UNIVERSITI TUNKU ABDUL RAHMAN
Wholly owned by UTAR Education Foundation
(Co. No. 578227-M)
DU012(A)

UNIVERSITI TUNKU ABDUL RAHMAN

LEE KONG CHIAN FACULTY OF ENGINEERING & SCIENCE (LKCFES)

Course Description	UECS2053 Artificial Intelligence
Title	Lab Report 3: Supervised Learning
Lecturer	Dr Shalini a/p Darmaraju
Practical Group	P3-G2
Date of submission	29/08/2024

Group Member:

No	Name	ID	Course	Year/Trimester
1	Lai Jien Weng	2104338	AM	Y2T3
2	Tan Wan Xuen	2207214	AM	Y2T3
3	Janice Ng Zhi Yan	2104783	AM	Y2T3
4	Ooi Xin Yi	1902703	BI	Y4T2

Table of Contents

1.0 Introduction 3

2.0 Data Preparation 3

2.1 Data Loading and Initial Exploration 3

2.2 Date Filtering 3

2.3 Data Aggregation 3

2.4 Data Merging 3

2.5 Exploratory Data Analysis (EDA) 3

Heatmap Analysis 4

Scatter Plot 5

Bar Plot 5

Line Chart 5

3.0 Model Creation 6

3.1 Mutiple Regression 6

3.2 Keras Sequential Neural Network 6

3.3 Random Forest 7

3. 4 XGBoost 7

3.5 Support Vector Regression (SVR) 8

4.0 Conclusion 8

References 10

1.0 Introduction

The Covid-19 pandemic has significantly impacted global health systems, prompting extensive research into its effects and the factors influencing disease outcomes. This report outlines the methodology and findings of a supervised learning project aimed at predicting new Covid-19-related deaths in Malaysia using various features derived from Covid-19 datasets. The objective is to utilize machine learning techniques to analyze the relationships between different variables and their impact on mortality rates during the pandemic.

The datasets utilized for this analysis, including “cases_malaysia.csv”, “deaths_malaysia.csv”, “icu.csv”, and “pkrc.csv”, sourced from the Malaysian Ministry of Health's public GitHub repository. The focus is on predicting the “deaths_new” variable from the deaths_malaysia.csv dataset, leveraging features from the other three datasets. The analysis will specifically cover data from June 1, 2021, to January 1, 2022, spanning 6 months.

2.0 Data Preparation

2.1 Data Loading and Initial Exploration

The first step involved loading the relevant datasets into a Jupyter Notebook using Pandas. The datasets were examined for their structure, including the number of rows, columns, and data types.

pkrc.csv	icu.csv	cases_malaysia.csv	deaths_malaysia.csv

2.2 Date Filtering

The data was filtered from June 1, 2021, to January 1, 2022, ensuring a consistent timeframe for the analysis.

2.3 Data Aggregation

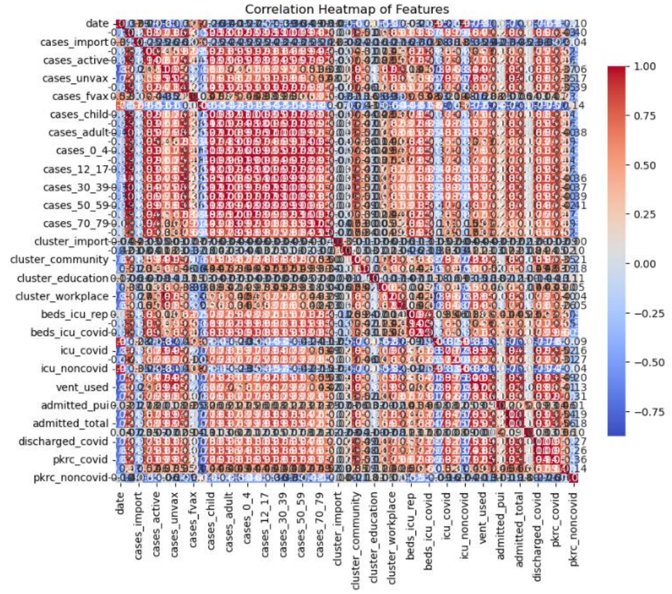
The "cases" and "deaths" datasets have 215 rows, while "icu" and "pkrc" have 3,440 rows due to state-level recordings. To align these datasets, "icu.csv" and "pkrc.csv" were aggregated by date using Pandas, summing relevant features to reflect total daily counts.

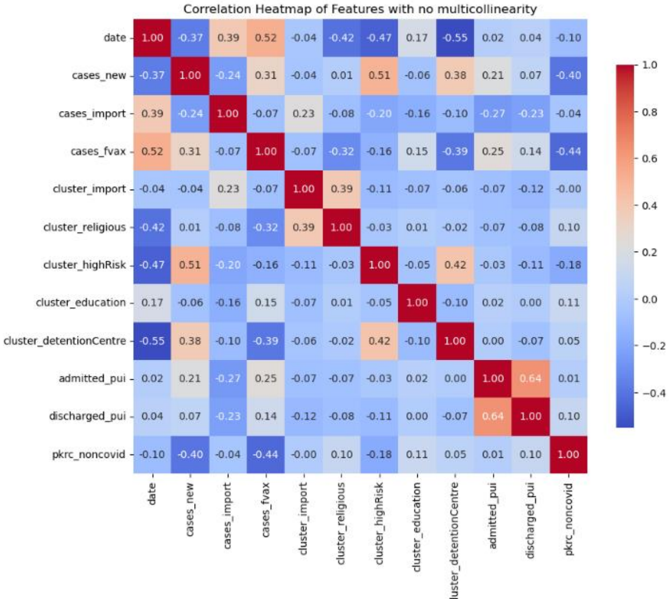
2.4 Data Merging

The datasets were merged sequentially using outer joins based on the 'date' column to ensure all unique dates were included. The final merged data frame, 'df,' contains all necessary features for analysis.

2.5 Exploratory Data Analysis (EDA)

With the merged data frame containing a total of 52 features, we recognized the potential for multicollinearity among the variables. Multicollinearity occurs when two or more predictor variables in a statistical model are highly correlated, which can lead to unreliable coefficient estimates and affect the model's performance.





Heatmap Analysis

A heatmap of the correlation matrix revealed potential multicollinearity among variables. We set a correlation threshold of 0.7, reducing the features from 52 to 11 by retaining only the first feature in highly correlated pairs. Here's the features selected for modelling:

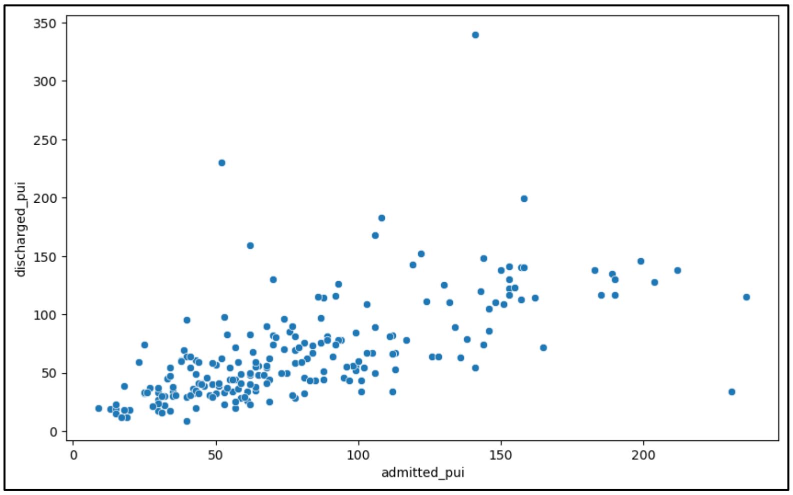
Feature Name	Description
cases_new	Total new Covid-19 cases reported daily.
cases_import	Number of imported Covid-19 cases.
cases_fvax	Covid-19 cases among fully vaccinated individuals.
cluster_import	Cases originating from imported sources.
cluster_religious	Cases linked to religious clusters.
cluster_highRisk	Cases associated with high-risk groups or environments.
cluster_education	Cases originating from educational institutions.
cluster_detentionCentre	Cases linked to detention centers.
admitted_pui	Number of patients admitted under investigation (PUI).
discharged_pui	Number of PUIs discharged.
pkrc_noncovid	Non-Covid patients in PKRC (low-risk quarantine centers).

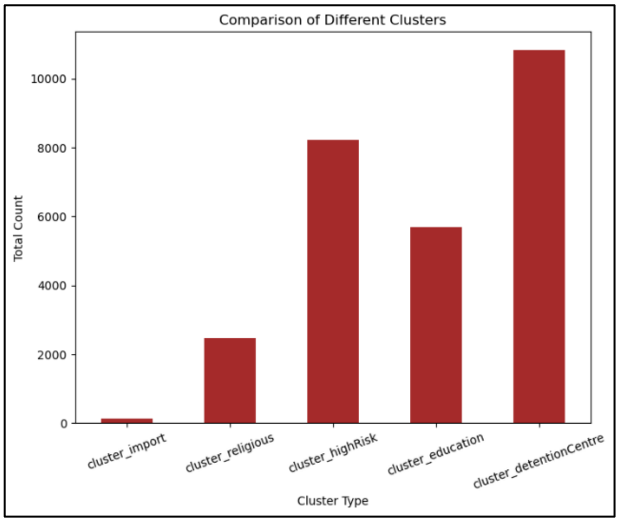
Target Selected

Feature Name	Description
deaths_new	Number of new death cases reported based on date reported to public.

Scatter Plot

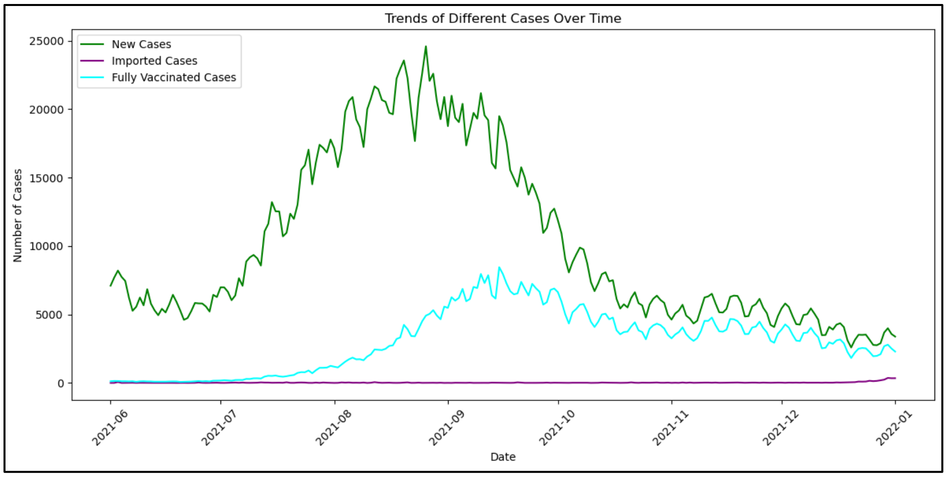
We explored the relationship between 'admitted_pui' and 'discharged_pui', finding a moderate correlation (0.64). Since it’s below the multicollinearity threshold (0.7), both features were retained, though they may be revisited if issues arise in the final model.





Bar Plot

The bar plot revealed key insights. Cluster Import had minimal impact, while Cluster Religious had moderate influence. Clusters like High Risk, Education, and especially Detention Centre, which had the highest case count, are critical for understanding Covid-19 death trends and were retained in the model.



Line Chart

New Cases: Increased from June 2021, peaking in September, then declining.

Fully Vaccinated Cases: Rose from July 2021, reflecting breakthrough infections.

Imported Cases: Remained minimal throughout.

These trends show how pandemic dynamics changed during the study.

2.6 Data Standardization

Z-score normalization was applied to standardize the data, ensuring all features contributed equally to the model and improving the convergence rate of gradient-based optimization algorithms.

2.7 Splitting Data into Training and Test Sets

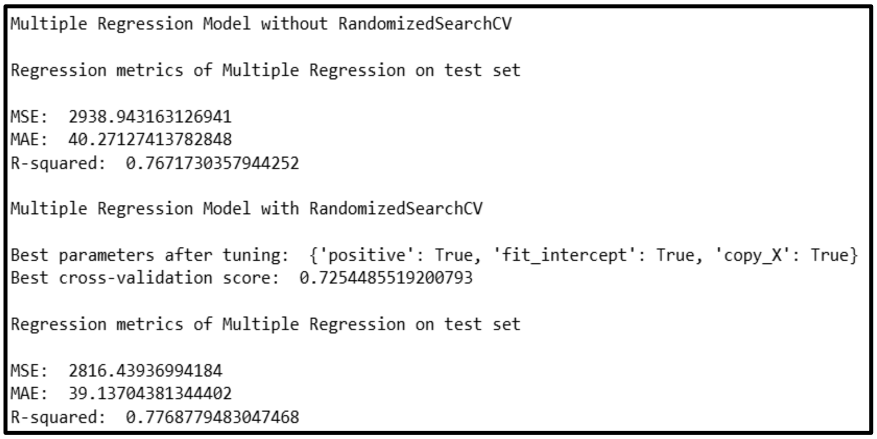
The dataset was split into an 80:20 ratio for training and testing. This split ensured sufficient data for training while preserving enough data to assess the model's generalization capability.

2.8 Cross-Validation on Training Data

4-fold cross-validation was performed on the training data to evaluate the model's performance. This process provided a reliable estimate of the model's generalizability and helped identify potential overfitting or underfitting issues.

3.0 Model Creation

3.1 Multiple Regression

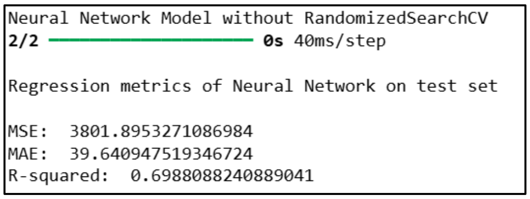


```
Multiple Regression Model without RandomizedSearchCV

Regression metrics of Multiple Regression on test set

MSE:  2938.943163126941
MAE:  40.27127413782848
R-squared:  0.7671730357944252

Multiple Regression Model with RandomizedSearchCV

Best parameters after tuning:  {'positive': True, 'fit_intercept': True, 'copy_X': True}
Best cross-validation score:  0.7254485519200793

Regression metrics of Multiple Regression on test set

MSE:  2816.43936994184
MAE:  39.13704381344402
R-squared:  0.7768779483047468
```

A statistical method named multiple linear regression (MLR), or just multiple regression, makes use of many explanatory variables to forecast the value of a response variable. Modelling the linear relationship between the response (dependent) variables and the explanatory (independent) variables is the aim of multiple linear regression (MLR). Since multiple regression uses more than one explanatory variable, it is essentially an extension of ordinary least-squares (OLS) regression (Hayes, 2024).

Without RandomizedSearchCV, the model’s performance on the test set shows a Mean Squared Error (MSE) of 2938.94, a Mean Absolute Error (MAE) of 40.27, and an R-squared value of 0.767. On the test set, the tuned model indicates a slight improvement in the model’s accuracy and predictive power, explaining about 77.7% of the variance with a lower average error of 39.14 units. Overall, hyperparameter tuning with RandomizedSearchCV has enhanced the model’s performance, making it more accurate and reliable for predictions.

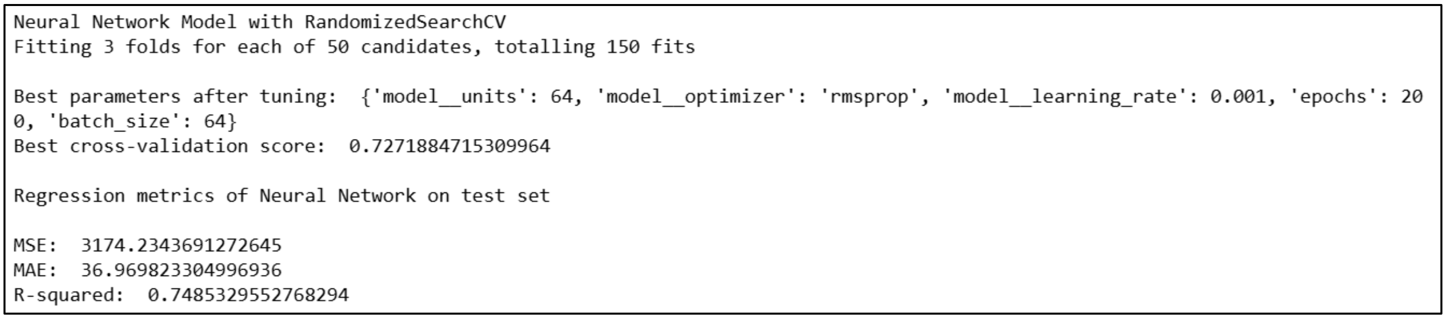
3.2 Keras Sequential Neural Network



```
Neural Network Model without RandomizedSearchCV
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step

Regression metrics of Neural Network on test set

MSE:  3801.8953271086984
MAE:  39.640947519346724
R-squared:  0.6988088240889041
```

Keras is a well-known open-source library for training neural networks. The Sequential class in Keras is a simple and straightforward way to build neural networks that allows us to create a model by stacking layers on top of each other in a linear fashion (Lazy Programmer, 2023). The results for the basic neural network model are shown in the figure above.

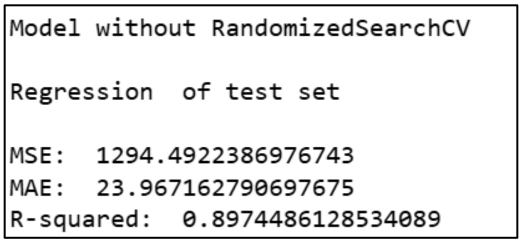
Below are the results from neural network model with RandomizedSearchCV.



```
Neural Network Model with RandomizedSearchCV
Fitting 3 folds for each of 50 candidates, totalling 150 fits

Best parameters after tuning:  {'model__units': 64, 'model__optimizer': 'rmsprop', 'model__learning_rate': 0.001, 'epochs': 20
0, 'batch_size': 64}
Best cross-validation score:  0.7271884715309964

Regression metrics of Neural Network on test set

MSE:  3174.2343691272645
MAE:  36.969823304996936
R-squared:  0.7485329552768294
```

The R-squared means that approximately 74.85% of the variance in the target variable can be explained by the model. Although neural networks are complex and computationally expensive, they are flexible and can dynamically pick the best type of regression, and if that is not enough, hidden layers can be added to improve prediction (GeeksforGeeks, 2024).

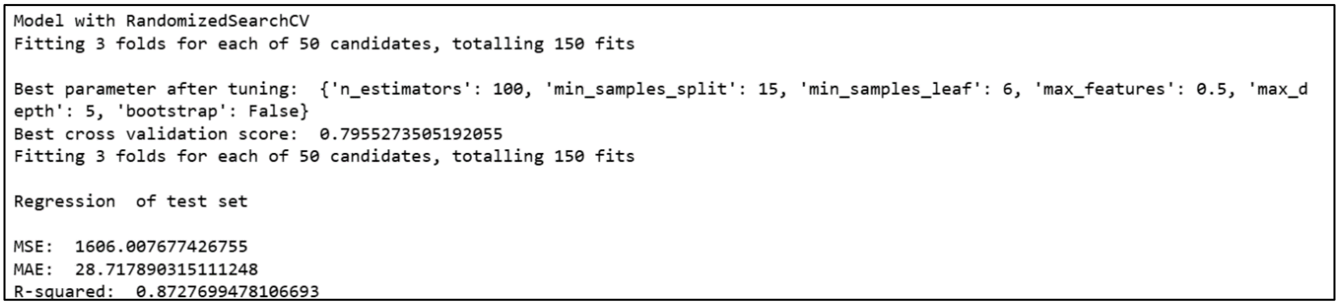
3.3 Random Forest



```
Model without RandomizedSearchCV

Regression  of test set

MSE:  1294.4922386976743
MAE:  23.967162790697675
R-squared:  0.8974486128534089
```

Random Forest is used to train the dataset. Leo Breiman and Adele Cutler created the popular machine learning algorithm Random Forest, which aggregates the output of several decision trees to produce a single outcome. Its versatility and ease of use, combined with its ability to handle both regression and classification problems, have driven its popularity (R, 2024). Since the problem is related to regression, then the Random Forest Regressor is used. The figure above shows the model performance without applying Randomized Search.

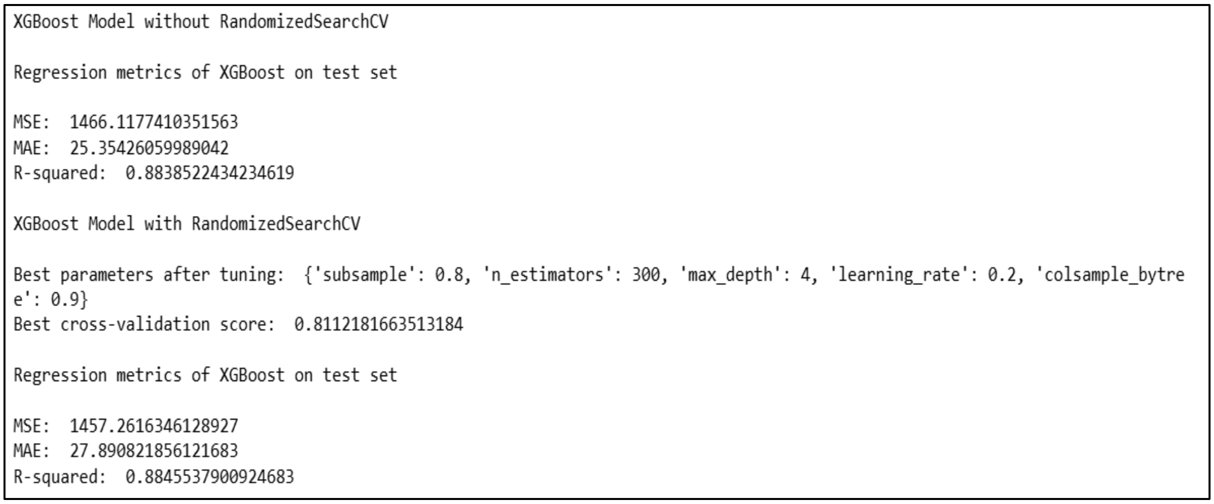
Hyperparameter tuning was performed by using Randomized Search. It does not equally consider all given parameter values. Instead, it samples a random combination of hyperparameters with each iteration, and this sampling can be specified in advance. The figure below illustrates the result of the model after applying Randomized Search. However, by Randomized Search, the model performance had no improvement since R-squared decreased from 0.897 to 0.873. Therefore, the Random Forest Model without Randomized Search is considered.



```
Model with RandomizedSearchCV
Fitting 3 folds for each of 50 candidates, totalling 150 fits

Best parameter after tuning:  {'n_estimators': 100, 'min_samples_split': 15, 'min_samples_leaf': 6, 'max_features': 0.5, 'max_d
epth': 5, 'bootstrap': False}
Best cross validation score:  0.7955273505192055
Fitting 3 folds for each of 50 candidates, totalling 150 fits

Regression  of test set

MSE:  1606.007677426755
MAE:  28.717890315111248
R-squared:  0.8727699478106693
```

3. 4 XGBoost

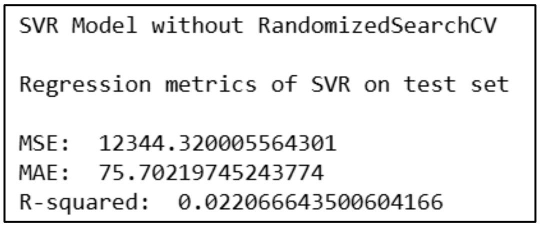
A powerful machine-learning technique called XGBoost can assist us in better understanding our data and decision-making. An application of gradient-boosting decision trees is called XGBoost. Large dataset performance, speed, and ease of use are the main goals of XGBoost's design. It can be utilised right away after installation without the need for any additional configuration because it doesn't require parameter tuning or optimisation (Simplilearn, 2023).



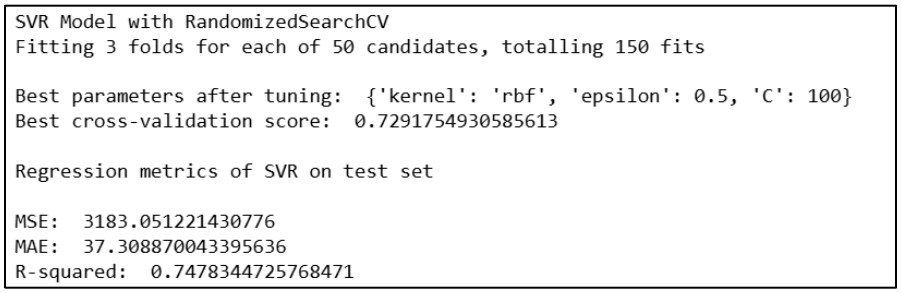
```
XGBoost Model without RandomizedSearchCV

Regression metrics of XGBoost on test set

MSE:  1466.1177410351563
MAE:  25.35426059989042
R-squared:  0.8838522434234619

XGBoost Model with RandomizedSearchCV

Best parameters after tuning:  {'subsample': 0.8, 'n_estimators': 300, 'max_depth': 4, 'learning_rate': 0.2, 'colsample_bytre
e': 0.9}
Best cross-validation score:  0.8112181663513184

Regression metrics of XGBoost on test set

MSE:  1457.2616346128927
MAE:  27.890821856121683
R-squared:  0.8845537900924683
```

Without RandomizedSearchCV, the XGBoost model achieved a Mean Squared Error (MSE) of 1466.12, a Mean Absolute Error (MAE) of 25.35, and an R-squared value of 0.884 on the test set. Furthermore, the hyperparameter tuning with RandomizedSearchCV has resulted in a marginal improvement in the model’s R-squared value, indicating a slightly better fit. This suggests that while the model’s overall predictive power improved, the average error per prediction increased.

3.5 Support Vector Regression (SVR)



```
SVR Model without RandomizedSearchCV

Regression metrics of SVR on test set

MSE:  12344.320005564301
MAE:  75.70219745243774
R-squared:  0.022066643500604166
```



```
SVR Model with RandomizedSearchCV
Fitting 3 folds for each of 50 candidates, totalling 150 fits

Best parameters after tuning:  {'kernel': 'rbf', 'epsilon': 0.5, 'C': 100}
Best cross-validation score:  0.7291754930585613

Regression metrics of SVR on test set

MSE:  3183.051221430776
MAE:  37.308870043395636
R-squared:  0.7478344725768471
```

Support Vector Regression (SVR) is an extension of Support Vector Machines (SVM) that is specifically designed for solving regression problems. The results for the basic SVR model are shown in the figure above. The default values do not optimize the model’s ability to fit the data and has led to poor performance. Therefore, hyperparameter tuning is performed to improve the performance. In this case, we are using grid search method. After tuning, the model’s performance improved significantly because the hyperparameters were optimized to better capture the underlying patterns in the data.

4.0 Conclusion

By analyzing the features from various Covid-19 datasets, the new Covid-19-related deaths are predicted. Several supervised learning models, incorporating Multiple Regression, Keras Sequential Neural Network, Random Forest, XGBoost, and Support Vector Regression (SVR), are used. Generally, evaluation metrics are numerical measurements employed to assess the performance and efficiency of a machine learning model (Srivastava, 2019). These metrics help in comparing various models or algorithms and offer insights into the model's effectiveness. The metrics included in the comparison and evaluation process are R-squared, MSE and MAE. Below shows the evaluation metrics with Randomized Search, except Random Forest.

Models	**R-squared**	**MSE**	**MAE**
Multiple Regression	0.777	2816.439	39.137

Keras Sequential Neural Network	0.766	2952.583	35.423
Random Forest	0.897	1294.492	23.967
XGBoost	0.885	1457.262	27.891
SVR	0.748	3183.051	37.309

Note that Random Forest was evaluated without undergoing Randomized Search Cross Validation (CV), as it shows a decline in R-squared after the hyperparameter tuning. It is probably that the hyperparameter search space was not optimal given that tuning dropped the model's intrinsic strengths. This underlines the necessity of carefully selecting hyperparameters and confirming their effects on model performance. Random Forest is well-known for its durability and ability to handle huge, high-dimensional datasets while efficiently controlling overfitting via ensemble learning. One of the key strengths is its ability to capture complicated patterns and relationships between features without requiring substantial adjustment, as seen by its good initial performance metrics.

From the table above, Random Forest has the highest R-squared (0.897), and the lowest MSE (1294.492) and MAE (23.967). The high R-squared value denotes that there are approximately 90% of variance in the target variable, suggesting that the model captures the underlying patterns in the data effectively. This indicates that Random Forest has the best model performance among the other models as it is the most accurate in predicting new Covid-19 deaths.

In contrast, the SVR shows the lowest R-squared (0.748), and high MSE and MAE. It reveals that SVR has the worst model performance. Overall, the training speed is efficient. The model performance is ranked from worst to best:

SVR < Keras Sequential Neural Network < Multiple Regression < XGBoost < Random Forest

In conclusion, after evaluating R-squared, MSE and MAE, we determined that the Random Forest model is the optimal choice for deployment as it demonstrates the highest accuracy on the new Covid-19-related deaths. It is suggested as the model to make predictions on new datasets.

References

Hayes, A. (2023, April 29). *How Multiple Linear Regression Works*. Investopedia. https://www.investopedia.com/terms/m/mlr.asp

Lazy Programmer. (2023, July 6). *Understanding the differences between Tensorflow Keras Sequential class and Model class*. Lazy Programmer. https://lazyprogrammer.me/understanding-the-differences-between-tensorflow-keras-sequential-class-and-model-class/

Simplilearn. (2022, November 22). *What is XGBoost? An Introduction to XGBoost Algorithm in Machine Learning | Simplilearn*. Simplilearn.com. https://www.simplilearn.com/what-is-xgboost-algorithm-in-machine-learning-article

Srivastava, T. (2019, August 6). *12 Important Model Evaluation Metrics for Machine Learning Everyone Should Know (Updated 2024)*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error

R, S. E. (2021, June 17). *Understanding Random Forest Algorithm With Examples*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/06/understanding-random-