CPSC 340 Machine Learning Take-Home Midterm Exam
Kaggle

Template for Question 2

1 Team

Team Members	Harper Cheng (z2f3b), Xinyao Fan (x8x1b), Ning Shen (70533633; i0c1p)
Kaggle Team Name	*123*

2 Solution Summary

To predict daily death counts in Canada with linear regression, we first decide on which part of data should be used for prediction. The growth curve of the total death toll in Canada shows a sigmoid-shaped trend over the given period. It is likely that the most recent data better captures the trend in the near future and has the most desirable predicting power. Thus, we truncate the raw data and use only the last 100 data points for training the regression model. We adopt the autoregressive (AR) model which allows us to predict future behaviours based on trend in the past. In order to have more reliable prediction on total death counts, we decide to fit two AR models with one using daily death and the other using total death as responses, respectively. The prediction for total death counts will be based on the average prediction given by these two models.

Features are selected based on explanatory analysis including visual examination on the scatterplots, correlation matrices and heatmaps as well as stepwise selection for choosing the most relevant features. It is suggested that lag 1-3 of daily and total death counts in Canada are the most relevant features in predicting the total death. In addition, we used Euclidean distance to pinpoint six countries that are most similar to Canada in terms of daily death tolls in the hope of borrowing information from these countries for better prediction. However, correlation is weak between the daily death of some selected countries and that of Canada. Based on the preliminary analysis, the features we decide to include in the first AR model (whose response variable is daily death) are lag 1-3 of daily deaths in Canada. For the second AR model (whose response variable is total death), lag 1-3 of total deaths in Canada are included.

The AR model is implemented where the least squares objective is minimized to obtain estimates on the weights. The feature matrices are coded to include Canadian daily or total deaths and its corresponding three lags. Below is the mathematical formulation of the two models our approach:

$$X_t = \alpha + \sum_{i=1}^{3} \psi_i L^i X_i,$$

$$(1-L)X_t = \beta + \sum_{i=1}^{3} \phi_i L^i (1-L) X_t,$$

where X_t is the cumulative death toll on date t, ψ_i and ϕ_i are the linear regression weights, α and β are the intercepts, and L is the lag operator. Predictions obtained from two separate models are averaged to yield a final prediction.

3 Experiments

Prior to feature selection, a time series plot depicting the trend in daily deaths in Canada is inspected. We noticed a surge in death counts on October 4th. It is suspected that by using the death counts in Canada as the only training set might not be reliable due to the presence of this recent outlier. We therefore consider to borrow information from other countries that have similar trends with Canada in terms of number of daily deaths. The top three most similar countries are selected based on their proximity to Canada calculated with Euclidean distance and the time series plot depicting the trend in their death counts is shown in Figure 1. It can be seen that all four countries have similar trend in death tolls and it might be appropriate to incorporate their information for training the model.

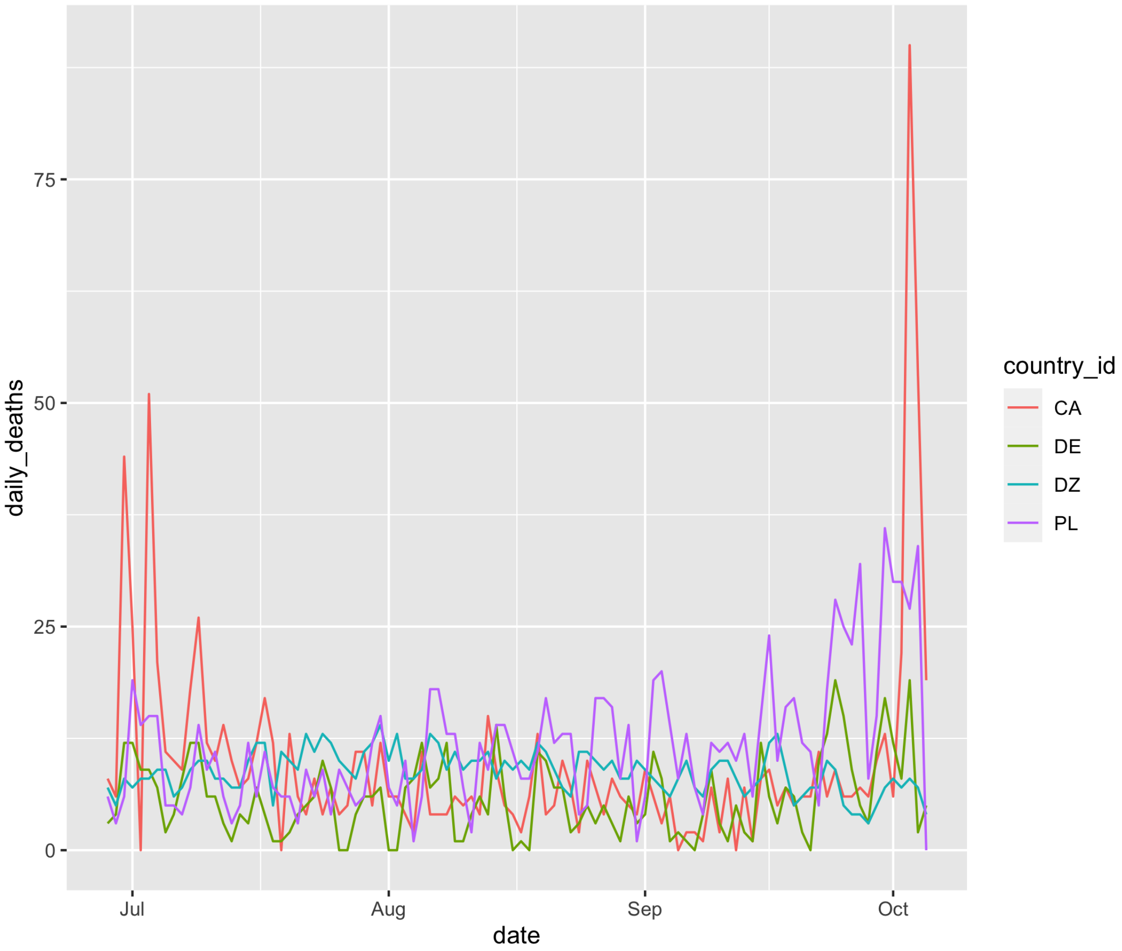

Figure 1: Time series plot of changes in daily deaths in Canada and three other countries that are most similar to Canada in terms of trend in death tolls.

In terms of feature selection, a heatmap is first plotted to help us understand the correlation between each features. The main idea is to select features that are correlated to the response (daily death counts) while keeping the multicollinearity to the minimum, that is, features that are highly correlated should not appear in the model at the same time. Since the death counts in Canada is most predictive of its own trend, we decide to consider lag 1, 2, and 3 of Canadian daily death counts as possible features. As for other countries, even though they show similar trends, countries are fundamentally different in so many aspects. In order to minimize the influence of other countries on the prediction and not to introduce too much noise, we only consider testing lag 1 of daily death counts for these three countries. Correlation between each feature is shown in Figure 2. It seems that lag 1 of daily death in country “DE”, “PL”, and “BE” have relatively high correlation to Canadian daily death counts whereas the correlations between Canada and the other three countries are not obvious. We notice on the upper-right corner, four features show very high correlation. We decide to choose `cases_14_100k_lag1_CA` as a potential feature. Thus, based on the heatmap, we have selected the following features for stepwise selection: `daily_deaths_lag1_CA`, `daily_deaths_lag2_CA`, `daily_deaths_lag3_CA`, `daily_deaths_lag1_DE`, `daily_deaths_lag1_PL`, `daily_deaths_lag1_BE`, `cases_14_100k_lag1_CA`. As for the second model, we use total death counts instead of daily death counts as responses.

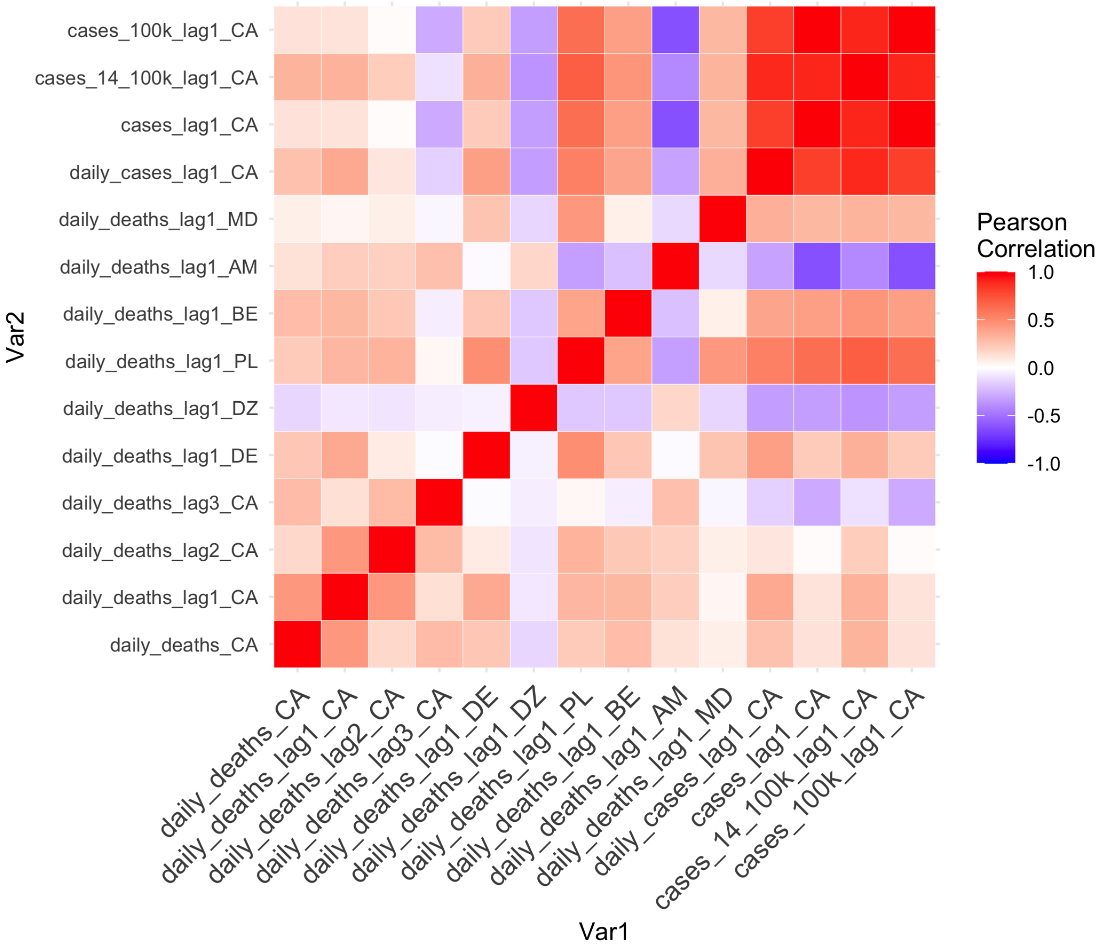

Figure 2: Visualization of correlation matrix

The stepwise selection is used to select the most relevant features for predicting Canadian death counts. The best model is the one with the lowest AIC value meaning that the ideal model would have a relatively satisfactory fit to the given data with the minimum possible number of features. We would like to obtain a decent model without including too many features because such a model might overfit to the training set and perform poorly on the test set. More specifically, we used the forward selection technique where a single feature is added to the model for each run and the corresponding AIC values are compared. The model with the lowest AIC value includes the following features: `daily_deaths_lag1_CA`, `daily_deaths_lag2_CA`, `daily_deaths_lag3_CA`. It turns out the daily deaths from other countries do not help much when it comes to predicting the death counts in Canada. After careful consideration, we think it might be prudent to just use Canadian data for training the model. We have come to the conclusion based on results from preliminary analyses as well as a common rationale that the best predictor of future behaviour is their own past behaviour. Even though a noticeably large fluctuation might undermine its future predictions, we have reasons to believe that the death toll for Canada better reflects its future trend than a model with information from other countries incorporated where noise might be introduced.

We fit two regression models, namely, one uses daily deaths in Canada and its lag 1-3 as features, the other uses total deaths in Canada and its lag 1-3 as features. The least squares objective is minimized to obtain estimates on weights. Predictions from both models are then averaged and used as the final prediction.

4 Results

Team Name	Kaggle Phase 1 Score	Kaggle Phase 2 Score
123	90.48933	*your Phase 2 Kaggle score*

5 Conclusion

We learnt that there is no gold standard for feature selection. One has to try as many methods as possible before arriving to a conclusion. Even with all the efforts, we still cannot state definitively that the selected features are the best for prediction. We gained some understanding of autoregressive model and how it can be applied to forecast the future. Had more time been given, we would have tried to implement the feature matrix given in the instruction pdf to make the model country-agnostic. By doing so, it might help us better utilize the Covid data of the entire world and not limit our model to Canada alone. Additionally, the linearity between some features and the response is not obvious which means that feature or response transformation might be helpful. However, after some trial and error, we still could not decide on a satisfactory transformation to be applied.

Appendix - Python code

```python
import os
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import datetime as dt
import csv
from linear_model import LeastSquaresBias


def euclidean_dist_squared(X, Xtest):
    return np.sum(X**2, axis=1)[:,None] + np.sum(Xtest**2, axis=1)[None] - 2 * np.dot(X,
    Xtest.T)

def rmse(predictions, targets):
    return np.sqrt(((predictions - targets) ** 2).mean())


filename = "phase2_training_data.csv"
with open(os.path.join("..","data",filename),"rb") as f:
    df0 = pd.read_csv(f)

df0.head()

#reorganize the dataset
df = df0.pivot_table(index="date",columns=’country_id’,values=[’deaths’,’cases’,’
    cases_14_100k’,’cases_100k’])
```

```python
dates = [dt.datetime.strptime(date, "%m/%d/%Y").date() for date in df.index.values]
df = df.iloc[np.argsort(dates),:]
df.head()

#extract death information
df_deaths = df[’deaths’]
df_deaths.head()
# daily deaths
df_diff0 = df_deaths.diff(axis=0)
print("the shape of df_diff0",df_diff0.shape)

df_diff=df_diff0.iloc[200:300,:]
print(df_diff.shape)
euclid_dis = euclidean_dist_squared(np.array(df_diff[’CA’])[None], np.array(df_diff).T)
# sorted countries close to Canada in terms of daily deaths
df_diff.columns.values[np.argsort(euclid_dis.flatten())[range(10)]]


#compute the lag of daily death of canada
daily_death_ca=df_diff0[’CA’]
daily_death_ca_lag1=daily_death_ca.shift(periods=1)
daily_death_ca_lag2=daily_death_ca.shift(periods=2)
daily_death_ca_lag3=daily_death_ca.shift(periods=3)

feature_space=pd.concat([daily_death_ca,daily_death_ca_lag1,daily_death_ca_lag2,daily_death_ca_lag3],axis=1)
feature_space.columns=["daily_death_ca","daily_death_ca_lag1","daily_death_ca_lag2","daily_death_ca_lag3"]
fs_sub=feature_space.iloc[200:300,:]

print(fs_sub.head())

model=LeastSquaresBias()
X=feature_space.iloc[200:300,1:4]
y=feature_space.iloc[200:300,0]
model.fit(X=X,y=y)
print(model.w)


dat_pred = feature_space
for i in range(5):
    new_data = np.array([dat_pred.iloc[-1,0], dat_pred.iloc[-2,0], dat_pred.iloc[-3,0]])[None]
    y_pred = model.predict(X=new_data)
    dat_pred = pd.concat([dat_pred, pd.DataFrame(np.append(y_pred, new_data[0])[None], columns=dat_pred.columns.values)], axis=0)

pred_deaths_CA = np.cumsum(dat_pred.iloc[1:,0])

#compute the lag of daily death of canada
death_ca=df_deaths[’CA’]
print(death_ca)
death_ca_lag1=death_ca.shift(periods=1)
death_ca_lag2=death_ca.shift(periods=2)
death_ca_lag3=death_ca.shift(periods=3)

feature_space=pd.concat([death_ca,death_ca_lag1,death_ca_lag2,death_ca_lag3],axis=1)
feature_space.columns=["death_ca","death_ca_lag1","death_ca_lag2","death_ca_lag3"]

model=LeastSquaresBias()
X=feature_space.iloc[200:300,1:4]
y=feature_space.iloc[200:300,0]
model.fit(X=X,y=y)

dat_pred = feature_space
```

```python
for i in range(5):
    new_data = np.array([dat_pred.iloc[-1,0], dat_pred.iloc[-2,0], dat_pred.iloc[-3,0]])[None]
    y_pred = model.predict(X=new_data)
    dat_pred = pd.concat([dat_pred, pd.DataFrame(np.append(y_pred, new_data[0])[None], columns=dat_pred.columns.values)], axis=0)

pred_deaths_CA2 = dat_pred.iloc[:,0]


#write the prediction results
prediction = (pred_deaths_CA2[-5:]+pred_deaths_CA[-5:])/2
print(prediction)
prediction.to_csv("../data/prediction.csv", index = False, sep = ",")
```