucial to highlight the synergistic relationship between these libraries: NumPy and Matplotlib extend the functionalities of SciPy, while Statsmodels, built upon NumPy and SciPy, seamlessly integrates with Pandas for efficient data management.

3 Technical Analysis of Stock Market

According to Audalovic et.al. (2017), Stock markets are critical in bridging the gap between businesses in need of capital to launch new projects or expand existing operations and investors with excess funds to invest in those businesses. Hunger and Wheelen (2011) go on to say that financial analysis, macro environmental analysis, and stock market performance analysis can all be included in an organization's strategy growth efforts. This report will focus on conducting Descriptive analysis and Fundamental analysis through stock performance analysis to help arrive at recommendations. According to Kuttner (2012), the housing industry in the United Kingdom is a significant determinant of economic stability since it booms when the economy and consumer wealth are relatively large. Berkeley Plc will be used as a major competitor in this report. Berkeley is a serious player in the real estate market. The Berkeley Group has offices in London, Birmingham, and the South of England (Berkeley Group, 2021)

3.1 Descriptive Analysis

The Descriptive Analysis section in your report on Barratt Development PLC is designed to provide a foundational understanding of the company's financial performance using various statistical tools. The descriptive analysis aims to provide a clear overview of Barratt's stock performance over time. By utilizing visual representations like line plots, box plots, and histograms, you can effectively convey the behavior of stock prices and returns, as well as compare Barratt's performance with its competitors and market indices. This analysis is crucial for identifying underlying trends and patterns that might not be apparent through mere observation of raw data. The incorporation of statistical summaries, such as mean, median, standard deviation, and quartiles, further enriches the understanding of the data's distribution and variability. These statistics offer insights into the volatility and risk associated with Barratt's stock, forming a basis for more complex analyses and investment decisions.

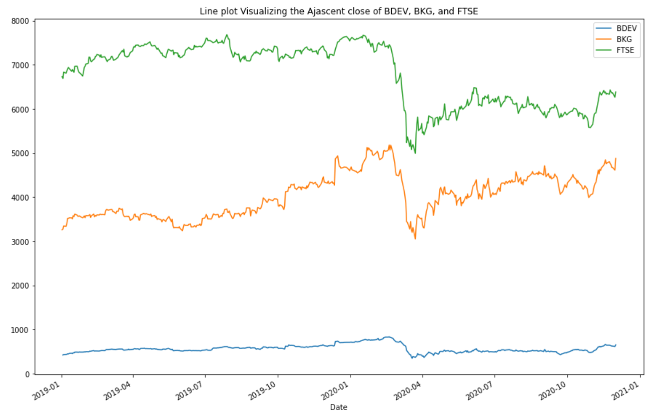

Figure 1: Line plot showing the Adjacent close of Barratt Development, Berkeley, and FTSE

The above figure depicts the adjacent close (price) made by the companies. Similar trends are observed across the industry with respect to Barratt its competitor and the market. It can be observed that Barratt Plc, Berkeley Plc and FTSE was stable until they experienced a fall in price around February 2020 and hit their lowest mark in March 2020.

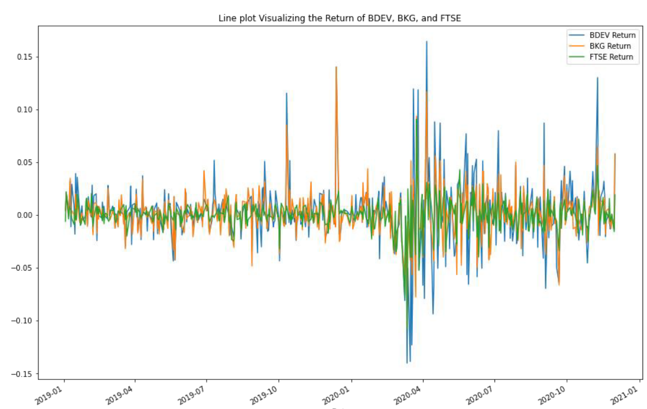

Figure 2: Line plot showing the Return of Barratt Development, Berkeley, and FTSE.

From what we have above, Barratt Plc has both highest positive and negative return. Barratt has its peak at 0.15 in May 2020 and least around March 2020.

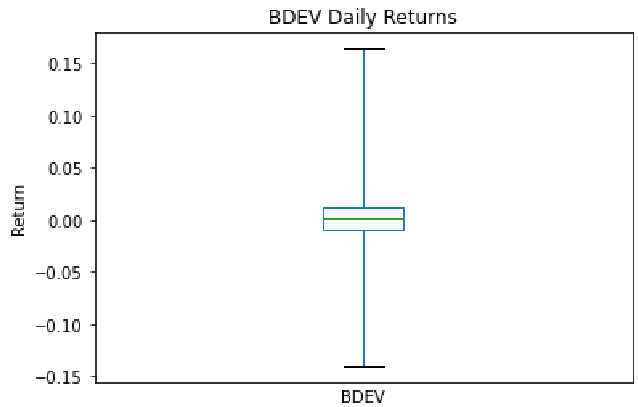

Figure 3: Box plot visualizing Barratt Plc daily returns

We can immediately make a few key observations from the plot above:

1. The minimum number of returns is around -0.14 (min), maximum number is around 0.16 (max), and median number of returns is around 0.001 (median).

1. 25% of the years for the period 2019-01-02 to 2021-01-01 had returns of ∼ -0.009836 or fewer (First quartile).

1. 75% of the years for period 2019-01-02 to 2021-01-01 had returns of∼ 0.012400 fewer (Third quartile).

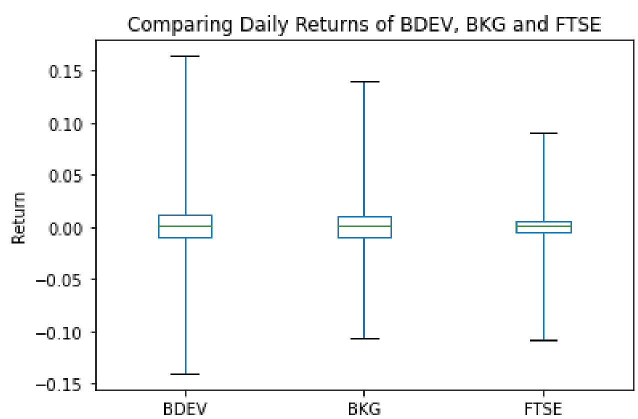

Figure 4: Comparing daily returns of Barratt, Berkeley, and FTSE

We can observe that, while Barratt Plc, Berkeley Plc and FTSE have around the same median return (0.001), Barratt return range is most spread out than

Berkeley and FTSE. The maximum return for Berkeley (0.14) is higher than the total max return of FTSE (0.09). However, this further shows that Barratt has the highest volatility, followed by Berkeley.

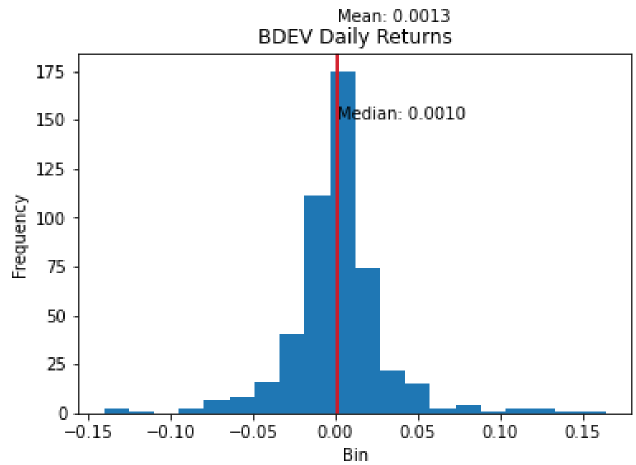

Figure 5: Histogram showing the daily returns of Barratt Plc

The above figure depicts the daily return mean and median of Barratt Plc as 0.0013 approx. and 0.0010 respectively which means that Barratt daily returns is negatively skewed. This however suggests that Barratt has higher volatility which makes them risky for business.

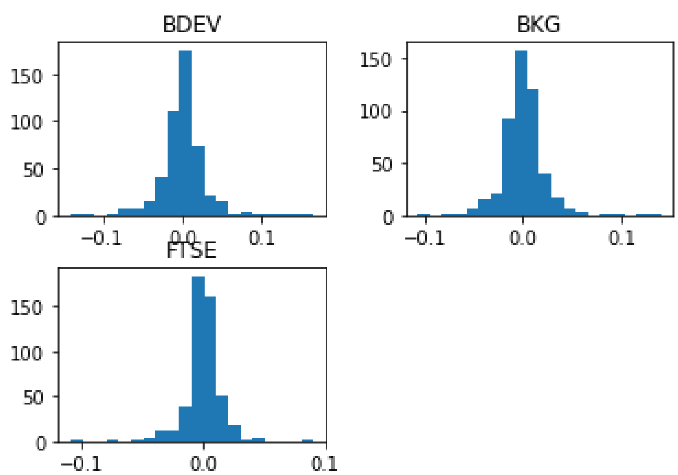

Figure 6: Histogram showing the daily returns of Barratt, Berkeley, and FTSE

Comparing BDEV to BKG and FTSE from the above figure, we can imply that BDEV is not close in value which suggests that the data is not near symmetry.

3.2 Statistical Table

```
BDEVtr.describe()
```

	Adj Close	Return
count	486.000000	486.000000
mean	562.108915	0.001342
std	88.830349	0.029828
min	346.101532	-0.140422
25%	509.069061	-0.010090
50%	540.697418	0.001023
75%	602.774185	0.012400
max	833.604492	0.164195

```
BKGtr.describe()
```

	Adj Close	Return
count	487.000000	486.000000
mean	4028.715394	0.001061
std	482.598878	0.021676
min	3052.493408	-0.106722
25%	3596.847778	-0.009836
50%	4026.444824	0.000815
75%	4366.522461	0.010074
max	5182.437988	0.139880

```
FTSEtr.describe()
```

	Adj Close	Return
count	487.000000	486.000000
mean	6785.624433	-0.000007
std	683.928666	0.014241
min	4993.899902	-0.108738
25%	6104.800049	-0.005494
50%	7151.100098	0.000682
75%	7353.449951	0.006470
max	7686.600098	0.090530

Figure 7: Barratt, Berkeley, and FTSE Descriptive Statistical summary

Volatility, according to Kuepper (2021), is a statistical measure of the dispersion of returns for a given security or market index and is often expressed as the standard deviation or difference between returns from the same security or market index. Hayes (2020) went on to define financial return as the amount of money made or lost on an investment over time. Comparing the above statistical table above, Barratt has the highest mean return of 0.001342 followed by Berkeley Plc of 0.001061, and FTSE with the least mean return of -0.000007. Barratt also shows to have the highest standard deviation of 0.0298 followed by Berkeley, 0.02167 and FTSE, 0.01424. This however implies that Barratt has both the highest return and volatility/risk. According to our findings we can deduce that Barratt is considered a very good company to invest but at the same time tends to be risky because of its volatility/risk rate.

4 Correlation and Regression Analysis

When it comes to investing in the financial markets, the correlation between two variables is particularly useful. Investors may use correlation statistics to see whether the correlation between two variables changes. (Fernando, 2021). The Pearson correlation method will be used to assess the linear relationship between Barratt Development Plc, Berkeley Plc, and the FTSE in this report.

A scatter plot could be used to visualize the relationship and suggests the forecast of prices. A suitable model should include BDEV and BKG as an explanatory variable. The scatter plot function 'scatter' function from matplotlib is used with arguments being BDEV and BKG and price as separate entries. Would be good to note that pandas also has the function 'scatter_matrix', which can generate scatter plots for many variables in one go.

The statistic corresponding to a number between -1 and 1 for the measurement of linear relationship for bivariate data (i.e when there are tow variables) is known as Correlation, which has the 'corrcoef' function from numpy.

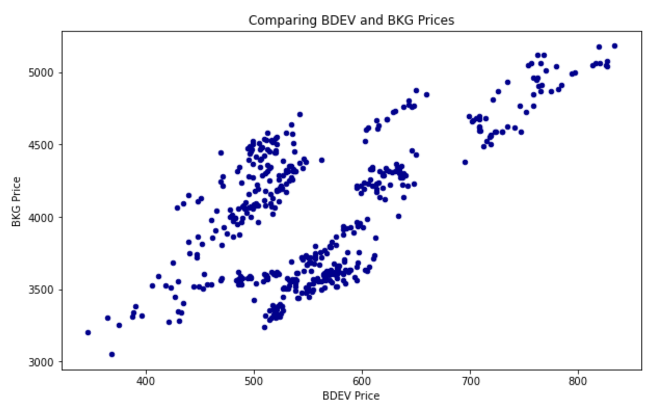

Figure 8: Scatter plot showing Barratt and Berkeley Price relationship

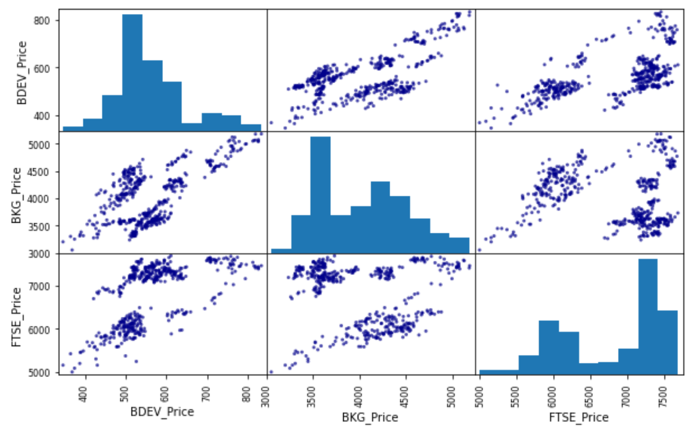

Figure 9: Correlation matrix of Barratt, Berkeley, and FTSE Price

From the figure above, we observe that the prices of Barratt and Berkeley are loosely or randomly scattered about. So, we say there is little / no linear relationship between the two variables.

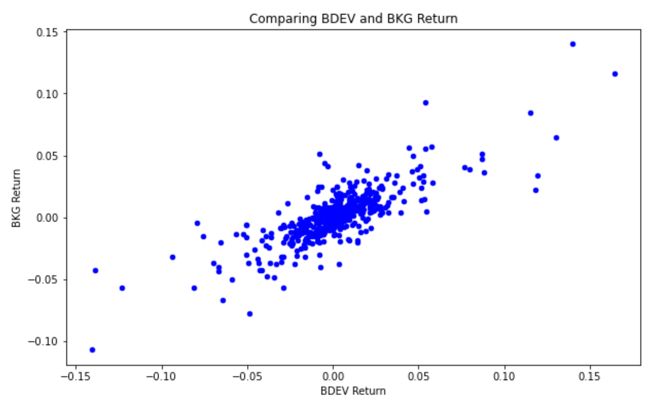

Figure 10: Scatter plot showing Barratt and Berkeley Returns relationship

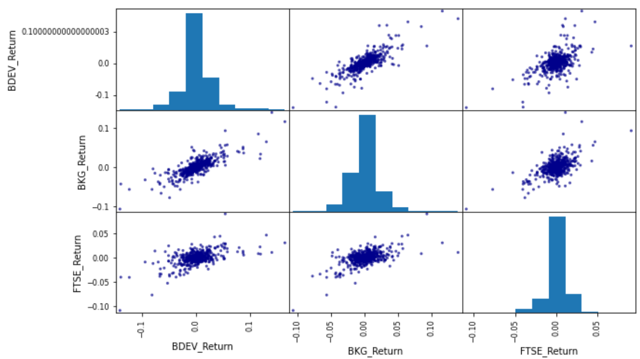

Figure 11: Correlation matrix of Barratt, Berkeley, and FTSE Return

We can observe that the circles on the scatterplot are reasonably closely scattered about an underlying straight line (as opposed to a random scattering), however, we can say there is a stronger linear relationship between the two variables.

	BDEV_Price	BKG_Price	FTSE_Price
BDEV_Price	1.000000	0.633497	0.601156
BKG_Price	0.633497	1.000000	-0.100745
FTSE_Price	0.601156	-0.100745	1.000000

	BDEV_Return	BKG_Return	FTSE_Return
BDEV_Return	1.000000	0.803682	0.575842
BKG_Return	0.803682	1.000000	0.600036
FTSE_Return	0.575842	0.600036	1.000000

Figure 12: Showing Pearson method of correlation for Barratt, Berkeley and FTSE Prices and Returns

According to Fernando (2021), a value of exactly 1.0 indicates that the two variables have a perfect positive relationship. There is a positive increase in the second variable for every positive increase in the first. A value of -1.0 indicates that the two variables have a perfect negative relationship. There is no linear relationship between two variables if their correlation is 0. Barratt and Berkeley Returns have the best relationship of 0.8, which is nearest to 1.0, according to the table above.

OLS Regression Results

Dep. Variable:	BDEV_Return	**R-squared:**	0.646
Model:	OLS	**Adj. R-squared:**	0.645
Method:	Least Squares	**F-statistic:**	882.9
Date:	Wed, 21 Apr 2021	**Prob (F-statistic):**	3.47e-111
Time:	23:21:26	**Log-Likelihood:**	1270.2
No. Observations:	486	**AIC:**	-2536.
Df Residuals:	484	**BIC:**	-2528.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0002	0.001	0.209	0.835	-0.001	0.002
BKG_Return	1.1059	0.037	29.713	0.000	1.033	1.179

Omnibus:	67.536	**Durbin-Watson:**	2.090
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	553.565
Skew:	-0.233	**Prob(JB):**	6.24e-121
Kurtosis:	8.208	**Cond. No.**	46.2

Figure 13: Barratt and Berkeley Price Regression summary table

OLS Regression Results

Dep. Variable:	BDEV_Return	**R-squared:**	0.646
Model:	OLS	**Adj. R-squared:**	0.645
Method:	Least Squares	**F-statistic:**	882.9
Date:	Wed, 21 Apr 2021	**Prob (F-statistic):**	3.47e-111
Time:	23:21:26	**Log-Likelihood:**	1270.2
No. Observations:	486	**AIC:**	-2536.
Df Residuals:	484	**BIC:**	-2528.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0002	0.001	0.209	0.835	-0.001	0.002
BKG_Return	1.1059	0.037	29.713	0.000	1.033	1.179

Omnibus:	67.536	**Durbin-Watson:**	2.090
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	553.565
Skew:	-0.233	**Prob(JB):**	6.24e-121
Kurtosis:	8.208	**Cond. No.**	46.2

Figure 14: Barratt and Berkeley Return Regression summary table

4.1 Regression table interpretation

Standard Error shows the accuracy of prediction for each variable. The lower the standard error, the better the estimate. Comparing the table 3 and 4 above we can imply that Barratt and Berkeley return shows a better standard error.

R-squared shows measures the goodness of fit of a regression model. The closer your R-squared value to 1, the better your regression model. According to the tables above, return R-square for return is better.

Prob F-statistics the regression model is examined using prob F-statistics. It evaluates the null hypothesis, which states that if the p-value is less than the significance mark, your sample results are sufficient to conclude that your regression model matches the data. (2021, Frost). The prob F-statistics of 3.47e-111 are shown in Table 4. This adds to the evidence that BDEV and BKG return have a significant relationship.

4.2 Regression Equation

$$Y = \beta_0 + \beta_1 x \tag{1}$$

Where Y is the expected value, β_0 is the regression constant, β_1 quantifies the effect of the X and Y variable, and X is the sample data.

Therefore, we have:

$$\begin{aligned}
\beta_0 &= 0.002, \\
\beta_1 &= 1.1059, \\
Y &= 0.002 + 1.1059x, \text{ or} \\
\text{BDEV Return} &= 0.002 + 1.1059 \times \text{BKG Return.}
\end{aligned}$$

Figure 15: Scatter plot showing Barratt and Berkeley Returns relationship with fitted line

5 Time Series Analysis

Time series is defined by Erica (2019) as data collected at various points in time. She further suggested that since data points in a time series are obtained at close intervals, there is the possibility of observational correlation. Several companies make use of judgmental forecasting techniques that rely on the knowledge of experienced employees and managers. Such a qualitative approach is common in the case where there is no historical data; for example, if one wants to forecast the sales of a new product. Other typical recent examples of situations where judgemental forecasting can also be crucial include Brexit (exit of the UK from the European Union), and the COVID-19 pandemic.As there is no precedent to these situations, it is almost impossible to accurately identify any historical factor that can be used to make reliable predictions of its impact on the UK or world economy.

5.1 Data Preparation

There are some steps before forecasting to simplify the problem and assist forecasters in identifying the patterns inherent in the time series.We describe here the adjustments that may be needed to a dataset before it is ready for application of forecasting models, after preliminary analysis.

5.2 Time series length

Consideration should be given to the length of the time series to be used for calculations of forecasts. Usually the entire available dataset is used, but sometimes changing conditions can produce radical changes in observed patterns. Sometimes forecasts cope well with such changes, but sometimes the methods do not cope well and it is better to truncate a dataset to more recent conditions.

5.3 Missing and erroneous data

Real-life data is liable to contain human errors, most of which cannot be known with certainty by the forecaster. However, some clear outliers may be considered with high probability to be erroneous. For example, a missing or extra numeral will produce a resulting number that is ten times smaller or larger than neighbouring entries in a time series, and could seriously disrupt a forecast. If the source of the data can be referred to, it might then be possible to correct the error: if not, an estimate should be made. If it is considered necessary to add an estimated value where missing or erroneous data is present, then this has to be carried out with due regard to the time series in question, and clear justification given. A local average value might be appropriate, but seasonality might also need to be considered.

```python
# Download BDEV.L Sports prices
BDEV = yf.download('BDEV.L', start='2019-01-02', end='2021-01-01')

# Define the training and test sets
# Download BDEV.L Sports prices
BDEV=yf.download('BDEV.L',start='2019-01-02',end='2021-01-01')

# Download BDEV.L Sports prices
BDEV=yf.download('BDEV.L',start='2019-01-02',end='2021-01-01')
```

Listing 1: Python Code for Data Analysis

5.4 Methods for Tuning Hyperparameters

In the realm of time series forecasting, selecting hyperparameters is a pivotal process. Research by Weerts et al. (2020) emphasizes the critical role of hyperparameter tuning in optimizing model performance. Hyperparameters, set before training begins, are crucial for determining the model's structural aspects like complexity and learning capabilities. These parameters, not derivable during training, require expertise for initial setting. This involves an experimental approach, starting with preset hyperparameters, training the model, evaluating specific metrics or results, and then interpreting these findings. Based on this interpretation, adjustments are made to the hyperparameters, iteratively refining the model to achieve satisfactory results. This method is complex and demands significant technical know-how.

The hyperparameter tuning process can be summarized in the following steps:

5.4.1 Selection of Cross-Validation Method

The choice of cross-validation method is crucial, as detailed in the referenced book. Cross-validation assists in hyperparameter optimization by offering reasonable metric estimations on the validation set. However, it typically overestimates these metrics, as it doesn't utilize the entire training dataset, leading to potential model bias.

5.4.2 Data Slitting for Training and Evaluation

The data split into training and validation sets must ensure no correlation to prevent data leakage and inflated performance metrics. In time series, the sequence in the training data must precede that in the validation set, known as rolling cross-validation, to maintain the integrity of prediction from past data.

5.4.3 Defining Hyperparameter Ranges

Determining potential hyperparameter values and conducting cross-validation with distinct training and validation sets help identify the most effective hyperparameter configuration.

5.5 Python-Based Tools for Hyperparameter Optimization

Hyperparameter tuning, an intricate yet essential part of model development, is often facilitated by automated algorithms. The following are key Python frameworks for this purpose:

5.5.1 Sklearn

Sklearn offers grid search and random search for hyperparameter tuning. Models compatible with Sklearn's estimator objects, including a variety of machine learning models such as linear, discriminant analysis, support vector machines, and ensemble models, can utilize these methods.

5.5.2 Keras Tuner and Bayesian Optimization

For deep learning models in Keras, Keras Tuner provides automated tuning with algorithms like random search, Bayesian optimization, and Hyperband. The BayesianOptimization package, utilizing Gaussian processes and built on scikit-learn, is another effective tool, especially for calculating validation set errors during cross-validation.

5.5.3 Practical Application in Hyperparameter Tuning

To demonstrate the effectiveness of the discussed methods, this report employs the 'BDEV.L' dataset, a manageable dataset size for practical application. The code implementation, as presented, is tested against this dataset to validate its efficiency in tuning hyperparameters for optimal forecasting performance.

5.5.4 Application in the Report

In the context of this report, the following hyperparameters were specifically tuned:

- **Time Series Decomposition:** Adjustment of parameters related to trend, seasonality, and residual components in the time series for optimal decomposition strategy.
- **Smoothing Levels and Trends:** Experimentation with various smoothing levels for both the level and trend components, impacting the model's ability to detect underlying data patterns.
- **Error Metrics Analysis:** Evaluation of models using MAE and MSE to understand the average magnitude of errors and their variance.

5.6 Moving Average

Moving Average is a technique used to smooth out short-term fluctuations and highlight longer-term trends in data. It's calculated by taking the average of any subset of numbers. n time series analysis, it helps in identifying trends and patterns by smoothing out noise from random short-term fluctuations. In Python, moving average can be implemented using the rolling() method from Pandas library, which provides rolling window calculations.

Figure 16: Weekly and Monthly Moving Average of Barratt Prices

Similar patterns can be seen in the weekly and monthly prices, as seen in the graph above. They're all ascending in the same direction. At the point where it starts ascending is the buying signal until it reaches its peak in February 2020. The selling signal begins when it begins to decline at March 2020, till it reaches its lowest point. This graph however, demonstrates Covid-19's negative effect on the real estate industry.

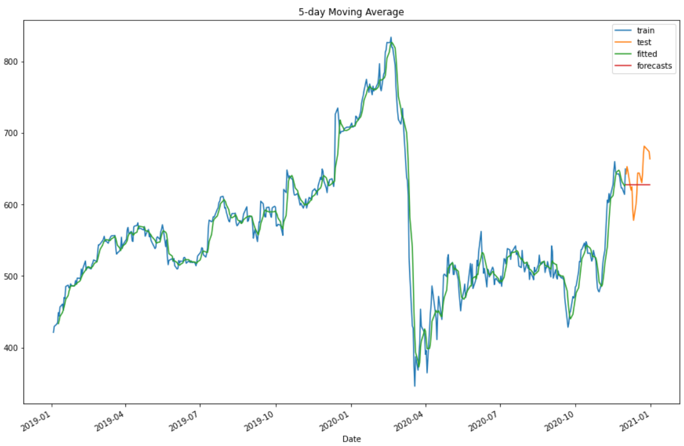

Figure 17: Barratt MA Price as compared with test price

MAE: 24.3739
MSE: 857.3291

The above figure shows the comparison of 20 forecasts by the weekly Moving Average (MA) for Barratt prices using the test set price. The forecast line, as seen, cuts through the test line. The test line shows to have moved below the forecast line and later ending above.

5.7 Single Exponential Smoothing(SES)

In this section, we introduce three exponential smoothing methods of forecasting that may be used to suit different conditions in a time series, i.e., depending on whether trend and seasonality are present or not. We start with the simplest form, single exponential smoothing. SES is a time series forecasting method for univariate data without trends or seasonality. It assigns exponentially decreasing weights as the observations get older. It's used when the data does not have trends or seasonal patterns and is useful for smoothing data and making short-term forecasts. The 'SimpleExpSmoothing' function from the statsmodels library can be used to implement SES in Python.

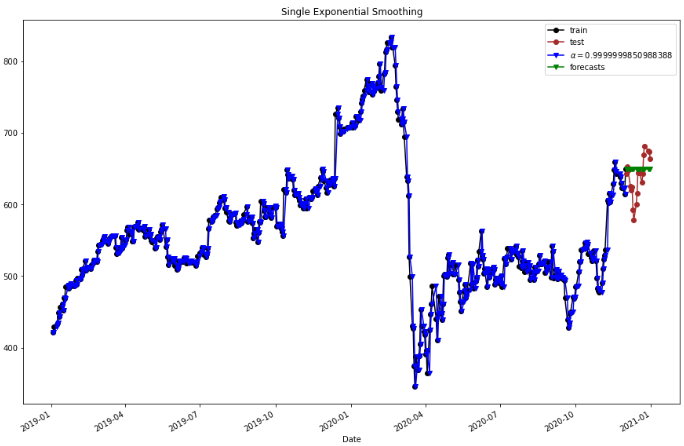

Figure 18: Single Exponential Smoothing of Barratt prices 20 forecast

MAE: 23.1260
MSE: 875.7185 The above figure shows the test prices also known as 'ground truth' moving downwards and upwards towards the forecast line. From close observations the forecast line looks like a straight line from December to January 2020. Four points of forecast ran through the test price line. This further shows that the performance is higher compared to the moving average.

5.8 Double Exponential Smoothing (DES)

Also known as Holt's Linear Trend Method, DES extends SES by adding support for trends in the univariate time series data. It was introduced by Charles Holt in 1957. DES is useful in forecasting data where there is a trend but no seasonality. It not only smoothens the level of the series but also the trend. Python's statsmodels library provides the Holt function to implement DES.

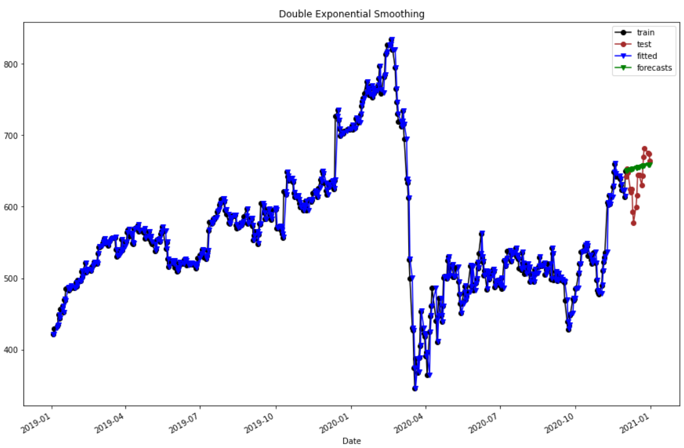

Figure 19: Double Exponential Smoothing method Barratt 20 forecast prices

MAE: 23.7379
MSE: 939.3185 The figure above shows the forecast meets the average level of test prices at the same point in January 2020. The test price in SES and DES has the same trend pattern. However, from the graph observation the forecast line only matched the test line at 3 forecast points. This is one point lesser than SES. The MAE and MSE is higher than the single exponential smoothing.

5.9 Holt's Winters Method (Additive)

This method is an extension of Holt's LES method for time series data that exhibits both trends and seasonality. The Holt's Winders additive assumes the seasonal variations are roughly constant through the series. The 'ExponentialSmoothing' function in the statsmodels library can be used for both additive and multiplicative Holt-Winters methods.

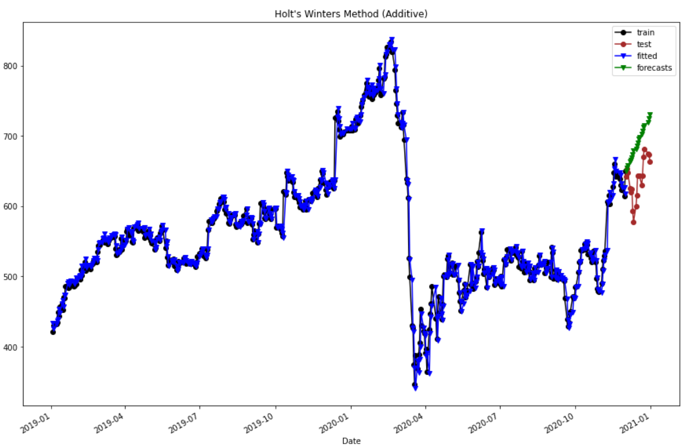

Figure 20: Holt's winters method (addictive) of Barratt 20 forecast prices

MAE: 50.3231
MSE: 3, 143.8824 The figure above shows the forecast to be above the average level of the test prices. There is no matching point between test and forecast lines. The forecast is observed to ascend upwards from December 2019. This is in major contrast as observed to the DES. This however means the forecasting performance of this method is poor.

5.10 Holt's Winters Method (Multiplicative)

This method is an extension of exponential smoothing for time series data that exhibits both trends and seasonality. This variation method assumes the seasonal variations are changing proportionally to the level of the series. These methods are used when the data exhibits both trends and seasonality. The choice between additive and multiplicative depends on the nature of the seasonal pattern. The 'ExponentialSmoothing' function in the statsmodels library can be used for both additive and multiplicative Holt-Winters methods.

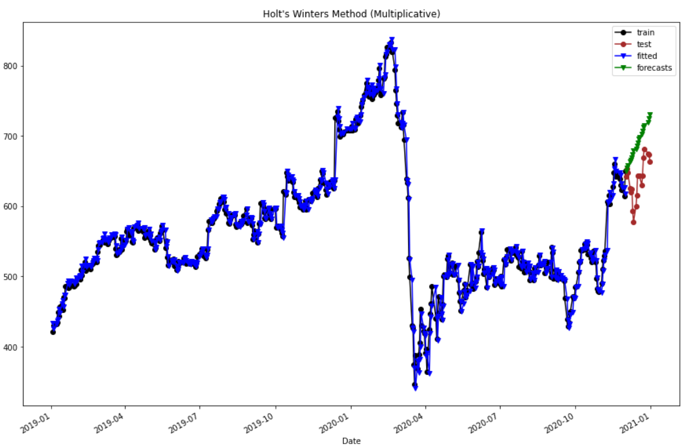

Figure 21: Holt's winters method (multiplicative) of Barratt 20 forecast prices

MAE: 50.3284
MSE: 3,144.4479 The figure above shows the forecast above the level of the average test prices with little difference compared to the Holt winter's additive method.

5.11 Summary

In summary, for a method/model to be regarded as good, the forecast line should match or be closely aligned on the same line with the test prices which is also known as the 'ground truth' or expectations. Erica (2021). The MAE and MSE methods should also have the smallest numbers. Single Exponential Smoothing is seen to have the better forecasting method for Barratt because it has the lowest MAE and MSE value. (MAE: 23.1260 MSE: 875.7185) Also, SES also has the highest number of forecast points (four) that matched with the test prices which however, is an indicator for higher performance. A structural break or non-linearity was observed from March 2020 because of covid-19 pandemic amongst all the forecast methods. The trend is observed to go up again from April 2020. Although Barratt has volatility, it is a good company to invest in because of its steady rise in stock prices.

6 Fundamental Analysis

The 2016 to 2020 annual report was extracted from Barratt Plc annual financial report website to visualize and observe changes in their comprehensive income and capital structure. The comprehensive income was analysed using Microsoft Excel software. Matplotlib with python was used to analyse and visualize the capital structure of Barratt Development Plc.

6.1 Financial Statements

6.1.1 Comprehensive Income

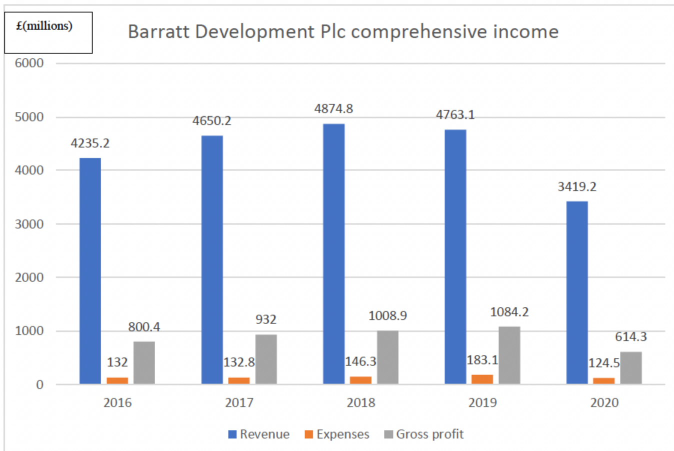

Figure 22: Barratt Comprehensive income (Data source: Barratt development report)

From the above chart, we can observe a progressive increase in revenue from 2016 that peaked in 2018, and a progressive gross profit that peaked in 2019. However, there is an observed decline in both revenue and gross profit in 2020. This can be attributed to the effect of the covid-19 pandemic that affected the business negatively.

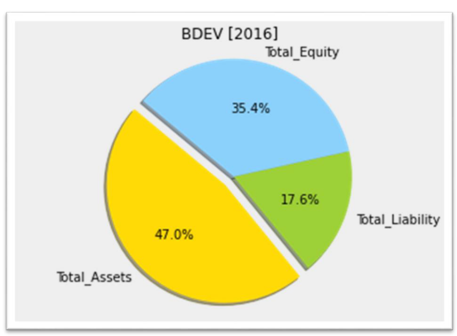

Figure 23: Pie chart showing Barratt 2016 capital structure

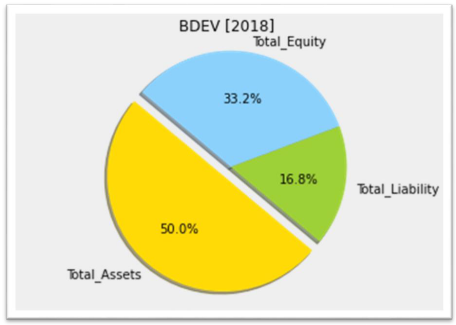

Figure 24: Pie chart showing Barratt 2017 capital structure

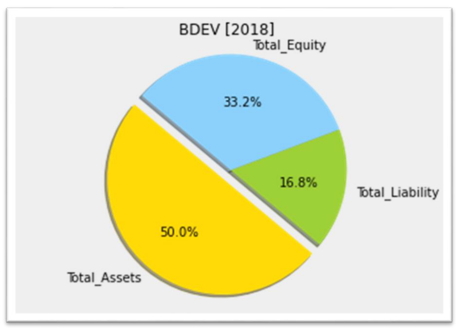

Figure 25: Pie chart showing Barratt 2018 capital structure

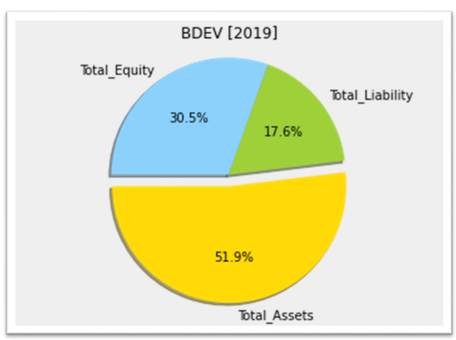

Figure 26: Pie chart showing Barratt 2019 capital structure

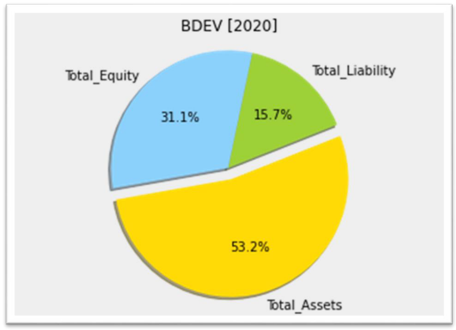

Figure 27: Pie chart showing Barratt 2020 capital structure

6.2 Strategic Analysis

A high Debt to Equity ratio is often correlated with high risk, according to Fernando (2021); it suggests that a business has been successful in funding its growth with debt. He went on to say that if the cost of debt financing outweighs the increased revenue produced, stock prices could fall. We can see from the above figures (Fig. 19-23) that assets accounted for the highest percentage in the five years. Despite the impact of the covid-19 pandemic, Barratt retained a higher ratio of Equity to Liability, i.e., the capital invested by investors is greater than the debt acquired by the business. This is another effort to establish Barratt as a secure investment. The company took a number of measures, including cancelling the interim dividend, which, although painful for shareholders in the short term, could ensure the company's long-term survival. (Barratt Plc, 2021).

7 Conclusion and Suggestions

In conclusion, while the box plot analyses showed that Barratt development has higher volatility/ risk than Berkeley and FTSE, they are a better investment in terms of higher return. The regression analyses fitted line had a better fit between Barratt and Berkeley returns because of the stronger linear relationship between the two variables. The time series analysis suggests Single Exponential Smoothing to be a better forecast model because it has the lowest Moving Average Error and the highest forecast line points matched with the test prices (ground truth or expectation). Barratt financial report shows a progressive increase in performance in terms of revenue, gross profit, total assets, equity and declined liability. All these are indicators of a healthy Return of Investment. Despite a significant drop in financial performance in March 2020 because of the Covid-19 pandemic, Barratt's pre-Covid-19 analyses results reveal a pattern among housebuilders: they're doing better than anticipated. When you add in management's conservative budget cuts, you get a strong cash position on top of brisk demand. Barratt has recovered well from the crisis and continues to be a market leader in the UK housing industry.

8 Appendix

8.1 Python code and Data Analysis

8.1.1 Data Processing

```python
# import required packages
import pandas as pd
import yfinance as yf
import numpy as np
import matplotlib as mp
import statsmodels as ss

# Download BDEV.L Sports prices
BDEV = yf.download('BDEV.L', start='2019-01-02', end='2021-01-01')

# select a column
BDEV['Adj Close']

# select a row
BDEV.loc['2019-01-02']

# select a cell
BDEV['Adj Close']['2019-01-02']

# select a column
BDEV.iloc[:,4]

# select a row
BDEV.iloc[0]

# select a cell
BDEV.iloc[0,4]

# Training set
BDEVtraining = BDEV.iloc[0:487]

# Drop the columns we don't need
BDEVtraining = BDEVtraining.drop
columns=['Open','High','Low','Close','Volume'])
print(BDEVtraining)

# Test set
BDEVtest = BDEV.iloc[487:507]

# or BDEVtest = BDEV.iloc[-20:-1].append(BDEV.iloc[-1])

# Drop the columns we don't need
BDEVtest = BDEVtest.drop(columns=['Open','High','Low','Close','
    Volume'])
print(BDEVtest)
```

Listing 2: Python Code for Data Analysis

8.1.2 Descriptive Analytics

```
returns = (BDEVtraining['Adj Close'] -
BDEVtraining.shift(1)['Adj Close']) /
BDEVtraining.shift(1)['Adj Close']

# BDEVtraining.shift(1) shifts all records down by one row.
BDEVtraining['Return'] = returns
BDEVtraining

# calculate daily returns of test set
BDEVtest['Return'] = (BDEVtest['Adj Close'] -
BDEVtest.shift(1)['Adj Close']) / BDEVtest.shift(1)['Adj Close']
BDEVtest

BDEVtest.iloc[0,1] = (BDEVtest.iloc[0,0] -
BDEVtraining.iloc[-1,0]) / BDEVtraining.iloc[-1,0]
BDEVtest

BDEVtraining.mean()
BDEVtraining.median()
BDEVtraining.var()

# Population variance, not required for your assignment
BDEVtraining.var(ddof=0)

# Sample standard deviation
BDEVtraining.std()

# Population standard deviation, not required
BDEVtraining.std(ddof=0)

# Return to risk = mean of returns / std of returns
BDEVtraining['Return'].mean()
/ BDEVtraining['Return'].std()
BDEVtraining.describe()

BDEVtraining.hist()

BDEVtraining.hist(column='Return', bins=20)

mp.pyplot.hist(BDEVtraining['Return'], bins=20)

htg = mp.pyplot.hist(BDEVtraining['Return'],
bins=20, density=True, cumulative=False, histtype='bar', color='c')
mp.pyplot.axvline(BDEVtraining['Return'].mean())
mp.pyplot.text(BDEVtraining['Return'].mean() *
1.1, 18, 'Mean: {:,.4f}'.format(BDEVtraining['Return'].mean()))
mp.pyplot.axvline(BDEVtraining['Return'].median(), color='r')
mp.pyplot.text(BDEVtraining['Return'].median() * 1.2, 14,
'Median: {:,.4f}'.format(BDEVtraining['Return'].median()))

htg = mp.pyplot.hist(BDEVtraining['Adj Close'],
bins=30, density=False, cumulative=False, histtype='bar', color='c'
    )
mp.pyplot.axvline(BDEVtraining['Adj Close'].mean())
mp.pyplot.text(BDEVtraining['Adj Close'].mean(), 55,
'Mean: {:,.4f}'.format(BDEVtraining['Adj Close'].mean()))
mp.pyplot.axvline(BDEVtraining['Adj Close'].median(), color='r')
```

```
mp.pyplot.text(BDEVtraining['Adj Close'].median(), 40,
'Median: {:,.4f}'.format(BDEVtraining['Adj Close'].median()))

BDEVtraining.boxplot(column='Adj Close')

priceline = BDEVtraining.plot.line(y='Adj Close', legend=False)
priceline.set_xlabel("Date")
priceline.set_ylabel("BDEV price")
```

Listing 3: Python Code for Calculating Returns and Statistics

8.1.3 Time Series

```
ma5 = BDEVtraining.rolling(5).mean()
ma20 = BDEVtraining.rolling(20).mean()
priceMAdata = pd.DataFrame({'Price': BDEVtraining['Adj Close'], '
    MA5': ma5['Adj Close'], 'MA20': ma20['Adj Close']})
priceMAdata.plot.line()

BDEVtraining['Adj Close'].plot(figsize=(15,8), title='BDEV price',
    fontsize=14)
BDEVtest['Adj Close'].plot(figsize=(15,8), title='BDEV price',
    fontsize=14)
priceMAdata['MA5'].plot(figsize=(15,8), title='BDEV price',
    fontsize=14)
mp.pyplot.show()

BDEVte.columns = ['BDEV_Price', 'BDEV_Return']
BKGte.columns = ['BKG_Price', 'BKG_Return']
FTSEte.columns = ['FTSE_Price', 'FTSE_Return']

### Moving Average
ma5 = BDEVtr.rolling(5).mean()
ma20 = BDEVtr.rolling(20).mean()

BDEVtr['BDEV_Return'].plot(figsize=(15,10), title='BDEV Return
    Moving average', label='Return')
ma5['BDEV_Return'].plot(label='MA5')
ma20['BDEV_Return'].plot(label='MA20')
plt.legend()
plt.ylabel('Return', fontsize=15)
plt.show()

BDEVma5fcsts = pd.Series([ma5['BDEV_Return'][-1]] * 20)
BDEVma5fcsts.index = BDEVte.index

# Including fitted lines
BDEVtr['BDEV_Return'].plot(figsize=(15,10), label='train', title='
    5-day Return Moving Average')
BDEVte['BDEV_Return'].plot(label='test')
ma5['BDEV_Return'].plot(label='fitted')
BDEVma5fcsts.plot(label='forecasts')
plt.legend()
plt.show()

```

```python
print('MAE: {:,.4f}'.format(MAE(BDEVte['BDEV_Return'], BDEVma5fcsts)), '\nMSE: {:,.4f}'.format(MSE(BDEVte['BDEV_Return'], BDEVma5fcsts)))

### Single Exponential Smoothing
BDEVses = SES(BDEVtr['BDEV_Return'], initialization_method="estimated").fit()
BDEVsesfcsts = BDEVses.forecast(20)
BDEVsesfcsts.index = BDEVte.index

BDEVtr['BDEV_Return'].plot(figsize=(15,10), marker='o', color='black', label='train', title='Single Exponential Smoothing')
BDEVte['BDEV_Return'].plot(marker='o', color='brown', label='test')
BDEVses.fittedvalues.plot(marker='v', color='blue', label=r'$\alpha=%s$' % BDEVses.model.params['smoothing_level'])
BDEVsesfcsts.plot(marker='v', color='green', label='forecasts')
plt.legend()
plt.show()

print('MAE: {:,.4f}'.format(MAE(BDEVte['BDEV_Return'], BDEVsesfcsts)), '\nMSE: {:,.4f}'.format(MSE(BDEVte['BDEV_Return'], BDEVsesfcsts)))

### Double Exponential Smoothing
BDEVdes = DES(BDEVtr['BDEV_Return'], initialization_method="estimated").fit()
BDEVdesfcsts = BDEVdes.forecast(20)
BDEVdesfcsts.index = BDEVte.index
BDEVtr['BDEV_Return'].plot(figsize=(15,10), marker='o', color='black', label='train', title='Double Exponential Smoothing')
BDEVte['BDEV_Return'].plot(marker='o', color='brown', label='test')
BDEVdes.fittedvalues.plot(marker='v', color='blue', label='fitted')
BDEVdesfcsts.plot(marker='v', color='green', label='forecasts')
plt.legend()
plt.show()

print('MAE: {:,.4f}'.format(MAE(BDEVte['BDEV_Return'], BDEVdesfcsts)), '\nMSE: {:,.4f}'.format(MSE(BDEVte['BDEV_Return'], BDEVdesfcsts)))

### Holt's Winters (Addictive)
BDEVtesa = TES(BDEVtr['BDEV_Return'], seasonal_periods=4, trend='add', seasonal='add', use_boxcox=True, initialization_method="estimated").fit()
BDEVtesfcstsa = BDEVtesa.forecast(20)
BDEVtesfcstsa.index = BDEVte.index

BDEVtr['BDEV_Return'].plot(figsize=(15,10), marker='o', color='black', label='train', title="Holt's Winters Method (Additive)")
BDEVte['BDEV_Return'].plot(marker='o', color='brown', label='test')
BDEVtesa.fittedvalues.plot(marker='v', color='blue', label='fitted')
BDEVtesfcstsa.plot(marker='v', color='green', label='forecasts')
plt.legend()
plt.show()

```

```python
print('MAE: {:,.4f}'.format(MAE(BDEVte['BDEV_Return'], BDEVtesfcstsa)), '\nMSE: {:,.4f}'.format(MSE(BDEVte['BDEV_Return'], BDEVtesfcstsa)))

### Holt's Winters (Multiplicative)
BDEVtesm = TES(BDEVtr['BDEV_Return'], seasonal_periods=4, trend='add', seasonal='mul', use_boxcox=True, initialization_method="estimated").fit()
BDEVtesfcstsm = BDEVtesm.forecast(20)
BDEVtesfcstsm.index = BDEVte.index
BDEVtr['BDEV_Return'].plot(figsize=(15,10), marker='o', color='black', label='train', title="Holt's Winters Method (Multiplicative)")
BDEVte['BDEV_Return'].plot(marker='o', color='brown', label='test')
BDEVtesm.fittedvalues.plot(marker='v', color='blue', label='fitted')
BDEVtesfcstsm.plot(marker='v', color='green', label='forecasts')
plt.legend()
plt.show()

print('MAE: {:,.4f}'.format(MAE(BDEVte['BDEV_Return'], BDEVtesfcstsm)), '\nMSE: {:,.4f}'.format(MSE(BDEVte['BDEV_Return'], BDEVtesfcstsm)))
```

Listing 4: Python Code for Various Time Series Methods

References

[1] Avdalović, S. M., & Milenković, I. (2017). *Impact of Company Performances on Stock Price: An Empirical Analysis on Select Companies in Serbia*. Economics of Agriculture, UDC: 347.471:336.761.5. Available at: https://core.ac.uk/download/pdf/208511661.pdf

[2] Barratt Development Plc. (2021). *About Us*. Available at: https://www.barrattdevelopments.co.uk/about-us

[3] Barratt Development Plc. (2021). *Annual Reports and Account*. Available at: https://www.barrattdevelopments.co.uk/investors/results-reports-and-presentations/rp-2016

[4] Berkeley Group. (2021). *About Us*. Berkeley Group. Available at: https://www.berkeleygroup.co.uk/about-us

[5] Erica, Aptech. (2021, March 8). *Introduction to the Fundamentals of Time Series Data and Analysis*. Available at: https://www.aptech.com/blog/introduction-to-the-fundamentals-of-time-series-data-and-analysis/

[6] Fernando, J. (2021). *Correlation and Coefficient*. Investopedia. Available at: https://www.investopedia.com/terms/v/volatility.asp

[7] Fernando, J. (2021). *Debt-To-Equity Ratio (D/E)*. Investopedia. Available at: https://www.investopedia.com/terms/r/return.asp

[8] Frost, J. (2021). *How to Interpret the F-test of Overall Significance in Regression Analysis*. Statistics By Jim. Available at: https://www.investopedia.com/terms/r/return.asp

[9] Hayes, A. (2020). *Correlation and Coefficient*. Investopedia. Available at: https://www.investopedia.com/terms/c/correlation.asp

[10] Hunger, J.D., Wheelen, T.L. (2011). *Essentials of Strategic Management*. Saint Petersburg, Florida: Prentice Hall.

[11] Hyndman, R., Koehler, A. B., Ord, J. K. & Snyder, R. D. (2008), *Forecasting with exponential smoothing: the state space approach*. Springer Science & Business Media.

[12] Kuepper, J. (2021). *Volatility*. Investopedia. Available at: https://www.investopedia.com/terms/v/volatility.asp

[13] Kuttner, K. N. (2013). *"Low interest rates and housing bubbles: still no smoking gun"*. In D. Evanoff (Ed.), *The Role of Central Banks in Financial Stability: How Has It Changed?*: World Scientific (pp. 159-185).

[14] Makridakis, S., Spiliotis, E. & Assimakopoulos, V. (2020), *'The M4 competition: 100,000 time series and 61 forecasting methods'*. International Journal of Forecasting, 36(1), 54–74.

[15] Stephens, M. (2005). *A Critical Analysis of Housing Finance Reform in a 'Super' Home-ownership State: The Case of Armenia*. Urban Studies, 42(10), 1795-1815. Retrieved April 24, 2021, from http://www.jstor.org/stable/43197200

[16] Yang, L. & Shami, A. (2020), *'On hyperparameter optimization of machine learning algorithms: Theory and practice'*. Neurocomputing, 415, 295–316.

[17] YCHARTS. (2021). *Barratt Development Plc a Debt to Equity Ratio*. YCHARTS.

[18] Zemkoho, A. (2022), *A basic time series forecasting course with python*. In 'Operations Research Forum', Vol. 4, Springer, p. 2.