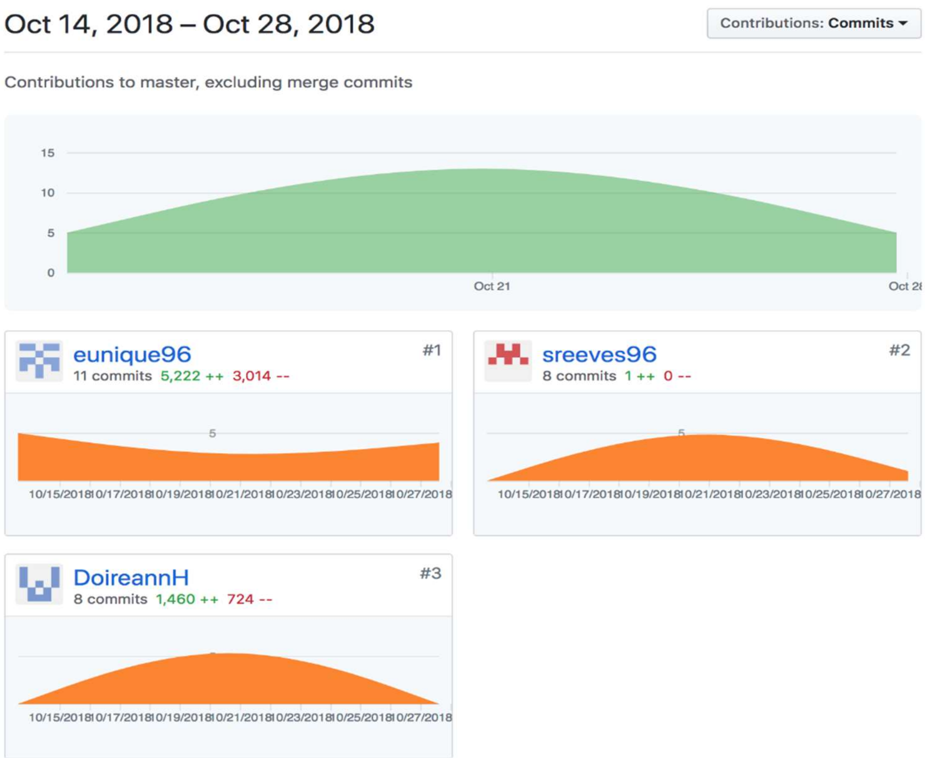
<u>ML1819 Research Assignment 1</u>

Team 26, Task 103 - Does ML algorithm performance change over time?

Eunice OreOluwa Fasan – 14311471 Methodology, Limitations and Outlook, Results and Discussion
Doireann Hanley – 15317527 Introduction, Limitations and Outlook
Sloane Reeves – 15320345 – Related Works, Limitations and Outlook

Word Count - 1053

Source Code:
https://github.com/eunique96/ML1819--task-103--team-26/blob/master/MMP_Forecast_Final.ipynb
https://github.com/eunique96/ML1819--task-103--team-26/blob/master/ARIMA.ipynb
Activity:
https://github.com/eunique96/ML1819--task-103--team-26/graphs/contributors



Single and ready to mingle?

Doireann Hanley
Trinity College Dublin
Computer Science and French
dhanley@tcd.ie

Eunice OreOluwa Fasan
Trinity College Dublin
Computer Science and French
fasane@tcd.ie

Sloane Reeves
Trinity College Dublin
Computer Science and German
sreeves@tcd.ie

1 INTRODUCTION

Machine Learning models are typically designed to predict future data. But often, over time, a model's predictive performance decreases when it is tested on a new, rapidly evolving dataset. The issue of 'model degradation' is ever present in the area of information security today and is essential in understanding a model's predictive performance.
"Are single-metric results a true reflection of an algorithm's performance as it pertains to time?", this question posed a problem to our research, single metric values do not provide a thorough insight into the performance of a model, but merely give us an idea of a model's accuracy. Our goal in this research assignment, is to determine the most accurate model out of the nine we chose, and implement them on a suitable dataset.

2 RELATED WORKS

While we aim to see how Machine Learning algorithms are affected over time, traditionally time has been only considered in seeking to predict the preference a user would have to an item, i.e. recommender-system research. Time is considered important in this case in marking the changes in user preference over time. Lee et al (2008) for example use a time-based approach to effective recommender systems using implicit feedback [4]. Their focus lies on adjusting algorithms to make them consider time and improve the effectiveness of the algorithms in accordance with the users wishes [3]. An example of a temporal feature as such is seasonality. Seasonality is something that is often considered as an important contextual feature, i.e it is something to be considered that one should use a different algorithm in winter than in summer. However, it seems from our study that all of the research work in concern to time is trying to use time to improve machine learning algorithms as opposed to a comparison study for machine learning models for regression, which we aimed to conduct. To the best of our knowledge, most of the works that relate to our studies is focused on the effectiveness of particular a ML algorithm over a predictive time-series. Kyoung-jae Kim et al (2002) tries to evaluate the feasibility of using Support Vector Machines in order to predict a financial time series [2]. This study examines the feasibility of applying SVM in financial forecasting by comparing it with back-propagation neural networks and case-based reasoning. Another study conducts time-series forecasting focusing on using SES, one of the models we observe in our study [5]. The work we found most closely related to ours, Atiya and Ahmed et al(2010), utilises a similar method to ours aiming to answer a similar question, as they present a large scale comparison study for the major machine learning models for time series forecasting. The models that they use differ from ours using 'multilayer perceptron, Bayesian neural networks, radial basis functions, generalized regression neural networks, K-nearest neighbor regression, CART regression trees, support vector regression, and Gaussian processes'. They also apply the models to monthly time series data however on a larger scale, with 'around a thousand time series'[1]'.

3 METHODOLOGY

The dataset we used, titled 'Monthly milk production: pounds per cow. Jan 62-Dec 75' was sourced from the Time Series Data Library (TDSL). The dataset consists of 168 fact values in 1 time series. The time granularity is Month and the units used are pound per cow. The time range is Jan 1962 – Dec 1975.
Our data frame was created by reading the data into a .csv file. We parsed and formatted the dates accordingly and set the attribute 'Month' to be our index. We then simply plotted the data to look for potential patterns, evidence of trend and seasonality. We defined our problem and based on our time series data, proceeded to implement a range of quantitative forecasting methods. These included Moving Average, Exponential Smoothing, Double Exponential Smoothing, Holt-Winter's model, ARIMA, Simple Lags/Linear regression, Ridge regression, Lasso regression and XGBoost.
We chose to measure the quality of our forecasts using Mean Absolute Percentage Error (MAPE) as we consider it an

interpretable metric that is particularly interesting because of its robustness to outliers.

We applied our baseline model, Moving Average, to smooth the original time series to identify trends.

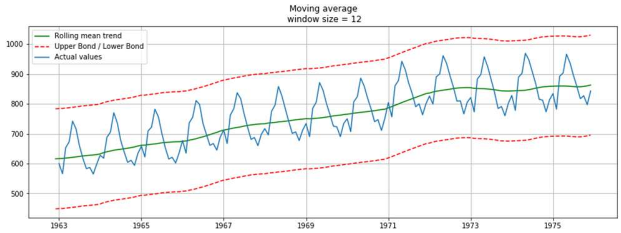


Figure 1: Results from smoothing original time series with the previous 12 months and plotting the confidence intervals

We then applied exponential smoothing and eventually extended this model to produce a double exponential smoothing whereby we tuned two parameters responsible for the series smoothing around itself, as well as the smoothing of the trend itself.

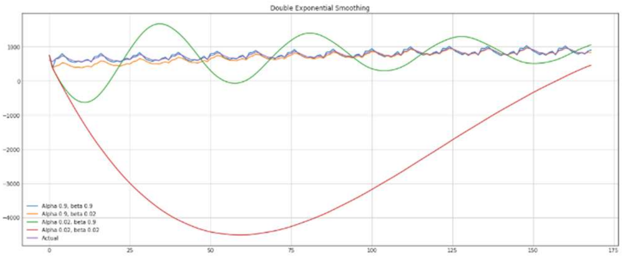


Figure 2: Double exponential smoothing of data with alpha parameter values of 0.9 and 0.02 and beta values of 0.9 and 0.02

The data was then split into training and test sets in the 80:20 ratio and thereafter we added a third component, seasonality, to our smoothing, effectively giving us a Holt-Winters model.

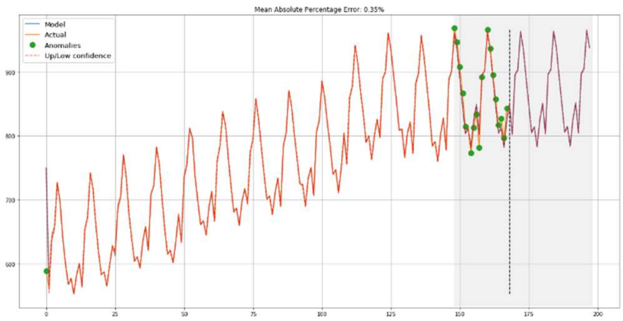


Figure 3: Holt-Winter model approximating initial time series, capturing annual seasonality, overall upwards trend and even some anomalies with MAPE of 0.35%

We then focused on achieving stationarity to increase our chances of predictions with minimal error. We used the Dickey-Fuller test, first order difference, ACF and PCF to reach stationarity.

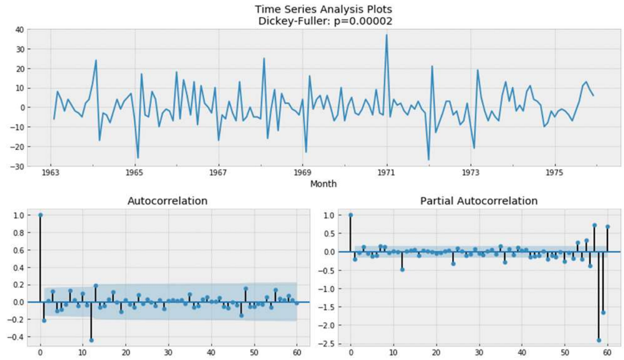


Figure 4: Stationary time series oscillating around zero. Dickey-Fuller test indicates stationarity as p< 0.05 i.e. p=0.00002 rejects null hypothesis that states the time series is non-stationary.

We then trained a series of models and determined their MAPE error.

4 RESULTS AND DISCUSSION

Holt-Winters model returned the least amount of error with a MAPE of 0.35% as opposed to Linear regression with simple lags which returned a MAPE of 2.43% and other algorithms such as Lasso and Ridge regression as well as XGBoost which returned errors of 2.66%, 2.45% and 3.45% respectively.

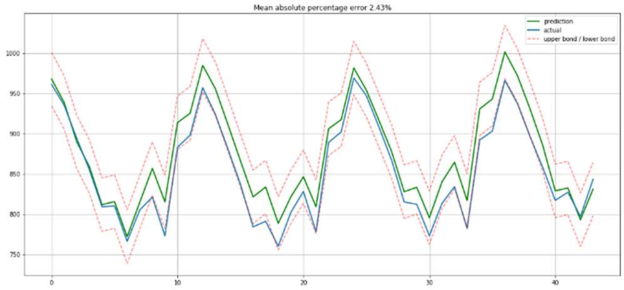


Figure 5: Simple lags and linear regression plot giving approximated predictions with an error of 2.43%

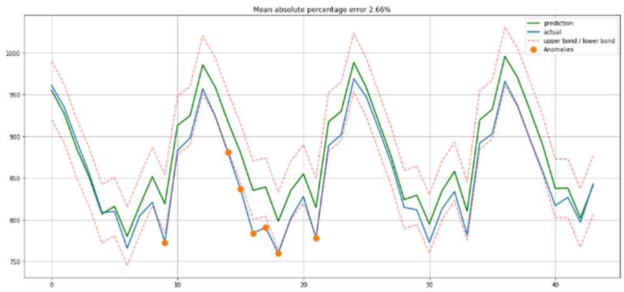


Figure 6: Lasso regression prediction with MAPE of 2.66%

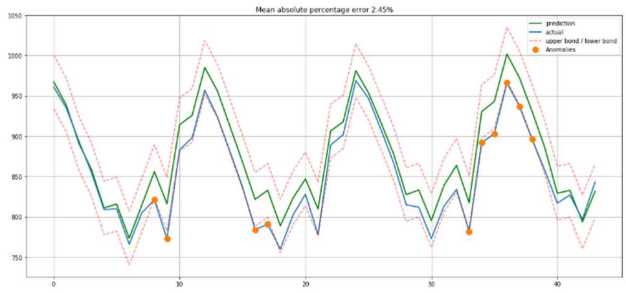


Figure 7: Ridge regression prediction with MAPE of 2.45%

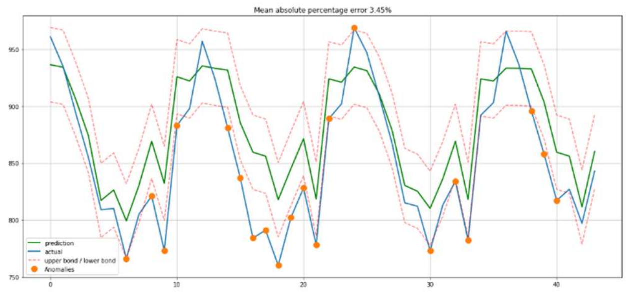


Figure 8: XGBoost model prediction with MAPE of 3.45%

If we were to rate these models based solely on their individual MAPE scores, Holt-Winters would be a clear winner. We must however take into account that these error values are only an average of the cumulative monthly errors. If we were to base our overall winner on plotted predicted values, expected values and most importantly error per time interval i.e. each month of the year, it's possible that the Holt-Winters model could be dethroned as there could be a decreasing trend with its predictions and an increasing trend with another model's predictions.

5 LIMITATIONS AND OUTLOOK

One crucial step to help us better evaluate our chosen models and their performance is to plot the error per time interval i.e. each month, as opposed to an overall single number metric telling only of the average error over the previous 12 months. Errors in our code denied us this insight therefore we will aim correct our code and produce the desired result as part of our second submission. We also had issues with identifying the correct parameters for the ARIMA model, we were not fully proficient in choosing the parameters from reading the displays of the partial autocorrelation function and the autocorrelation function. Hence, the ARIMA model has been left out of the first submission, but we will improve on this for our second submission.

6 REFERENCES

[1] Ahmed, N., Atiya, A., Gayar, N. and El-Shishiny, H. (2010). An Empirical Comparison of Machine Learning Models for Time Series Forecasting. Econometric Reviews, 29(5-6), pp.594-621.

[2] Kim, K. (2003). Financial time series forecasting using support vector machines. Neurocomputing, 55(1-2), pp.307-319.

[3] Koren, Y. (2010). Collaborative filtering with temporal dynamics. Communications of the ACM, 53(4), p.89.

[4] Lee, T., Park, Y. and Park, Y. (2008). A time-based approach to effective recommender systems using implicit feedback. Expert Systems with Applications, 34(4), pp.3055-3062.

[5] Ostertagová, E. and Ostertag, O. (2012). Forecasting using simple exponential smoothing method. Acta Electrotechnica et Informatica, 12(3).