**Data Science Test Report**

**Introduction**

This short report presents the analytics result of a time series dataset, which features on the total number of records and total submitted file sizes in GB (gigabytes) aggregated on a monthly basis for the past 10 years, from 2012-03 to 2022-12. The goal of the analysis is to build a non-neural network model that can accurately forecast the file size for the unrecorded next month, i.e., 2023-01, with quantitative error metrics to evaluate the model forecasting performance, whilst providing any insights and trends that can be gleaned from the data.

The most popular benchmark statistical model is the Autoregressive Integrated Moving Average (ARIMA), which takes a moving-average approach to estimate the upcoming series data, is not quite computational efficient and performs poorly in long range forecasting particularly for nonstationary data. In order to achieve high accuracy prediction for both point-based and vector-based prediction tasks, a more potent statistical model vector autoregression (VAR) and two non-neural network machine learning models, support vector regression (SVR) and a gradient boosted regression tree model (Xgbregressor) are tested.

The report results will be presented the analytics result in the following structure: first we will look at the dataset attributes and perform some explorations. Then we will introduce the statistical and machine learning models for the forecasting task. Lastly, the model performance comparison and analytics results will be given, followed by a report summary.

**The Dataset**

The dataset can be read into a Pandas DataFrame with three attributes *'submitted\_date'* which entails the month when the date was recorded, *'records'* which tells the total number of records received in that month, *'file\_size\_gb'* which gives the total submitted file sizes in GB. A direct visualization of how the numbers of records and submitted file sizes are shown in Fig. 1. In the plot, we simply perform an inspection on the data and are able to identify a few possible anomalies, which are likely to be a faulty record or incident trigger. Based on the plot of *'records'*, we see the record value at 2021-04 is abnormally high comparing the its adjacent values and any other in the series, hence it is likely to be a mistake. Similarly, the record values at 2020-07 and 2022-06 are also relatively high but not significantly so. Their corresponding data of the same timestamps in the *'file\_size\_gb'* plot also seem to peak, which indicates the anomalies might also be triggered by incident interventions. These can be further justified by a boxplot outlier detection method.

A boxplot is used to identify outliers by displaying data distribution by defining five quantities of the data, i.e., the median, the lower quantile (Q1), the upper quantile (Q3), minimum and maximum. InterQuartile Range (IQR) is defined as the range between the lower quantile and the upper quantile. The full range is further defined as minimum-maximum based on the IQR. An observation is considered a suspected outlier or potential outlier if out of IQR1.5 range ( ); whilst any observation is considered a very-likely outlier if out of IQR3 range ( ). The boxplot for the given dataset is illustrated in Fig. 2.1 and Fig. 2.2, where the IQR are set to 1.5 and 3, respectively. The suspect outliers are listed in Table 1.

From Fig. 2.1, Fig. 2.2 and Table 1, it can be observed that, with IQR1.5 criteria, two anomalies are identified as *‘Records’* values in 2021-04 and 2022-06. However with IQR3 criteria, only the 2021-04 anomaly is identified and indicates a strong outlying behavior comparing to the rest, despite there also seems to be an incident trigger. A separate DBACAN clustering detector has also been tested to only show that the 2021-04 *‘Records’* is the only anomaly. Nonetheless, we will only regard 2021-04 as an anomaly and amend its *‘Records’* using an interpolation method. In order to amend the anomaly, the corresponding value is first removed; then a simple interpolation method can be used to make a reasonable prediction. Here a polynomial interpolation with high order is used for its nonlinear characteristic to fill a missing value from its adjacent values. The amended data without anomaly is shown in Fig. 3. As can be seen the *‘Records’* data at 2021-04 has been adjusted to a reasonable level.

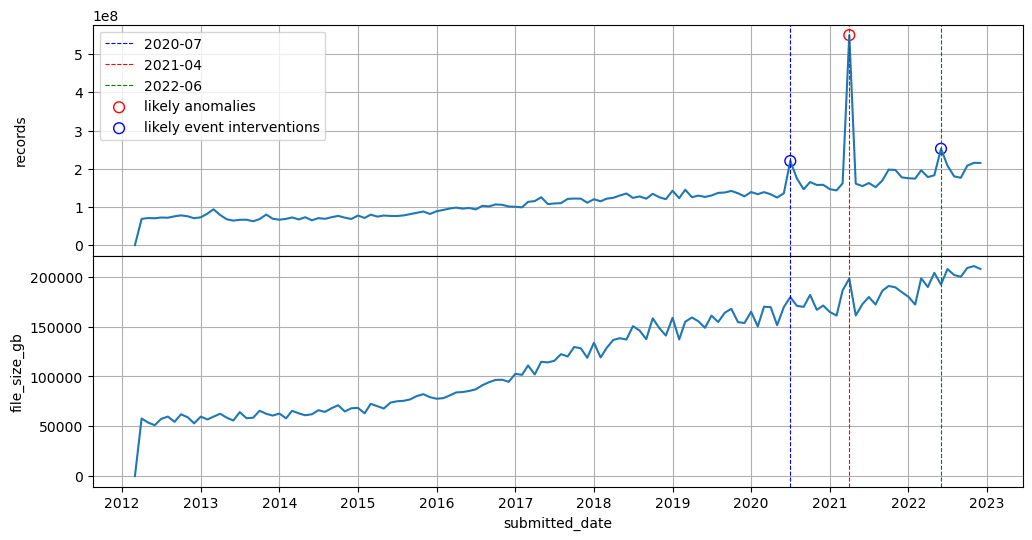


Fig. 1. Number of records and total file sizes by month from 2012-2022 with identified possible anomalies

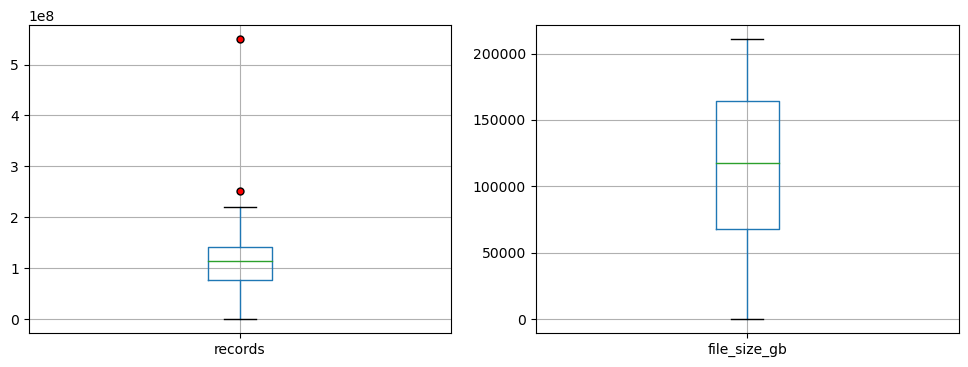


Fig. 2.1. Boxplot for outlier identification with IQR = 1.5

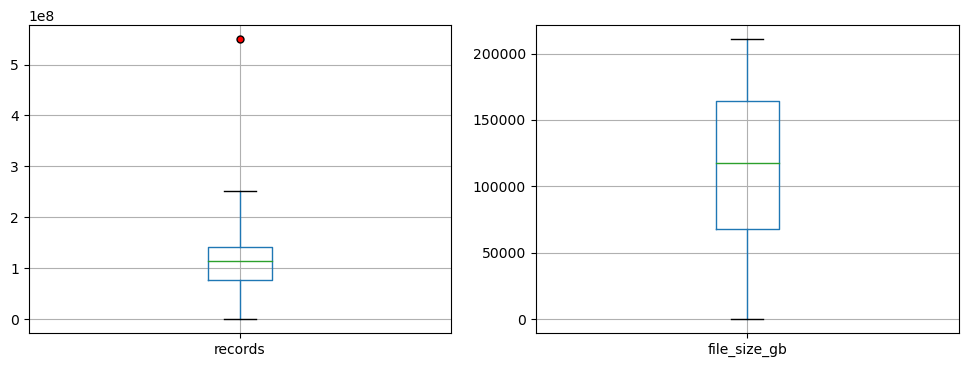


Fig. 2.2. Boxplot for outlier identification with IQR = 3

Table 1 Boxplot result parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *‘Records’* | | *'file\_size\_gb'* | |
|  | IQR1.5 | IQR3 | IQR1.5 | IQR3 |
| IQR | 64430061.25 | | 96646.38 | |
| Lower quantile | 76652594.0 | | 67904.11 | |
| Upper quantile | 141082655.25 | | 164550.49 | |
| Minimum range | 0 | 0 | 0 | 0 |
| Maximum range | 237727747.12 | 334372839.0 | 309520.06 | 454489.63 |
| Outliers | 2021-04  2022-06 | 2021-04 | nan | nan |

From Fig. 2.1, Fig. 2.2 and Table 1, it can be observed that, with IQR1.5 criteria, two anomalies are identified as *‘Records’* values in 2021-04 and 2022-06. However with IQR3 criteria, only the 2021-04 anomaly is identified and indicates a strong outlying behavior comparing to the rest, despite there also seems to be an incident trigger. A separate DBACAN clustering detector has also been tested to only show that the 2021-04 *‘Records’* is the only anomaly. Nonetheless, we will only regard 2021-04 as an anomaly and amend its *‘Records’* using an interpolation method. In order to amend the anomaly, the corresponding value is first removed; then a simple interpolation method can be used to make a reasonable prediction. Here a polynomial interpolation with high order is used for its nonlinear characteristic to fill a missing value from its adjacent values. The amended data without anomaly is shown in Fig. 3. As can be seen the *‘Records’* data at 2021-04 has been adjusted to a reasonable level.

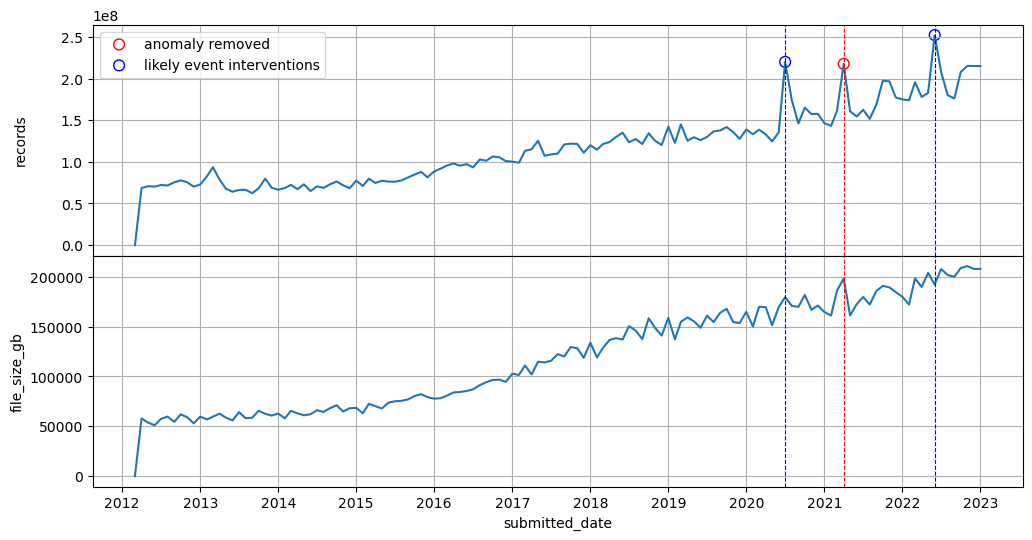


Fig. 3. Dataset without anomalies

Next, we conduct a statistical decomposition for the variables of the dataset separately, into three components: global trend, seasonality and residual which tells whatever remains if removing the trend and seasonality from the data. We use additive decomposition with a period of 3 months. The result is given in Fig. 4.

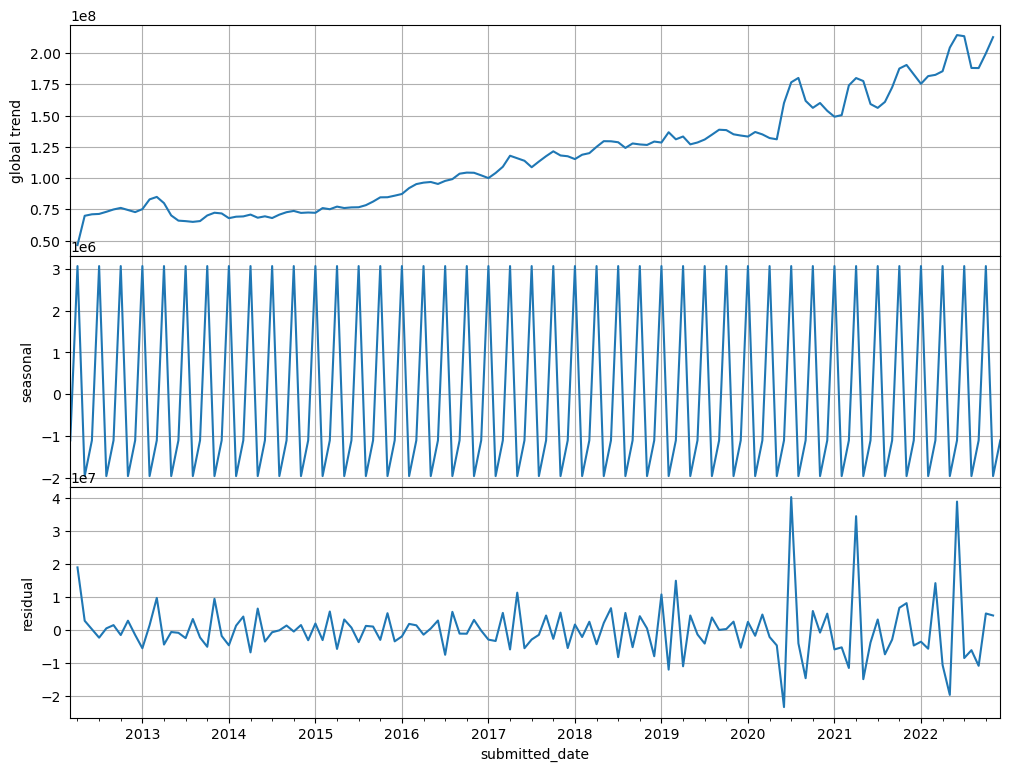


Fig.4. Trend, seasonality and residual decomposition of the variable *‘Records’*

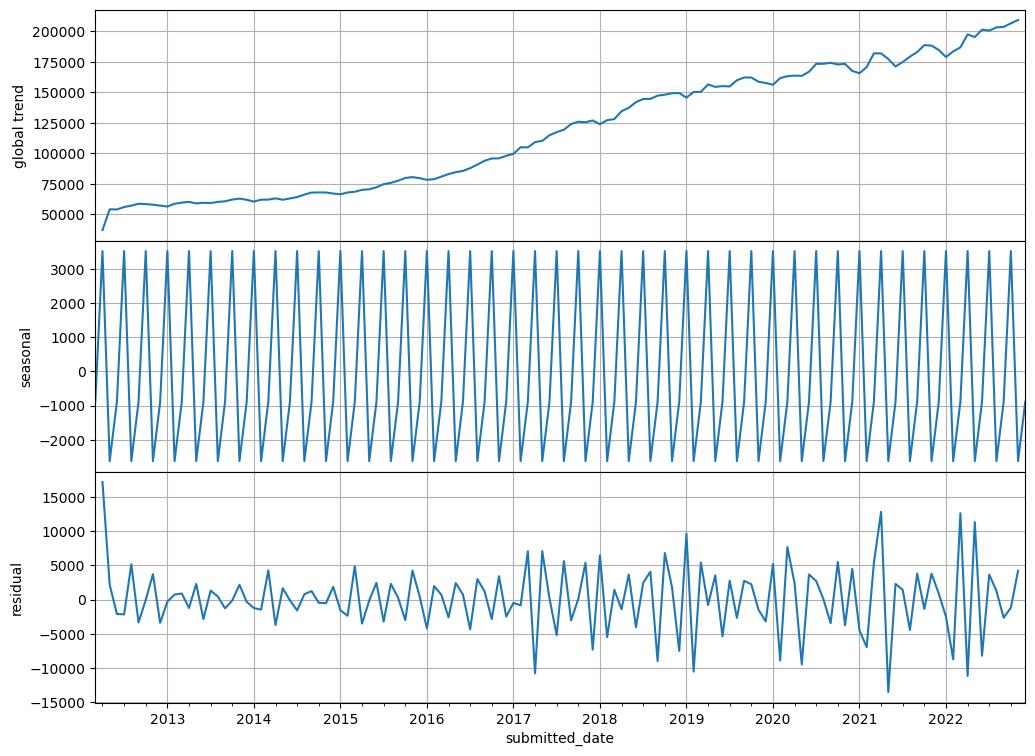


Fig. 4. Trend, seasonality and residual decomposition of the variable *'file\_size\_gb'*

It is a much more intuitive way to tell how the each of the variables varies at the microscope and repeats certain behaviors at a fixed frequency. From Fig.4. we can say that both variables have a steady increasing global trend but also cycles with small amplitudes locally. In the more recent years, their global trend also starts to have small seasonal behaviors due to the small fluctuations.

Next, we will investigate if there exists any collinearity between the two variables. This is important since if collinearity is observed, the same prediction model cannot be used to predict each of the variables independently. Fig. 5 shows the scatter plot of the two variables against each other representing the data points.

Both linear and nonlinear polynomial regressions are performed on the data. The data points distributed closely to the linear curve have a strong correlation between them, which in this case, are the data with smaller values in both *‘Records’ and 'file\_size\_gb'*. The data points distributed further out from the linear curve have a weak correlation between them, which in this case are data with the larger values in both variables, which correspond to the more recent years’ observations. AN insight from this is that during the more recent years, not only more file upload takes places but the file sizes are also more diverse.

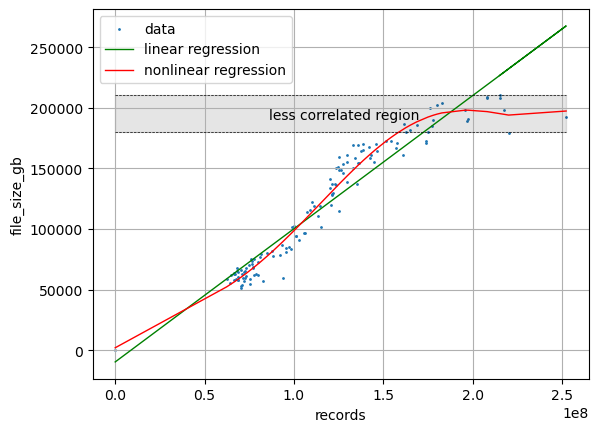


Fig. 5. Correlation analysis between ‘Records’ and 'file\_size\_gb'

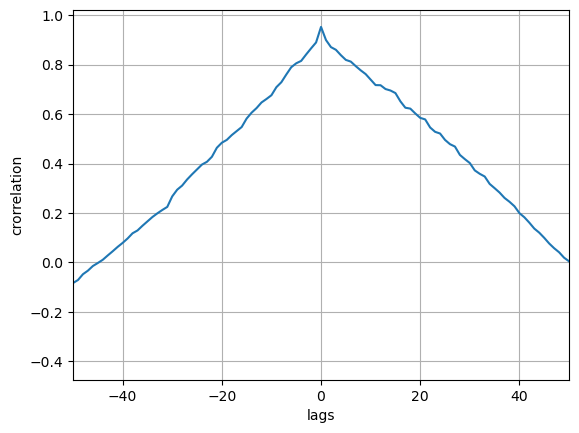


Fig. 6. Correlation analysis with time lags

The correlation with time lag effect is also analyzed and provided in Fig. 6, which plots the Pearson correlation of the variables with different lag values. It tells that the maximum correlation is achieved when there is no time lag (the case in Fig. 5). With time lags increases in either direction, the correlation decreases linearly.

Next, it can also be helpful if we know if the data is stationary or nonstationary. A stationary time series means stable statistical properties in the data whilst nonstationary means the series’ statistical properties in unstable and changes at different times. Augmented Dickey-Fuller (ADF) test is usually preferred for analyzing the stationarity of data by conducting a hypothesis test on if the time series is stationary or not. In ADF, the time series and its differenced versions with increasing order are analyzed (figure provided in Fig. 7.). If the p-value is obtained is greater than significance level of 0.05 and the ADF statistic is higher than any of the critical values. The null hypothesis suggests the time series is non-stationary. The original, first-order differencing and second-order differencing result are provided in Table 2. The result shows the only stationary series is the first-order differencing *‘Records’* series for it has a p-value less than 0.05 while the rest are nonstationary.

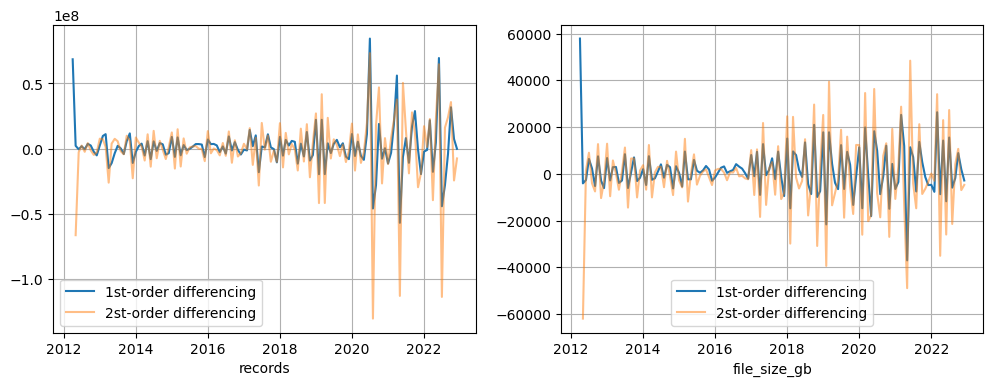


Fig. 7. First and second-order differencing for both variables.

Table 2: ADF stationarity test

|  |  |  |
| --- | --- | --- |
|  | *‘Records’* p-value | *'file\_size\_gb'* p-value |
| Original series | 0.999 | 0.994 |
| 1st-order differencing series | 0.006 | 0.139 |
| 2nd-order differencing series | 1.247 | 3.865 |

**The Models**

Now we will briefly describe the models used for conducting the forecasting tasks. A total of four models are tested, i.e. two statistical models including the benchmark model ARIMA, the Vector Autoregression (VAR) model; two non-neural network machine learning models, i.e., the Kaggle data science competition winner model XGBoost but in its regression version (XGBRegressor [[1]](#footnote-1)), Support Vector Regression (SVR[[2]](#footnote-2)).

Autoregressive Integrated Moving Average (ARIMA)

ARIMA makes use of lagged moving averages to smooth time series data is widely used to forecast future trend of time series data. The underpinning assumption of it is that the future resembles the past, which however is inaccurate since this does not account for data drift and change of trends and statistical properties, incident interventions and more. In ARIMA, parameters like time lags, differencing order and window size can be defined. It can be computational expensive and less effective in long range predictions.

Vector Autoregression (VAR)

VAR is yet another autoregressive model and similar to ARIMA, but it significantly differs from most autoregressive models by its capacity to predict all variables at once. It runs over a number of variable (endogenous variables in VAR’s term) simultaneously and make prediction based on their lag values. It also requires the data to stationary in order to guarantee best performance.

XGBRegressor

XGBRegressor is a part of the XGBoost model libraries but used for supervised regression purposes. The XGBoost is an open-source API where a large volume of gradient boosted tree models methods with flexible implementations, scalable processing framework and a large number of tunable hyperparameters such as depth of trees, learning rate etc. It is basically a machine learning model framework and usually much less computational expensive compared to neural networks. It uses a collection of decision trees (DT) to perform regression on different subset of the data for each tree, which can also be computationally parallelized. It has been applied and tested successfully for various real-world applications with satisfactory results and won the first place in Kaggle data competition not long ago since initial release.

Support Vector Regression (SVR)

SVR is a derivation of Support Vector Machines (SVM) with the same principle but for different machine learning tasks: whilst SVM is popular for classifications, SVM targets at performing regressions by finding the flattest hyperplane which also maximizes the number of data points on the hyperplane. When fitting the data, hyperparameters like its kernel, the epsilon value relating to the flatness of the hyperplane and regularization parameters require to be tuned. SVR can be computationally expensive when fitting a large dataset.

**Modelling results**

We will now divide the time series dataset into training and testing sets, train the above models with the training set and test on the test set, by evaluating and comparing the error metrics such as Mean Absolute Percentage Error (MAPE), Weighted Mean Absolute Percentage Error (WMAPW) and Root Mean Squared Percentage Error (RMSPE) for each of the models. Once the models are tested, they are used to make a final forecasting for the data at 2023-01 timestamp.

The dataset has a total of 130 data samples and is divided into 110 training and 20 testing samples. Note there is comparably more training to test data in this split is because the data are not shuffled for training ARIMA and VAR and we want to preserve the trend changes as much as possible into the later timestamps.

The data is fed directly to the ARIMA and VAR model. For ARIMA, the moving average window is set at 7, difference is set to 1 (based on previous stationarity analysis), autoregressive order is also set to 1. For the VAR model, the model order is manually optimized to 9. Modelling results are presented in Fig. 8 and Fig. 9

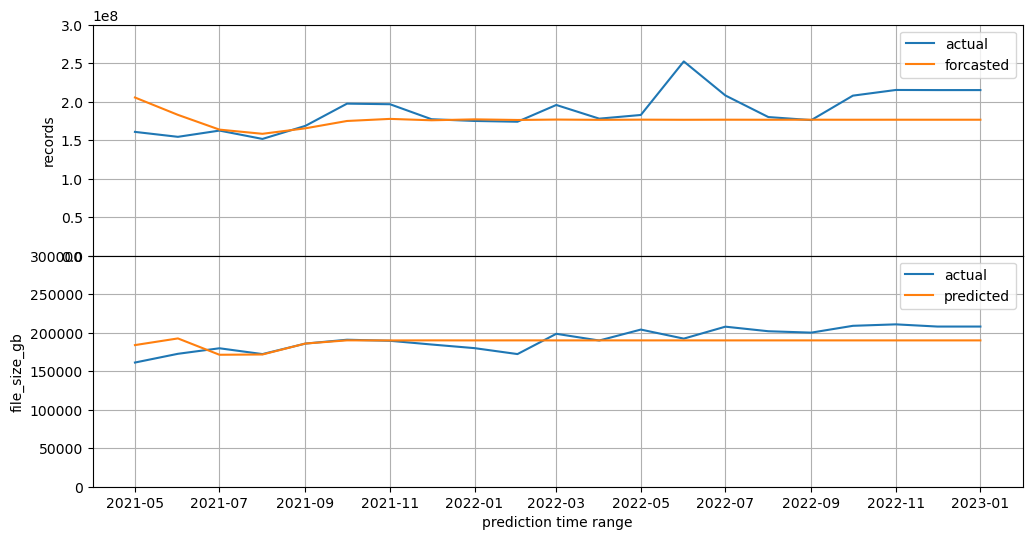


Fig. 8. ARIMA forecasting on the test data

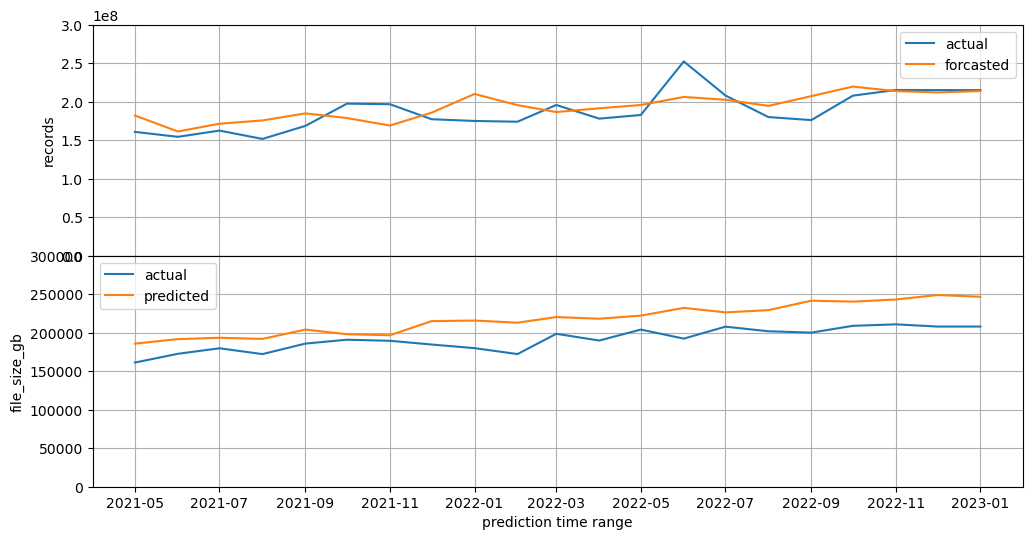


Fig. 9. VAR forecasting on the test data

For XGBRegressor and SVR machine learning models, the data requires to be preprocessed into proper shape to match their input-output requirements. To address this, the data is shifted a time step to the right each time to generate a new sample (see Fig. 10.). The window length is set to 10 so as to generate a total number of 120 new data sequences. Since both models are supervised learning and the task is to perform forecasting, each new sample is labeled with a target value that is just one time step further to the sequence. The completed new dataset is later randomly divided into 100 training and 20 testing samples. For model validation purpose, 20 out of the 100 training samples are used for cross validation. For hyperparameters of XGBRegressor, 100 decision trees with depth of 5 are used. For SVR, radial-basis function kernel is used with small epsilon-value equals 0.001 for precise optimization among normalized data points. Validation scores for both models are listed in Table 3. The process is done for each variable and so are for the training and prediction. The original data is also normalized and rescaled before training and after testing to ensure stability.

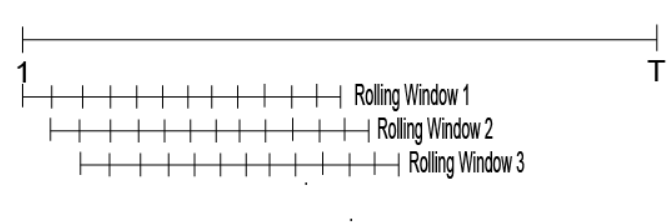


Fig. 10. Sampling on the original data using a shifting window method

Table 3: cross validation scores for XGBRegressor and SVR models

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | cv score 1 | cv score 2 | cv score 3 | cv score 4 | cv score 5 |
| XGBRegressor  *‘Records’* | 0.85859529 | 0.91775959 | 0.77649021 | 0.90625942 | 0.86414594 |
| XGBRegressor *'file\_size\_gb'* | 0.94404033 | 0.97571056 | 0.96179996 | 0.97125546 | 0.96525569 |
| SVR *‘Records’* | 0.92799065 | 0.88568179 | 0.74850591 | 0.84978276 | 0.91827566 |
| SVR *'file\_size\_gb'* | 0.97086331 | 0.96598467 | 0.96985196 | 0.97479508 | 0.97790266 |

It is noteworthy that the testing samples are randomly selected during the new data sample shuffling process. Hence, it is impossible to produce a prediction curve on the test data on continuous timestamps. However, it is possible to calculator the error metrics based on the random testing samples and perform a separate fitting performance test on the 20 most recent timestamps following after. In all plots, the data entry at 2023-01 has been padded with the same values as 2022-12 before any testing or forecasting. The fitting performance are presented in Fig. 11 and Fig. 12 respectively for XGBRegressor and SVR respectively.

Lastly, the error metrics of all four models, and the final forecasting for 2023-01 are shown in Table 4. As a result, XGBRegressor and SVR models achieve much better error metrics on the forecasting of *'file\_size\_gb'* compared to ARIMA and VAR, with MAPE and WAMPE errors below 5%. All four models do not seem to perform well on forecasting ‘Records’ with a average MAPE and MAPE around 10%.

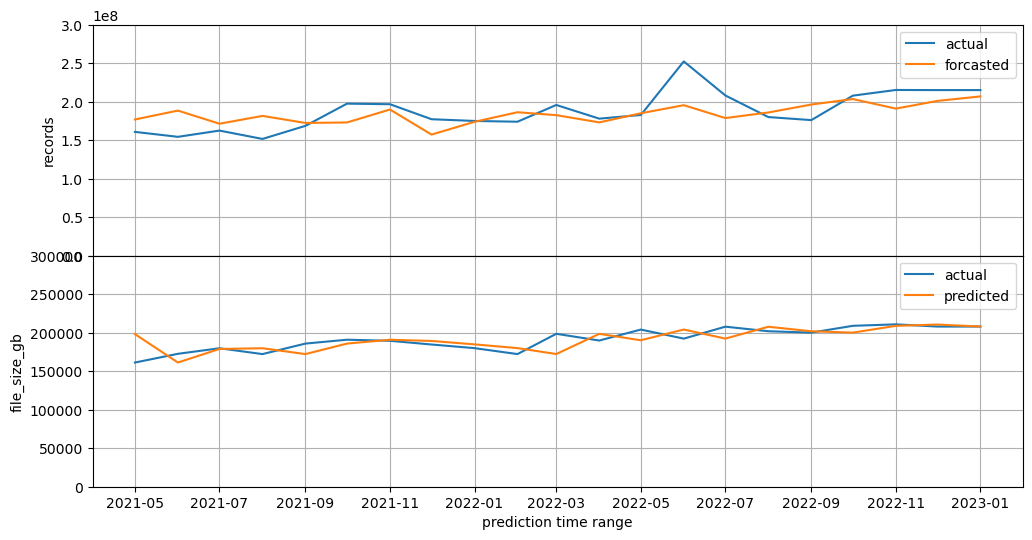


Fig. 11. XGBRegressor forecasting performance on test data

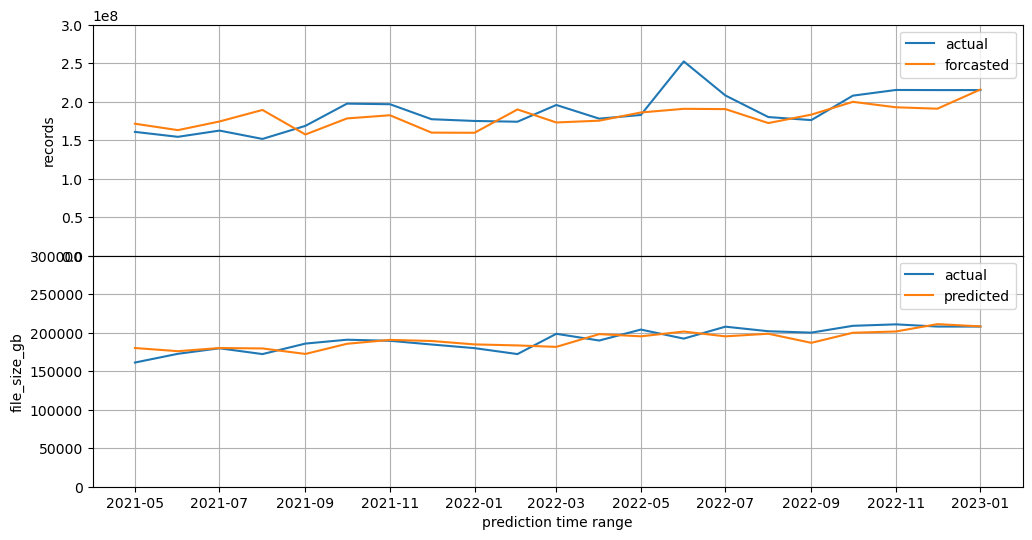


Fig. 12. SVR forecasting performance on test data

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | MAPE  *‘Records’* | MAPE  *'file\_size\_gb'* | WMAPE  *‘Records’* | WMAPE  *'file\_size\_gb'* | RMSPE  *‘Records’* | RMSPE  *'file\_size\_gb'* | Forecast for  2023-01  *‘Records’* | Forecast for  2023-01  *'file\_size\_gb'* |
| ARIMA | 0.105 | 0.064 | 0.109 | 0.061 | 0.132 | 0.077 | 1.766e+08 | 190034.679 |
| VAR | 0.091 | 0.135 | 0.090 | 0.135 | 0.107 | 0.146 | 2.138e+08 | 246684.944 |
| XGBRegressor | 0.096 | 0.046 | 0.093 | 0.048 | 0.108 | 0.056 | 2.038e+08 | 209613.61 |
| SVR | 0.104 | 0.045 | 0.105 | 0.048 | 0.113 | 0.051 | 2.09e+08 | 206835.630 |

Table 4: Testing Error metrics and final forecasting for all 4 models

**Summary**

The time series dataset has been investigated in this work including outlier detection, deletion and interpolation; statistical distribution decomposition and visualization on its global trend and local seasonality; collinearity and correlation analysis with and without time lags; stationarity tests up to the second-order differencing of all features. In order to perform forecasting for the most recent timestamp (2023-01) data entry, four different models comprising two statistical models (ARIMA and VAR) and two non-neural network models (XGBRegressor and SVR) have been tested and compared in regards to error metrics such as MAPE, WMAPE and RMSPE, where XGBRegressor and SVR grant the lowest errors particularly on the forecasting of submitted file size each month. The submitted file size in 2023 Jan are estimated to be between 200000-210000 based on the forecasting results from XGBRegressor and SVR models.

1. https://xgboost.readthedocs.io/en/stable/python/python\_api.html [↑](#footnote-ref-1)
2. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html [↑](#footnote-ref-2)