IBA Institute of Business Administration Karachi
Leadership and Ideas for Tomorrow

Project Report

Course: Introduction to Data Mining
Instructor: Dr. Sajjad Haider
Date: 31st December 2021

Ibrahim Ahmed Khan - 19683
Syed Muhammad Ahris - 18573
Huzaifa Hashim - 18591
Ahmed Abdul Ghafoor - 19735

Problem Description:

Our dataset is related to the recent and the prevalent worldwide pandemic Covid-19. The data is recently assembled and required to apply the recent advances of natural language processing and AI to generate valuable and new insights in support of the ongoing fight against this infectious disease. A rapid acceleration in new covid variants has led to the growing urgency for these approaches, to help the medical community develop answers to high priority answers by developing text and data mining tools furthermore connecting insights across and how the overall system of the world is affected.

The data is gathered across the globe and demands a series of questions which need to be sorted to understand the situation we are dealing with.

- How many cases were registered?
- Cases according to the different countries?
- The number of deaths reported due to covid?
- How many vaccinated, partially vaccinated, and unvaccinated people?
- How many people have recovered?
- How many new cases?
- What is the change in grocery, workplace, retails, transits etc?

These are some of the basic key questions and requirements to make a useful insight and good dataset for a data mining problem. The data gathered is analyzed and prepared to be then modeled to give some valuable information. How the whole system of the world is affected and how the overall economies of all the countries in the world have been affected. The overall change and modeled data will not only help us with our scientific solutions regarding covid but will predict the outcome of the overall effect on the whole world in terms of the entire system such as the GDP, trade, exchange rates, employment, and unemployment rates etc.

However, one of the biggest challenges that we must face is that a problem like this has never been dealt with digitally and no previous enough data is available to discern this issue. There are some major stakeholders such as the ministries of health, government agencies, influencers such as media, health partnerships, foundations, professional associations and WHO collaborating centers and one of the major assets is the engagement with the United Nations at the global, regional, and country level.

Since the spread of covid is evident, time is of essence which calls for an urgency to deal with this problem through such a way no humans could possibly do, and a result needs to be solved digitally through use of AI from the modeled datasets that we have mined. Some of the insights can be:

- Latest number of affected cases.
- Variations in cases at country level over the time.
- Variations in number of affected cases over the time.

Data Description:

Dataset 1 Covid Cases:

[Novel Corona Virus 2019 Dataset | Kaggle]

Columns	Description	Missing Value Percentage (%)
SNo	Serial Number for the data	0.00
ObservationDate	Date on with the values were collected	0.00
Country/Region	Countries for the data that was collected.	
Province/State	Further classification on provincial/state data for each countries	25.49
Last Update	Date on which the values were last updated	0.00
Confirmed	Cumulative data on confirmed cases.	0.00
Deaths	Cumulative data on number of deaths.	0.00
Recovered	Cumulative data on recovered cases..	0.00

Total Size: 306429

Dataset 2 Google's Mobility:

https://www.gstatic.com/covid19/mobility/Global_Mobility_Report.csv

Columns	Description	Missing Value Percentage (%)
country_region_code	Code of the country/region	0.06
country_region	Country or Region of area the data was taken	0.00
sub_region_1	Provincial/State of the country	1.66
sub_region_2	City or further classification of the province/state	16.33
metro_area	Metro area of the sub_region_2	99.46
iso_3166_2_code	Code system for countries	82.50
census_fips_code	Unique identification of countries and area.	79.02
place_id	Identification given to each place.	0.18
date	Date on which the values were collected.	0.00
retail_and_recreation_percent_change_from_baseline	Mobility in retail and recreational places	38.08
grocery_and_pharmacy_percent_change_from_baseline	Mobility in grocery and pharmacy areas	40.75
parks_percent_change_from_baseline	Mobility in parks	52.68
transit_stations_percent_change_from_baseline	Mobility around the transit stations	50.33
workplaces_percent_change_from_baseline	Mobility in workplaces.	3.66
residential_percent_change_from_baseline	Mobility in residential areas	39.47

Total Size: 8156037

Dataset 3 Vaccines:

https://raw.githubusercontent.com/govex/COVID-19/master/data_tables/vaccine_data/global_data/time_series_covid19_vaccine_global.csv

Columns	**Description**	**Missing Value Percentage (%)**
Country_Region	Country or Region of area the data was taken	0.00
Date	Date on which the values were collected.	0.00
Doses_admin	Cumulative doses administered for the particular country and its province	0.76
People_partrially_vaccinated	Number of cumulative partially vaccinated people	61.28
People_fully_vaccinated	Number of cumulative fully vaccinated people	61.28
Report_Date_String	Date on which the data was reported on	0.00
UID	Identification	0.32
Province_State	Province or State classification of the data of each country	39.46

Total Size: 135448

Combined Dataset:

Columns
SNo
ObservationDate
Country/Region
Province/State
Confirmed
Deaths
Recovered
retail_and_recreation_percent_change_from_baseline
grocery_and_pharmacy_percent_change_from_baseline
parks_percent_change_from_baseline
transit_stations_percent_change_from_baseline
workplaces_percent_change_from_baseline
residential_percent_change_from_baseline
Doses_admin
People_partrially_vaccinated
People_fully_vaccinated

Total Size: 77734

No Missing Values

List of Countries: Australia, Canada, Spain, Italy, Germany, Mexico, Chile, Japan, Peru, Colombia, India, Pakistan, Netherlands, Belgium

Date Range: For approach 1: 15/02/2020 - 29/05/2021

For approach 2: 16/02/2020 - 29/05/2021

Data Gathering and Data Pre-Processing:

Combining Datasets:

The dataset for predicting covid just had the date, location and number of cases. Predicting the number of cases on new data would be very difficult using only these attributes as there are not many factors associated with our prediction value.

To improve our dataset, we found google's mobility dataset. It notes down the change in mobility in certain areas from the previous day. For each day, for every country, and its provinces, and for some cities also.

We can combine these two dataset using the pandas library of python and using join operation on date, country, and province attributes.

```
merged_df = pd.merge(dfCovid, dfMobility,
left_on=["ObservationDate","Country/Region","Province/State"],
right_on=["date","country_region","sub_region_1"])
```

After combining these two dataset, the other factors that we thought that could affect the number of covid cases is the vaccination record of a country. Then after finding the vaccination data of a country then we needed to combine this new dataset with the one we created with the mobility. But the data we could find were just based on the country and not further classified based on the provincial data. Hence we decided to use just the country data to assign it to the every provincial values. So for every province of a country it would be the same values which were of the whole country. We joined these data again using the same join operation.

```
withVaccines = pd.merge(merged_df, dfVaccines,
left_on=["ObservationDate","Country/Region"], right_on=["Date","Country_Region"],
how="left", indicator=True)
```

Missing Values:

The vaccination data was not of all the dates, because vaccinations were administered in the 2nd quarter of this year. So by using the left join the dates for which the vaccination data was not present, were also picked up by the join operation. As technically the number of vaccines administered was zero so the missing values were filled with zero for these columns.

```
withVaccines['Doses_admin'] = withVaccines['Doses_admin'].fillna(0)
withVaccines['People_partially_vaccinated'] =
withVaccines['People_partially_vaccinated'].fillna(0)
withVaccines['People_fully_vaccinated'] =
withVaccines['People_fully_vaccinated'].fillna(0)
```

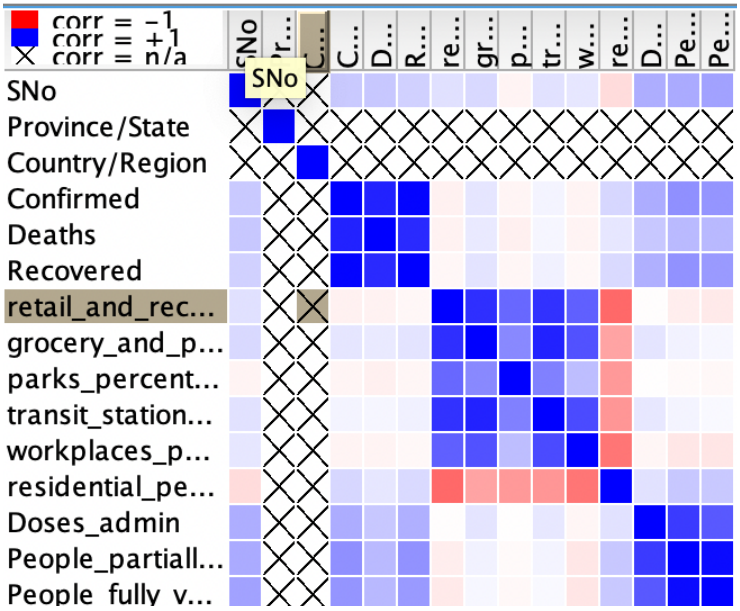
Some unnecessary columns were removed with the dataframes, to further clean our data.

```
merged_df.drop('census_fips_code', inplace=True, axis=1)
merged_df.drop('Last Update', inplace=True, axis=1)
withVaccines.drop('UID', inplace=True, axis=1)
withVaccines.to_csv('withVaccines.csv')
```

For another approach we wanted to combine the datasets so that for the current day, the values of mobility and vaccination were of the previous day. For this purpose before all the merge operations, we added a day to the dates of the mobility data and vaccination data.

```
dfVaccines['Date'] = dfVaccines['Date'] + pd.DateOffset(days=1)
dfMobilityReport['date'] = dfMobilityReport['date'] + pd.DateOffset(days=1)
```

Linear Correlation:



The data was further analyzed to see if there were any existing correlations within it. As can be seen, there are very strong correlations between confirmed cases, deaths and recovered(our dependent variables), which is obvious. However, it is interesting to note that mobility data also was significantly correlated with one another, except for the case of residential mobility. Furthermore, the three columns with vaccination data were all highly correlated with one another. To remove unnecessary columns, i.e. those with an excessively high correlation, we applied a linear correlation filter, with a correlation cutoff of 0.85, which removed transit_stations mobility data, as well as people fully vaccinated. However when training the data, we will still consider these at times in case they may add to the data. The difference in the model accuracy due to this was negligible however, due to which this correlation filter was not considered.

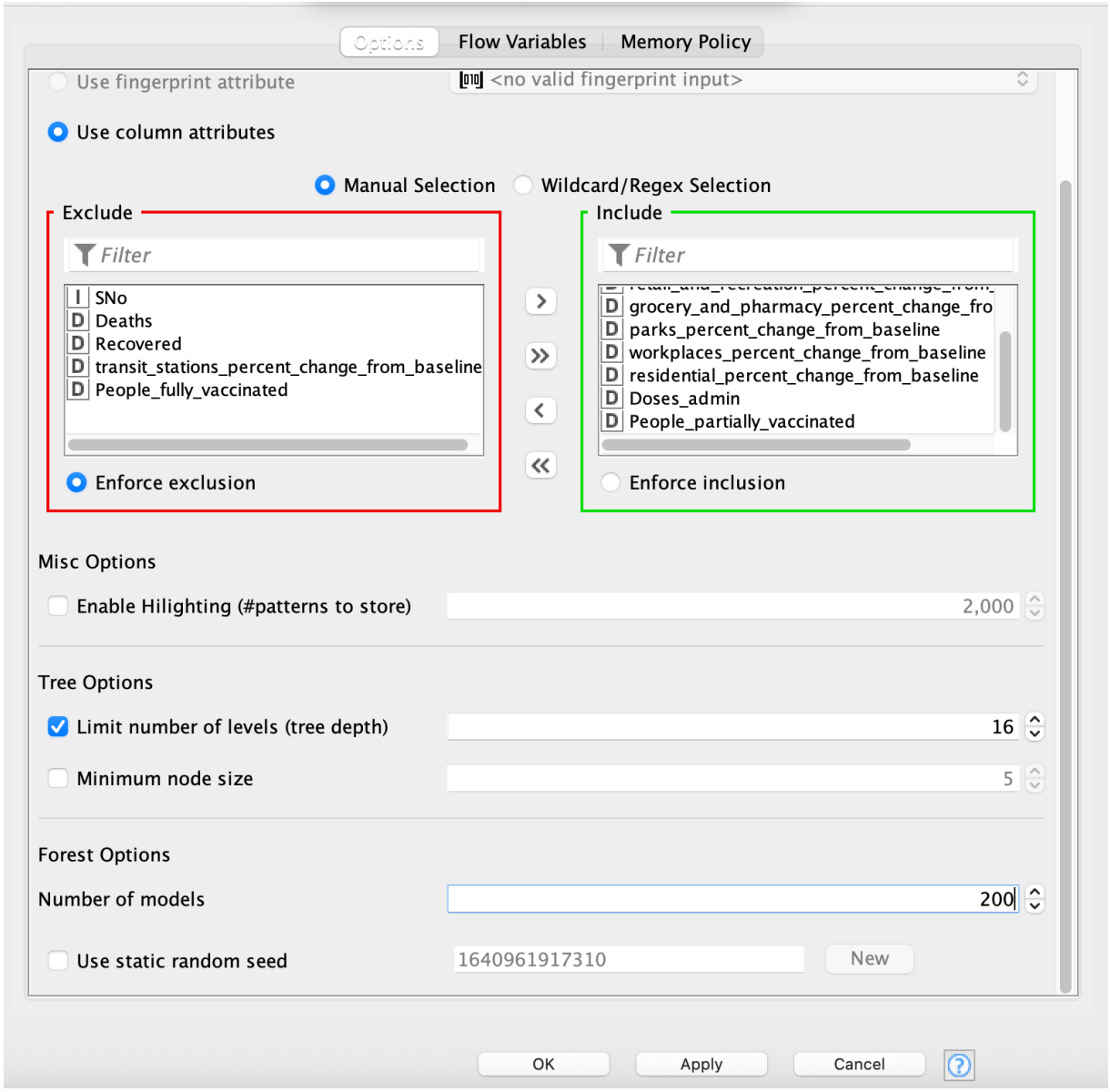
Data Modeling :

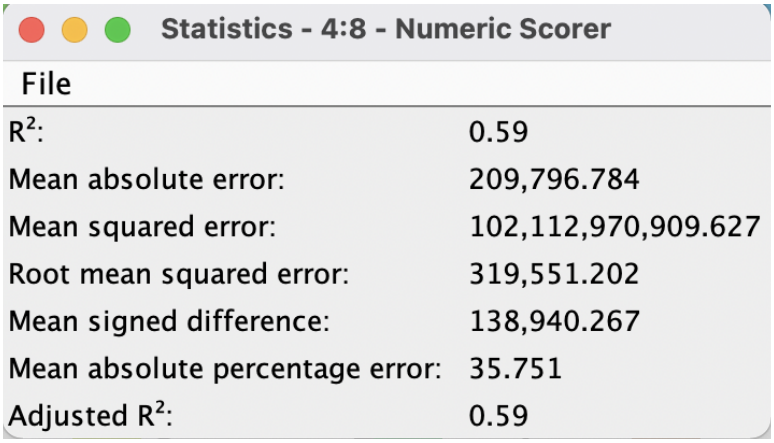
There were a total of 11 columns to choose from, from which 3 were interchangeably used as dependent variables, ie, deaths, confirmed and recovered. Out of these three our main predictions were on the confirmed column.
The data was split with 95% training data and 5% test data, and was split in order i.e., the last 5% of the data was used as the testing data, in accordance with the requirements of trend analysis.
The data was analyzed by a wide variety of models for regression, specifically:

Processor: Apple M1 Chip
Ram: 8GB
Knime Version: 4.5.0

1.Random forest regression

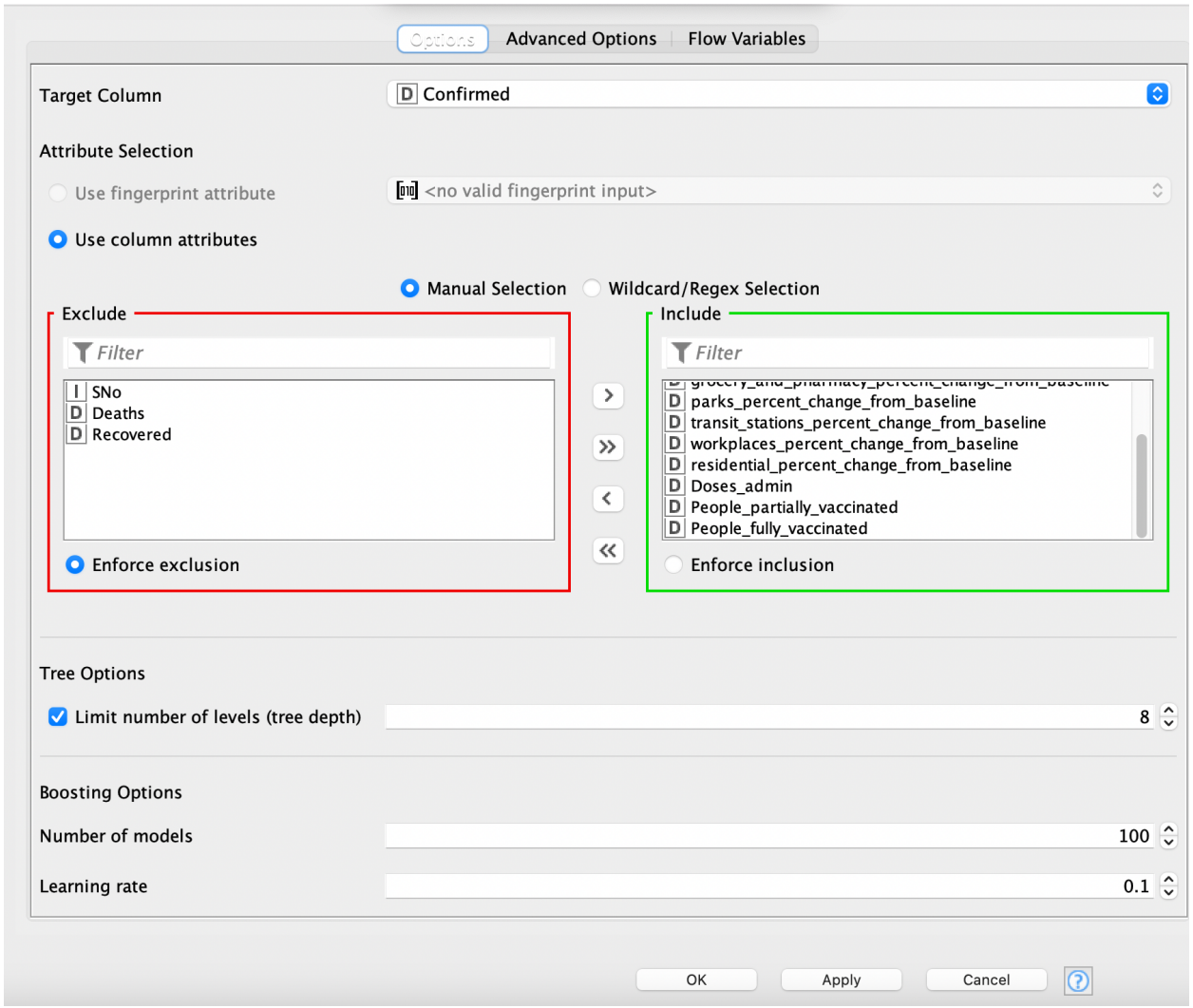
With the random forest tree learner, the values we got initially were reasonably well, with accuracies ranging(of adjusted R2) from 0.1 to 0.59 , and MAPE of 35.571(best) depending on the tuning of parameters. This model did not perform well at all, no matter how much we tuned the parameters, regardless of the columns that were included or excluded.
Below is an example of one such configuration that was performing the best relative to other configurations. This gave a time value of 18123 ms. The parameters were 200 models, with a tree depth of 16. Other tested parameters included tree depths of 4, 8, 12, with model sizes from 10, to 50 to 100, to 500, to 1000.

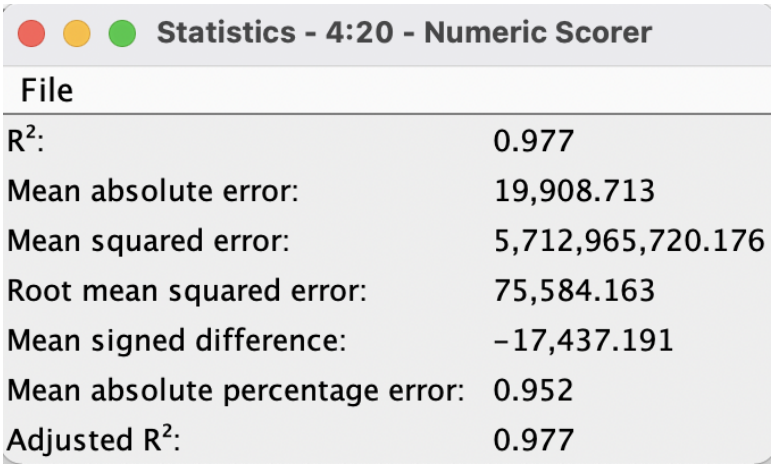

Options
Flow Variables
Memory Policy
Use fingerprint attribute
<no valid fingerprint input>
Use column attributes
Manual Selection
Wildcard/Regex Selection
Exclude
Filter
SNo
Deaths
Recovered
transit_stations_percent_change_from_baseline
People_fully_vaccinated
Enforce exclusion
Include
Filter
grocery_and_pharmacy_percent_change_fro
parks_percent_change_from_baseline
workplaces_percent_change_from_baseline
residential_percent_change_from_baseline
Doses_admin
People_partially_vaccinated
Enforce inclusion
Misc Options
Enable Hilighting (#patterns to store)
2,000
Tree Options
Limit number of levels (tree depth)
16
Minimum node size
5
Forest Options
Number of models
200
Use static random seed
1640961917310
New
OK
Apply
Cancel


Statistics - 4:8 - Numeric Scorer
File
R²: 0.59
Mean absolute error: 209,796.784
Mean squared error: 102,112,970,909.627
Root mean squared error: 319,551.202
Mean signed difference: 138,940.267
Mean absolute percentage error: 35.751
Adjusted R²: 0.59

2.Gradient boosted tree learner regression

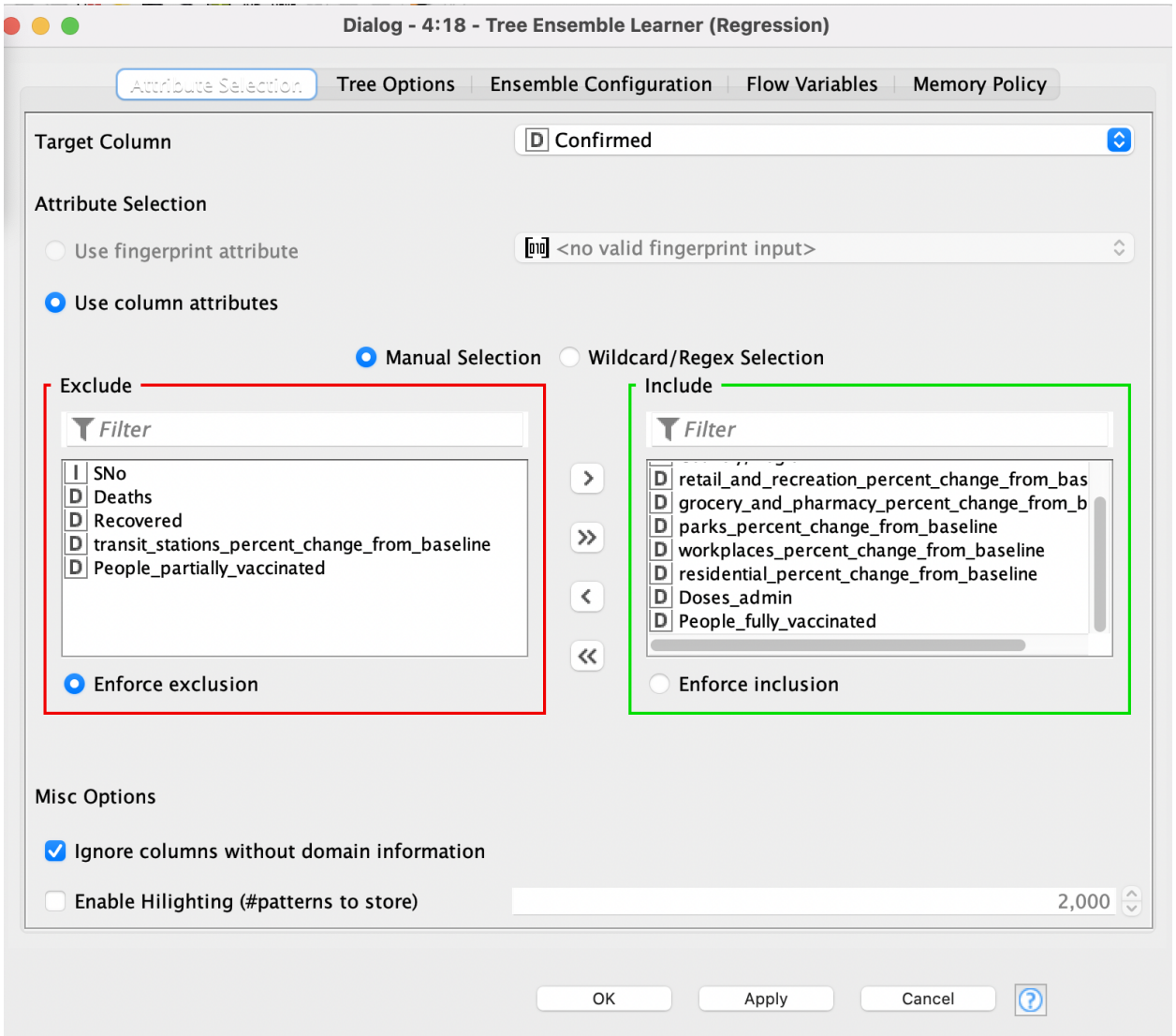
This predicted our outputs relatively better to the random forest learner.
On average most models were in the range from 0.8 to 0.977. The best model gave an adjusted R2 of 0.977, with an MAPE of 0.952. The parameters used for this were a tree depth 8, number of models being 100, and a learning rate of 0.1. Keep in mind that since the vaccination data was only available for the last few months, the data was heavily influenced by the partitioning percentage, ie, when we take a partitioning percentage of 80 20, where 20 is our test data size, the accuracy never goes above 90, however, when you move the data to a 95 5 split, the accuracy boosts to .977 at maximum, and ranges within the 90s for most parameters.

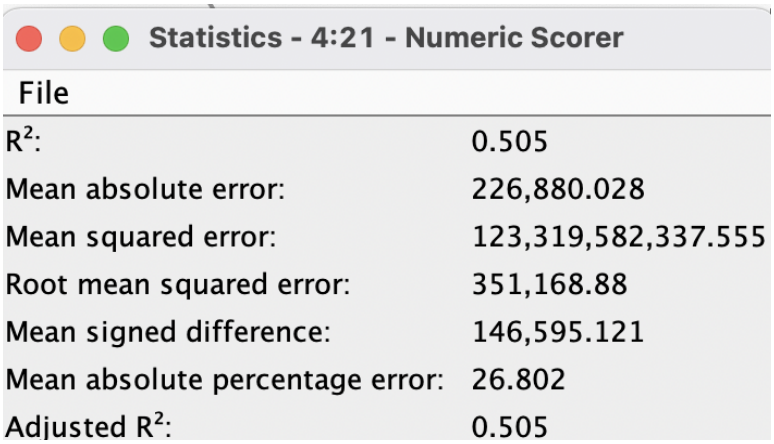



Statistics - 4:20 - Numeric Scorer
File
R^2: 0.977
Mean absolute error: 19,908.713
Mean squared error: 5,712,965,720.176
Root mean squared error: 75,584.163
Mean signed difference: -17,437.191
Mean absolute percentage error: 0.952
Adjusted R^2: 0.977

3.Tree ensemble predictor regression

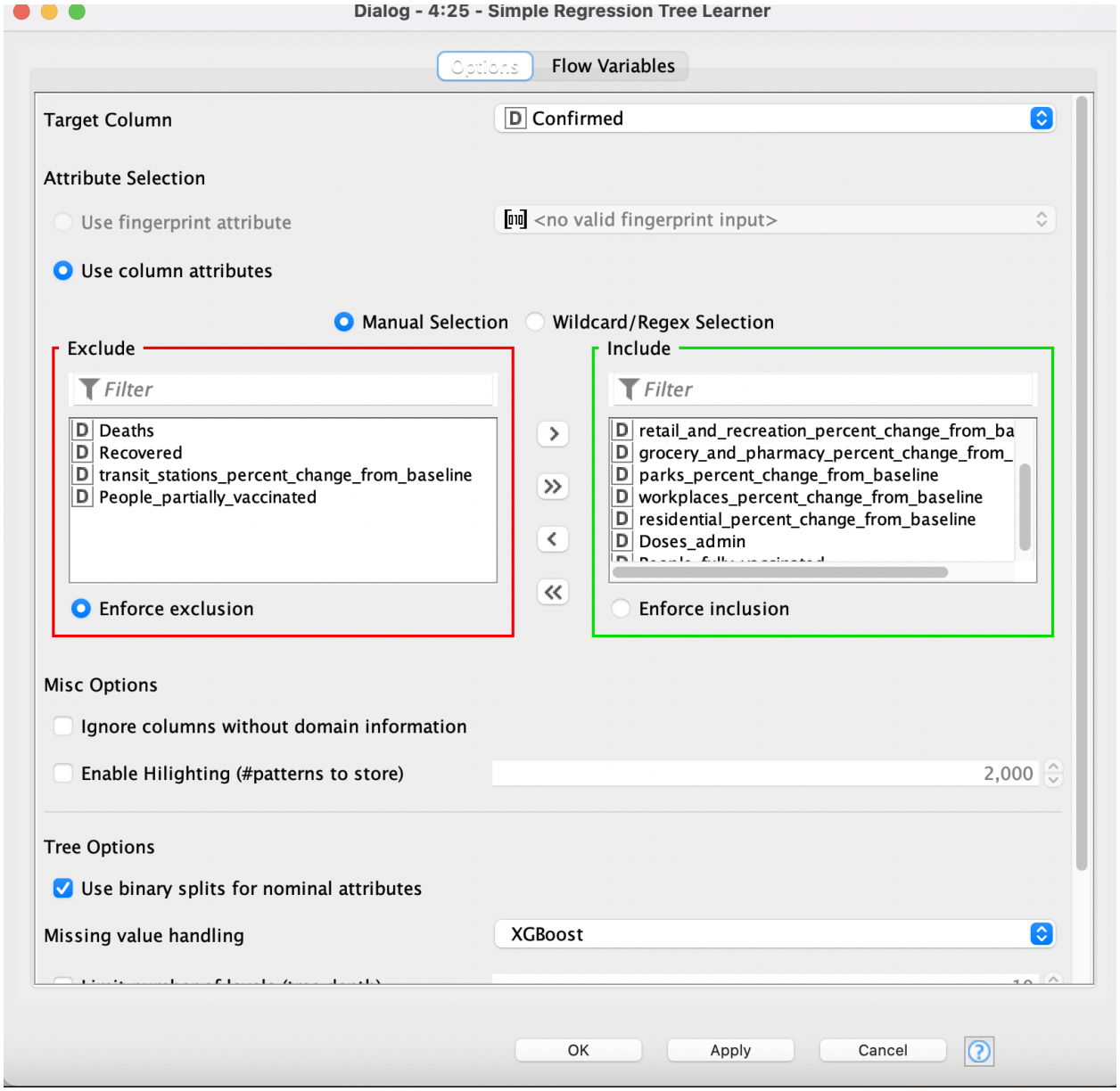
For this model, the predictions were similar to the random forest learner, and as such they were mediocre at best. They ranged in value from 0.4 to 0.5, maxing out at 0.505, with the presence of tree limit of 8. Interestingly tho, this value tended to deviate quite a bit, upon repeating with the same parameters. The time it took to run this was 14696ms.
Attached below are some screenshots of the outputs.

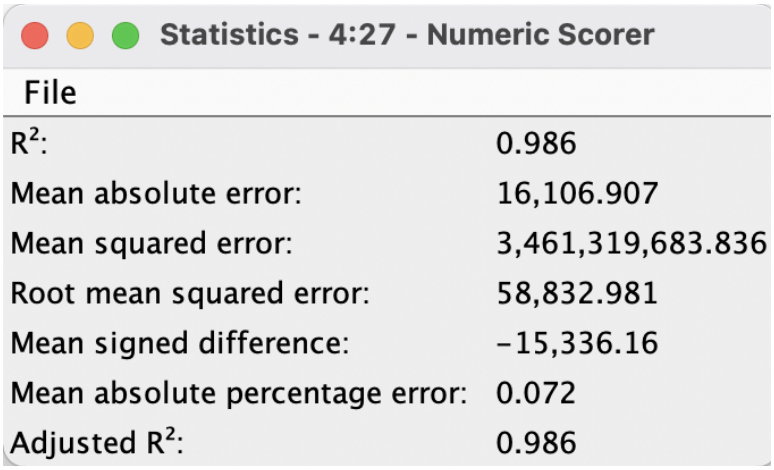



Statistics - 4:21 - Numeric Scorer
File
R^2: 0.505
Mean absolute error: 226,880.028
Mean squared error: 123,319,582,337.555
Root mean squared error: 351,168.88
Mean signed difference: 146,595.121
Mean absolute percentage error: 26.802
Adjusted R^2: 0.505

4.Simple regression tree learner

This model was the best out of all the models, and performed very well, over a large range of parameters and selected columns. It provided adjusted R2 values of upto 0.986 with a MAPE of 0.072, while most values were in the range of 0.9 to 0.97. This was the model which we selected for our recommendations.

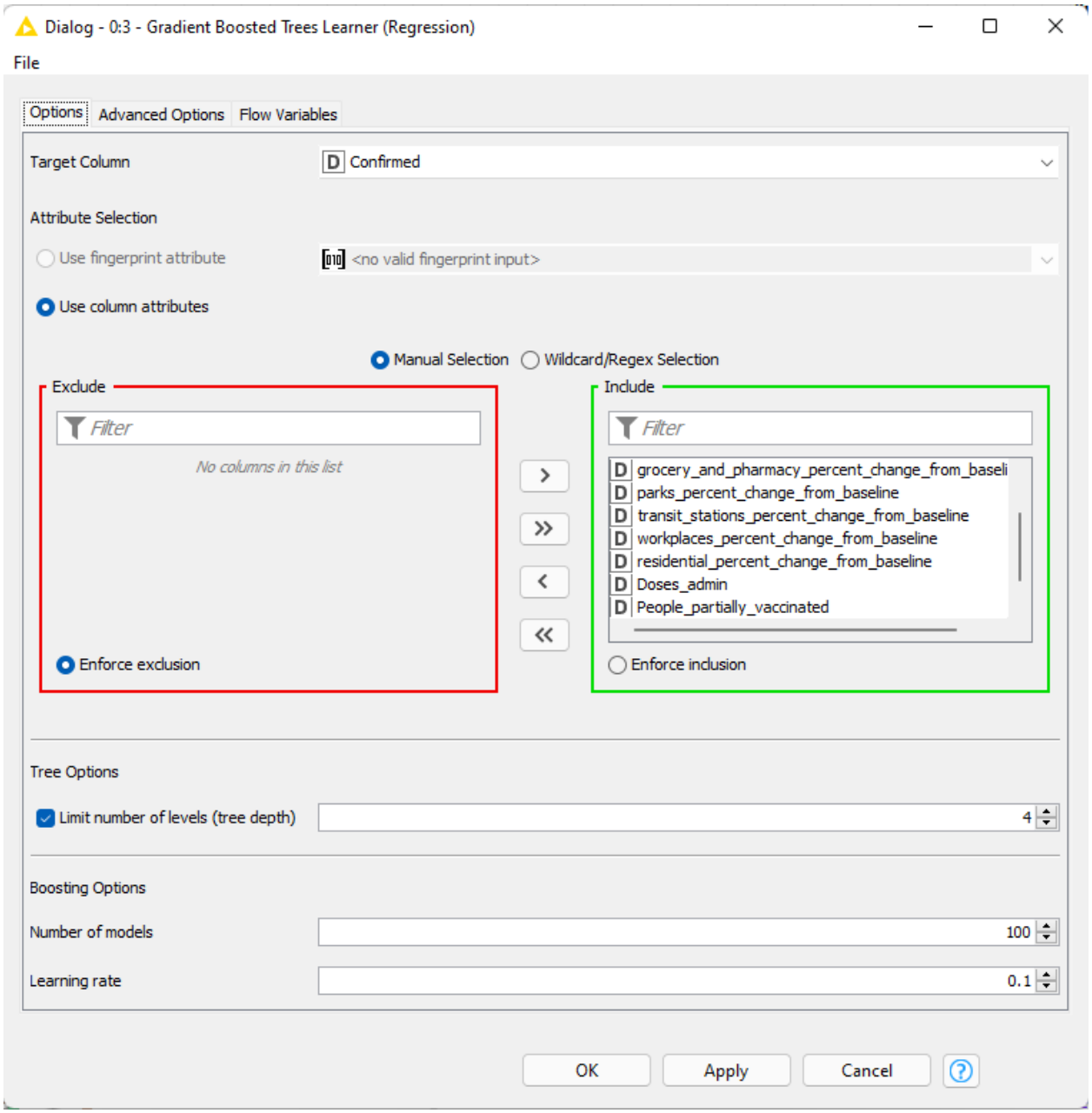


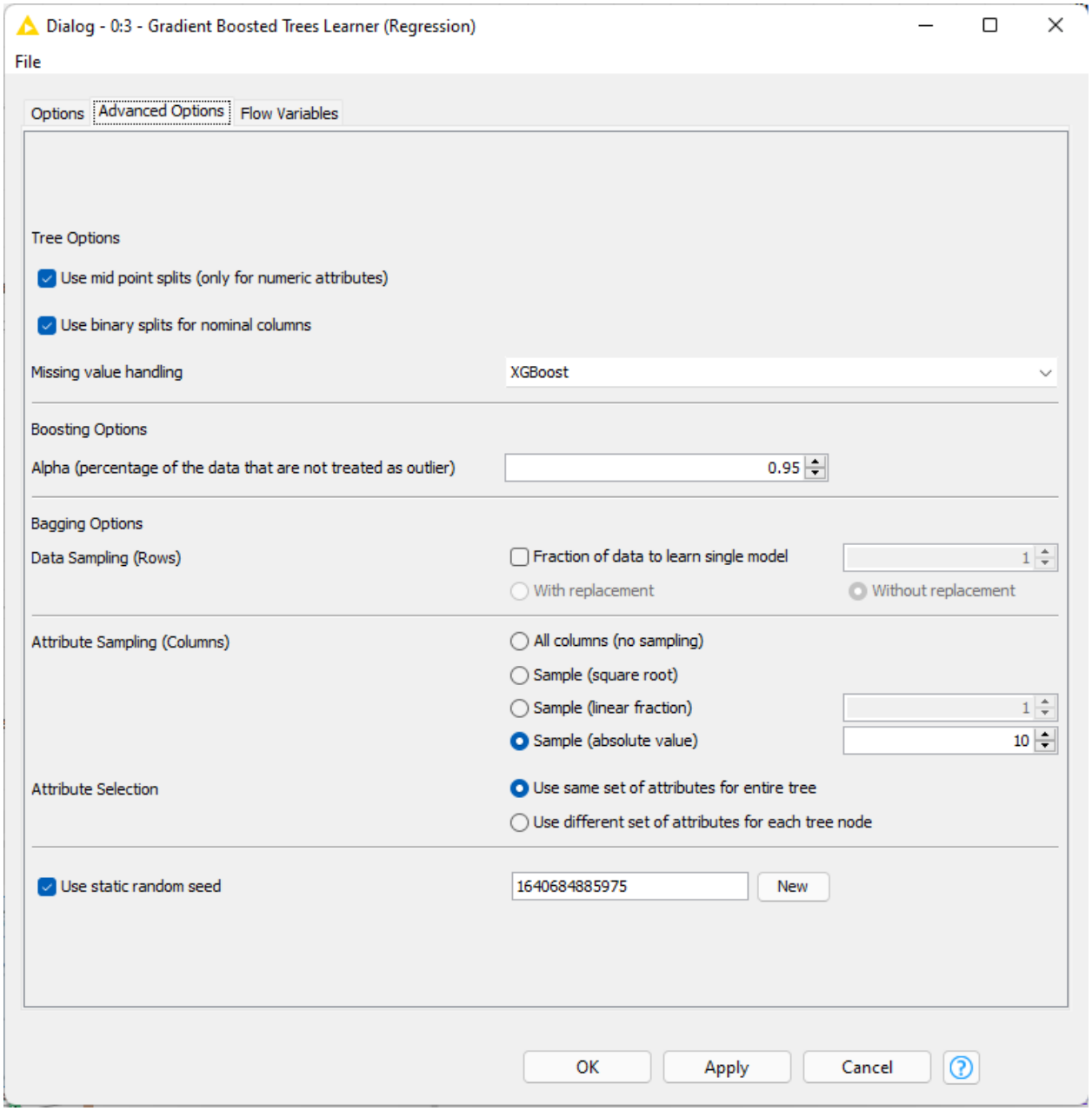


Principal Component Analysis:

As we had 9 columns some which had high collinearity in between each other. We decided to apply PCA to see if it could help with reduction in columns, consequently resulting in less runtime.Gradient boosted learner for regression was chosen as a good option as it had the longest executing time.

Before Applying PCA:



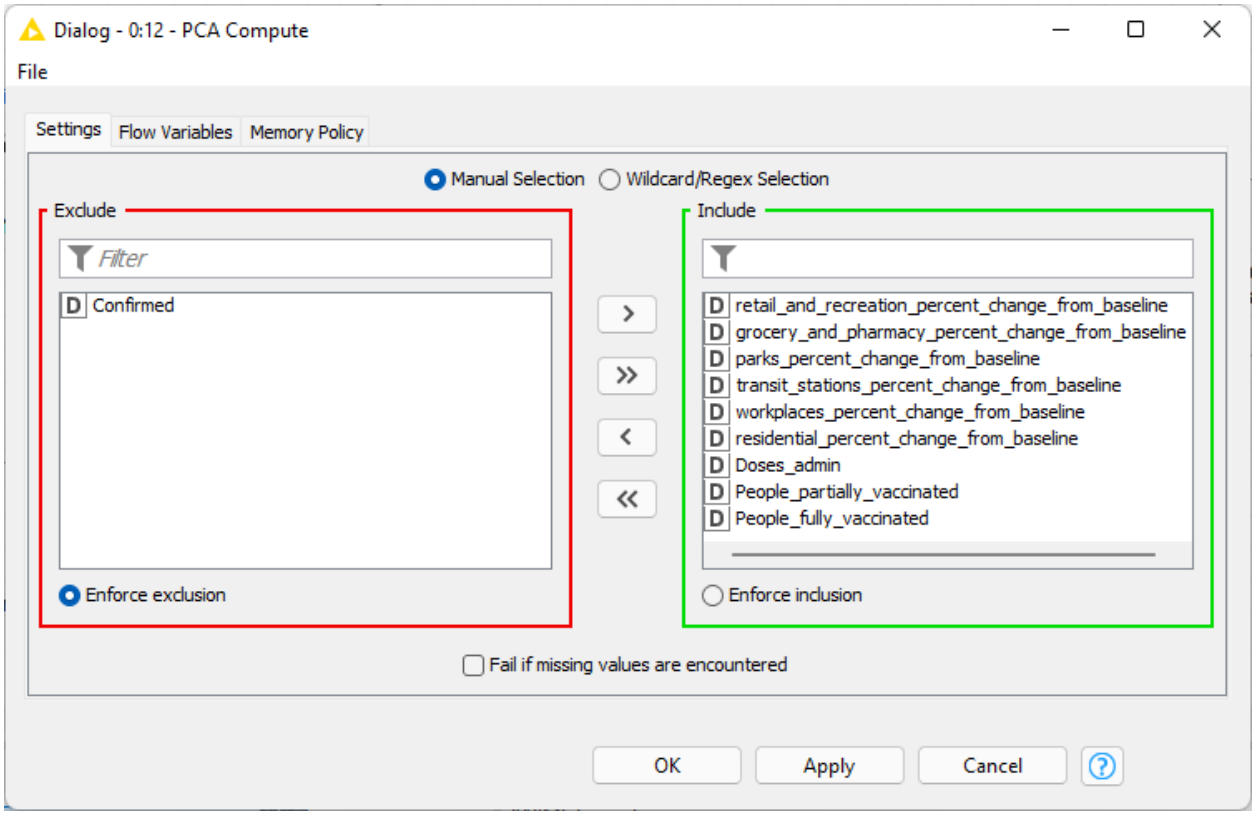


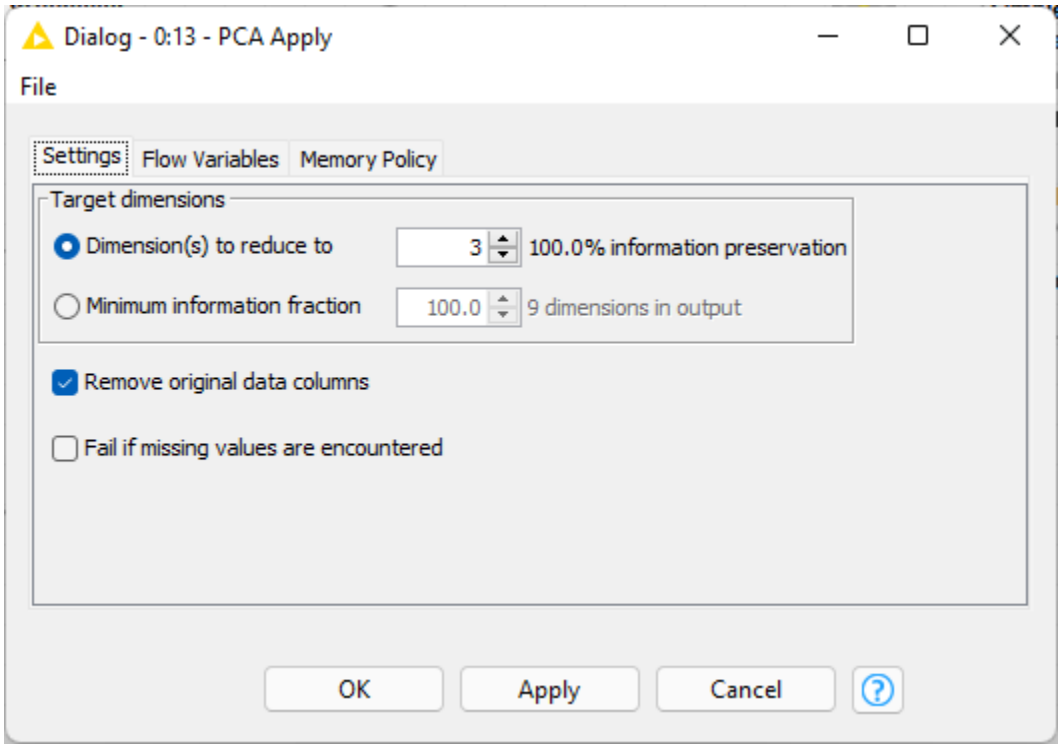
Gradient boosted tree for regression's learner was run on the settings as shown above, without the application of PCA. This gave us the following evaluations.

Row ID	D Prediction (Confirmed)
R^2	0.915
mean absolute error	32,481.277
mean squared error	21,236,910,130.347
root mean squared error	145,728.893
mean signed difference	-26,241.975
mean absolute percentage error	1.427

The execution time of the learner's node and the predictor's node in this case was 20173 and 91 respectively, with total time resulting in 20264.

After Applying PCA:





After computing PCA on every numerical column, besides confirmed cases columns as it is our prediction column, PCA application reduced the 9 columns to 3 with 100 percent information preservation. It gave 99.8% preservation on 2 dimension reduction, and 88.2 on 1 dimension reduction.
Now on this data when Gradient boosted trees for regression's learner was run it gave the following evaluations.

Row ID	D Prediction (Confirmed)
R^2	0.913
mean absolute error	35,644.528
mean squared error	21,616,359,560.911
root mean squared error	147,025.03
mean signed difference	-20,270.235
mean absolute percentage error	2.683

The difference in evaluations is negligible when compared in both cases. But when runtimes are considered the difference becomes significant. The time taken for PCA compute, PCA apply, learner, and predictor was 279, 706, 12137, 87, with total time combining to be 13207. This was approximately 1.5 times faster than without applying PCA. No this was just for 100 models, if ran on a higher number of models this rate might increase.
The specifications of system on which PCA part was ran is:
Processor: Xeon W3565
Ram: 8GB DDR3
GPU: AMD Radeon Rx570
Knime Version: 4.4.1

Findings and Insights:

The reason for designing an analytical model to predict the number of Covid Cases on a particular day/date was to help the major stakeholders such as the ministries of health, government agencies, foundations and countries to curb the growth of the Virus. The reason for the coronavirus outbreak being so lethal was that people were unprepared and did not know what will happen or can happen in the next few days. It was impossible to encompass and predict the growth of the virus accurately due to which it resulted in shortage of hospital beds, oxygen and necessary health supplies and a huge impact on the economy. By working on an analytical approach to predict the cases of Coronavirus we aim to help all stakeholders to be better equipped and prepared for the lethal wave of Coronavirus.

One very interesting insight was the presence of the variable observation date. Initially, we thought this variable was very significant to contributing to the output, and hence we included it in most of our models. This was giving rather bad accuracies, compared to when we excluded it, which sounded counterintuitive as why would the date, which was a primary determinant of covid cases, actually reduce the accuracy? The answer to that was because realistically, the cases have nothing to do with the current date. Instead they are merely based upon the cases of the previous days and the mobility data of the previous days, and the vaccination data of the previous days. And so, even if you exclude the date column entirely, and instead use the previous day cases, which we did in many models, the accuracy comes out a lot better, which makes sense, because realistically, how would you have the mobility, vaccination data for a theoretical date in the future?

However, there does exist a workaround for this flaw, and that is the use of theoretical values. So lets say for instance we wish to predict the cases for two months from today. SInce the information ie mobility data and vaccination data is not available for that future data, we can merely create our own theoretical mobility and vaccination data, essentially saying that if so and so vaccinations are done by the given time, and mobility is limited to so and so , the cases would be __, otherwise they would be __(where __ indicates the output predictions for that value). Hence our system is designed to work on theoretical data values.

Another noteworthy observation was that, if we excluded the earliest few months from our model, the predicted values became a lot better, The reason for this was that the vaccination data is only available for the last few months, so by removing the data from the first few months, we have a dataset that consists of a higher percentage of rows with vaccination data, meaning that the more initial rows we exclude, uptill a certain extent, the better the accuracy.

The way forward in future for organizations who want to implement this prediction model would be to use the findings and insights gained from the predictions of this model to as closely prepare:

1. The number of hospital beds capacity.
2. Impose bans on public gatherings and lockdown before the situation gets out of control.

3. Better equip the health supplies and oxygen needed for the number of predicted cases.
4. Use the number of cases to accurately manage the amount of workforce in offices and be prepared.

The insights gained from this prediction can be invaluable in a plethora of domains and if we use it carefully it can be even used to predict the cases of other contagious diseases except Coronavirus.

Data Limitations:

The dataset for predicting covid just had the date, location and number of cases. Predicting the number of cases on new data would be very difficult using only these attributes as there are not many factors associated with our prediction value. To help our regression model we combined different factors like mobility rate and number of vaccinations as discussed in the Data Gathering section. For the future the best advice for data collection would be to have a combined dataset with more factors to help predict the number of cases, factors like mortality rate and Life expectancy could further help in understanding the strength of immune systems of the people in a particular region. Other than that as evident from the new mutations of coronavirus evolving it is very hard to as accurately predict the action and spread of these viruses. So our data model is limited to the current 1st mutation of coronavirus and it could be possible that it is unable to give an accurate representation of the expected number of cases if more mutations continue to evolve.

Model Deployment

With KNIME Server, the designed regression model workflow can be hosted using KNIME Server in your data center or in the cloud via Microsoft Azure, Amazon AWS, or the cloud provider of your choice.. Data scientists within the organization will deploy the workflow to KNIME Server. End users can then interact with the workflow on the web in a controlled way and view results. Later you can build and publish detailed reports which can be sent via email or accessed on demand from the KNIME WebPortal.