SENG 474
Final Project

Topic: Predict unemployment rate of Canada

Ronald Liu V00838627

Brandon Ming V00844078

Table of Contents

1. Introduction:

Unemployment rate is a significant indicator of the labour market transition, which is widely being used by many companies for future decision making. Prediction of unemployment rate can be useful for business strategy in catching the key moment or avoiding potential loss. Although we normally said that unemployment basically related to the economy situation, there are minor perturbations hugely affecting the entire labour market, such as the immigrant status and job vacancy.

In this project, we are going to investigate the trend of unemployment rate with the use of machine learning technique and to build a data mining model for future prediction. The first part of the project is the data collection, which is about finding a clean training dataset. The second part is to work on the data preprocessing. Then the final part is about data mining with regression related models.

2. Data Collection:

The dataset is created by merging several aggregated tables from Statistics Canada and Bank of Canada. First, we collected the unemployment rate of Canada from 2010 January to 2019 January through the Labour Force Survey. Then we merged the explanatory variable (unemployment rate) with different sets of response variables, such as Age group, Sex and Gross Domestic Product (GDP), etc. One of the pros of choosing unemployment rate is because data can be obtained from the monthly LFS report, which means we may have enough tuples for training our model.

The dataset includes 25 attributes, contains monthly data from January 2010 up to January 2019. All the attributes are quantitative. The detail of each attributes is listed as follow:

- Unemp_rate
    - The unemployment rate is the number of unemployed persons expressed as a percentage of the labour force.
    - Formula: Unemployment population / Labour Force Total x 100%
- Total_pop
    - The total population in Canada, whose working age, 15 years and over.
    - Estimates in thousands, rounded to the nearest hundred.

- Labour_force
  - The Labour Force population in Canada.
  - Number of civilian, non-institutionalized persons 15 years of age and over who, during the reference week, were employed or unemployed.
  - Estimates in thousands, rounded to the nearest hundred.
- Not_labour_force
  - The population of Canada who is not in the Labour Force, 15 years of age and over.
  - Estimates in thousands, rounded to the nearest hundred.
- Males_tpop
  - The number of males in the total population of Canada.
  - Estimates in thousands, rounded to the nearest hundred.
- Females_tpop
  - The number of females in the total population of Canada.
  - Estimates in thousands, rounded to the nearest hundred.
- Males_lfpop
  - The number of males in the Labour Force population of Canada.
  - Estimates in thousands, rounded to the nearest hundred.
- Females_lfpop
  - The number of females in the Labour Force population of Canada.
  - Estimates in thousands, rounded to the nearest hundred.
- 15-24yrs
  - Number of people in the Labour Force who are aged between 15 to 24.
- 25-44yrs
  - Number of people in the Labour Force who are aged between 25 to 44.
- 45-64yrs
  - Number of people in the Labour Force who are between 45 to 64.
- 65over
  - Number of people in the Labour Force who are aged 65 and over.
- weeklyearn
  - Average weekly earnings including overtime for all employees (Industrial aggregate excluding unclassified businesses) in dollars.
- Job_vacancies
  - Number of job vacancies, which is the stock of vacant positions on the last business day of the month.
  - Estimates in thousands.
- Labour_demand
  - Labour demand is the sum of total payroll employment (met labour demand)

    and the number of job vacancies (unmet labour demand).
  - Estimates in thousands.
- Job_leavers
  - For those who worked in the previous 12 months, but currently unemployed.
  - Reasons of leaving the job: Own illness or disability, Personal or family reasons, Going to school, Dissatisfied, Retired and other reasons.
- Job_losers
  - For those who worked in the previous 12 months, but currently unemployed.
  - Reasons of losing the job: Permanent layoff or Temporary layoff
- GDP
  - Gross domestic product (GDP)
- CPI
  - Consumer Price Index (all items)
- GSPTSE
  - S&P/TSX Composite index in CAD
- Education level (4 different levels)

  Counting the number of people that are in Labour Force, 15 years of age over and with the particular education attainment.

I.	no _degree	-	No degree, certificate or diploma
II.	high_school	-	High school graduate
III.	diploma	-	Postsecondary certificate or diploma
IV.	uni	-	Attained at least a University bachelor's degree

- Immigrants_5yrs
  - Number of Immigrants in Labour Force, landed 5 or less years earlier
  - Landed immigrants (Refers to people who are, or have been, landed immigrants in Canada. A landed immigrant is a person who has been granted the right to live in Canada permanently by immigration authorities.

3. Data Preprocessing:

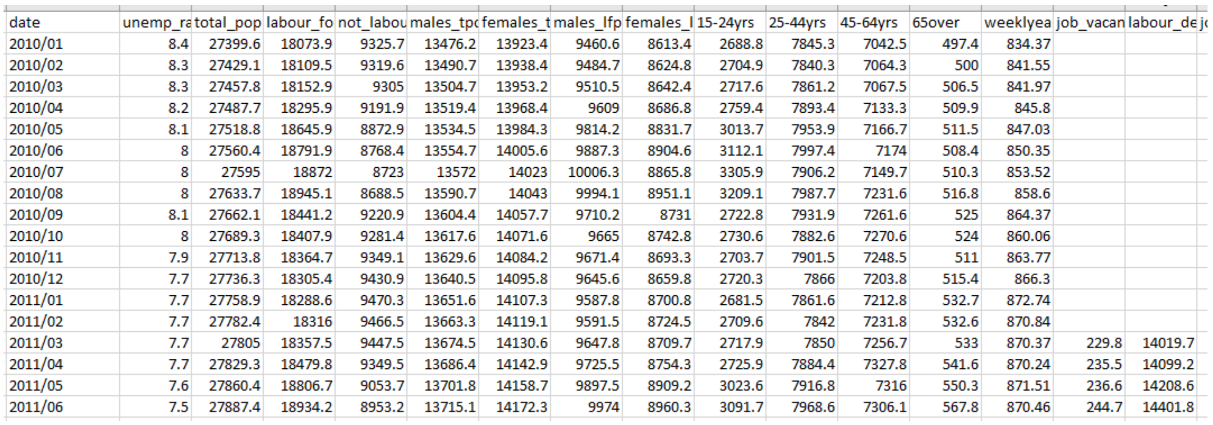
For the data preprocessing, we basically separated into 3 parts:

- Data screening/cleaning
  - Missing values.
- Feature selection
- Underfitting and overfitting

3.1 Data screening/cleaning:

In this process, we looked at the dataset that we used, and checked whether there is any missing value (empty cell). Unfortunately, we found that there are about 14 missing values in the features “job_vacancies” and “labour demand” (see figure1 below). After different testing, we decided to drop all the rows which have any empty cell. Then we have 95 rows left after dropping, which is enough for training our model.

Figure1: missing data



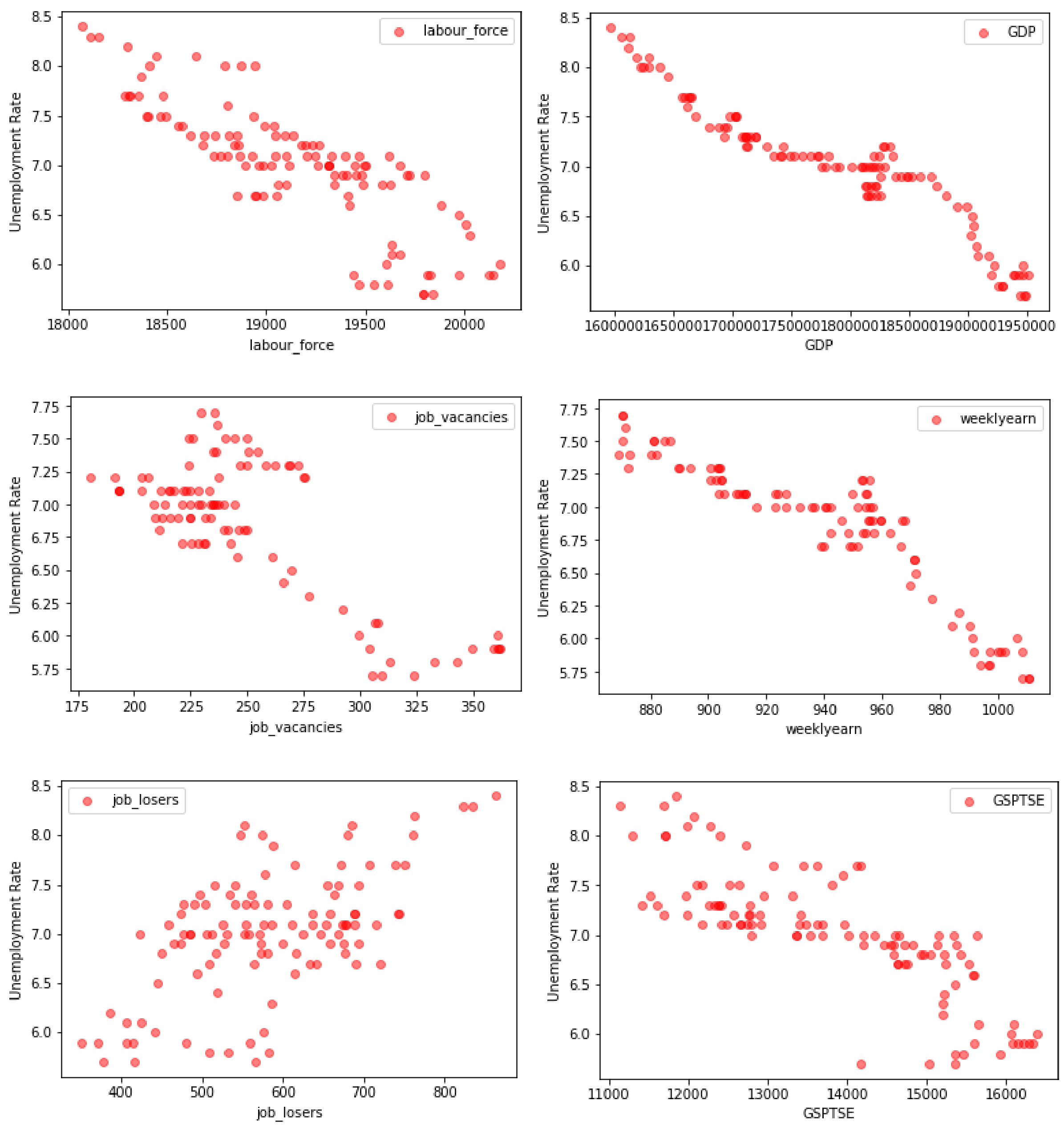
date	unemp_ra	total_pop	labour_fo	not_labou	males_tpo	females_t	males_lfp	females_l	15-24yrs	25-44yrs	45-64yrs	65over	weeklyea	job_vacan	labour_de	j
2010/01	8.4	27399.6	18073.9	9325.7	13476.2	13923.4	9460.6	8613.4	2688.8	7845.3	7042.5	497.4	834.37			
2010/02	8.3	27429.1	18109.5	9319.6	13490.7	13938.4	9484.7	8624.8	2704.9	7840.3	7064.3	500	841.55			
2010/03	8.3	27457.8	18152.9	9305	13504.7	13953.2	9510.5	8642.4	2717.6	7861.2	7067.5	506.5	841.97			
2010/04	8.2	27487.7	18295.9	9191.9	13519.4	13968.4	9609	8686.8	2759.4	7893.4	7133.3	509.9	845.8			
2010/05	8.1	27518.8	18645.9	8872.9	13534.5	13984.3	9814.2	8831.7	3013.7	7953.9	7166.7	511.5	847.03			
2010/06	8	27560.4	18791.9	8768.4	13554.7	14005.6	9887.3	8904.6	3112.1	7997.4	7174	508.4	850.35			
2010/07	8	27595	18872	8723	13572	14023	10006.3	8865.8	3305.9	7906.2	7149.7	510.3	853.52			
2010/08	8	27633.7	18945.1	8688.5	13590.7	14043	9994.1	8951.1	3209.1	7987.7	7231.6	516.8	858.6			
2010/09	8.1	27662.1	18441.2	9220.9	13604.4	14057.7	9710.2	8731	2722.8	7931.9	7261.6	525	864.37			
2010/10	8	27689.3	18407.9	9281.4	13617.6	14071.6	9665	8742.8	2730.6	7882.6	7270.6	524	860.06			
2010/11	7.9	27713.8	18364.7	9349.1	13629.6	14084.2	9671.4	8693.3	2703.7	7901.5	7248.5	511	863.77			
2010/12	7.7	27736.3	18305.4	9430.9	13640.5	14095.8	9645.6	8659.8	2720.3	7866	7203.8	515.4	866.3			
2011/01	7.7	27758.9	18288.6	9470.3	13651.6	14107.3	9587.8	8700.8	2681.5	7861.6	7212.8	532.7	872.74			
2011/02	7.7	27782.4	18316	9466.5	13663.3	14119.1	9591.5	8724.5	2709.6	7842	7231.8	532.6	870.84			
2011/03	7.7	27805	18357.5	9447.5	13674.5	14130.6	9647.8	8709.7	2717.9	7850	7256.7	533	870.37	229.8	14019.7	
2011/04	7.7	27829.3	18479.8	9349.5	13686.4	14142.9	9725.5	8754.3	2725.9	7884.4	7327.8	541.6	870.24	235.5	14099.2	
2011/05	7.6	27860.4	18806.7	9053.7	13701.8	14158.7	9897.5	8909.2	3023.6	7916.8	7316	550.3	871.51	236.6	14208.6	
2011/06	7.5	27887.4	18934.2	8953.2	13715.1	14172.3	9974	8960.3	3091.7	7968.6	7306.1	567.8	870.46	244.7	14401.8	

3.2 Feature selection:

We first looked at different scatter plots of the data, in order to see any numerical analysis method would be fit to out data.

We have looked at variables such as: “labour_force”, “GDP”, “job_vacancies”, “weeklyearn”, “job_losers” and “GSPTSE”. From the figure2 below, we can clearly see that “GDP” seems to have a linear relationship versus unemployment rate. Therefore, it is worthwhile to investigate more details about how these numerical features related to our explanatory variable (unemployment rate).

Figure2: Scatter plots of different variables



The next step we did is using Pearson Correlation to look at the correlation between each variable, which clearly show us all correlations in a square matrix format (-1 < r < 1).

Here is the equation of the Pearson Correlation Coefficient: (from Wikipedia)

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} \quad \text{(Eq.1)}$$

where:

- cov is the covariance
- σ_X is the standard deviation of X
- σ_Y is the standard deviation of Y

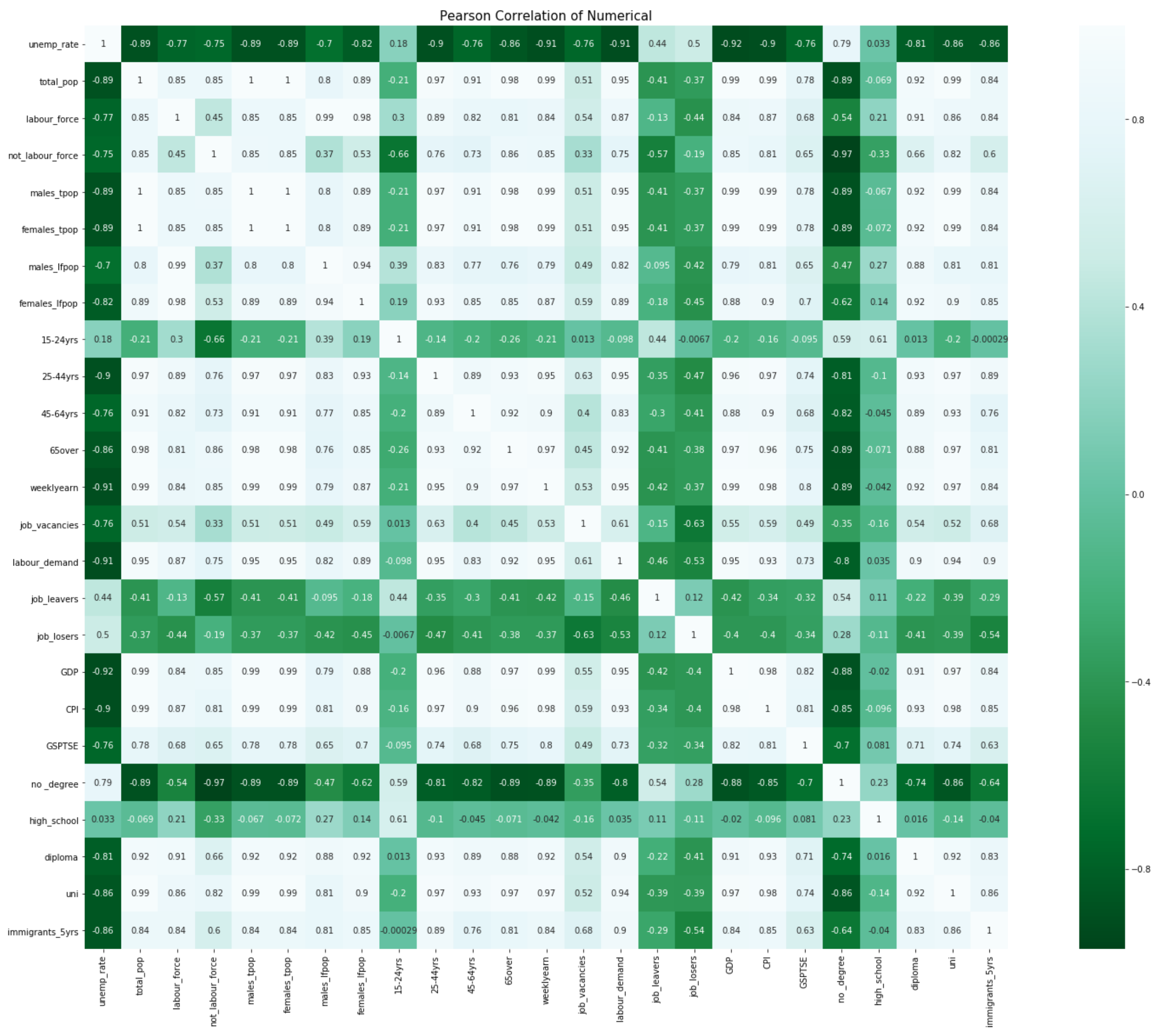
the formula for ρ can also be written as

$$\rho_{X,Y} = \frac{\mathrm{E}[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} \quad \text{(Eq.2)}$$

where:

- σ_Y and σ_X are defined as above
- μ_X is the mean of X
- μ_Y is the mean of Y
- E is the expectation.

Figure 3: Pearson Correlation



	unemp_rate	total_pop	labour_force	not_labour_force	males_tpop	females_tpop	males_lfpop	females_lfpop	15-24yrs	25-44yrs	45-64yrs	65over	weeklyearn	job_vacancies	labour_demand	job_leavers	job_losers	GDP	CPI	GSPTSE	no_degree	high_school	diploma	uni	immigrants_5yrs
unemp_rate	1	-0.89	-0.77	-0.75	-0.89	-0.89	-0.7	-0.82	0.18	-0.9	-0.76	-0.86	-0.91	-0.76	-0.91	0.44	0.5	-0.92	-0.9	-0.76	0.79	0.033	-0.81	-0.86	-0.86
total_pop	-0.89	1	0.85	0.85	1	1	0.8	0.89	-0.21	0.97	0.91	0.98	0.99	0.51	0.95	-0.41	-0.37	0.99	0.99	0.78	-0.89	-0.069	0.92	0.99	0.84
labour_force	-0.77	0.85	1	0.45	0.85	0.85	0.99	0.98	0.3	0.89	0.82	0.81	0.84	0.54	0.87	-0.13	-0.44	0.84	0.87	0.68	-0.54	0.21	0.91	0.86	0.84
not_labour_force	-0.75	0.85	0.45	1	0.85	0.85	0.37	0.53	-0.66	0.76	0.73	0.86	0.85	0.33	0.75	-0.57	-0.19	0.85	0.81	0.65	-0.97	-0.33	0.66	0.82	0.6
males_tpop	-0.89	1	0.85	0.85	1	1	0.8	0.89	-0.21	0.97	0.91	0.98	0.99	0.51	0.95	-0.41	-0.37	0.99	0.99	0.78	-0.89	-0.067	0.92	0.99	0.84
females_tpop	-0.89	1	0.85	0.85	1	1	0.8	0.89	-0.21	0.97	0.91	0.98	0.99	0.51	0.95	-0.41	-0.37	0.99	0.99	0.78	-0.89	-0.072	0.92	0.99	0.84
males_lfpop	-0.7	0.8	0.99	0.37	0.8	0.8	1	0.94	0.39	0.83	0.77	0.76	0.79	0.49	0.82	-0.095	-0.42	0.79	0.81	0.65	-0.47	0.27	0.88	0.81	0.81
females_lfpop	-0.82	0.89	0.98	0.53	0.89	0.89	0.94	1	0.19	0.93	0.85	0.85	0.87	0.59	0.89	-0.18	-0.45	0.88	0.9	0.7	-0.62	0.14	0.92	0.9	0.85
15-24yrs	0.18	-0.21	0.3	-0.66	-0.21	-0.21	0.39	0.19	1	-0.14	-0.2	-0.26	-0.21	0.013	-0.098	0.44	-0.0067	-0.2	-0.16	-0.095	0.59	0.61	0.013	-0.2	-0.00029
25-44yrs	-0.9	0.97	0.89	0.76	0.97	0.97	0.83	0.93	-0.14	1	0.89	0.93	0.95	0.63	0.95	-0.35	-0.47	0.96	0.97	0.74	-0.81	-0.1	0.93	0.97	0.89
45-64yrs	-0.76	0.91	0.82	0.73	0.91	0.91	0.77	0.85	-0.2	0.89	1	0.92	0.9	0.4	0.83	-0.3	-0.41	0.88	0.9	0.68	-0.82	-0.045	0.89	0.93	0.76
65over	-0.86	0.98	0.81	0.86	0.98	0.98	0.76	0.85	-0.26	0.93	0.92	1	0.97	0.45	0.92	-0.41	-0.38	0.97	0.96	0.75	-0.89	-0.071	0.88	0.97	0.81
weeklyearn	-0.91	0.99	0.84	0.85	0.99	0.99	0.79	0.87	-0.21	0.95	0.9	0.97	1	0.53	0.95	-0.42	-0.37	0.99	0.98	0.8	-0.89	-0.042	0.92	0.97	0.84
job_vacancies	-0.76	0.51	0.54	0.33	0.51	0.51	0.49	0.59	0.013	0.63	0.4	0.45	0.53	1	0.61	-0.15	-0.63	0.55	0.59	0.49	-0.35	-0.16	0.54	0.52	0.68
labour_demand	-0.91	0.95	0.87	0.75	0.95	0.95	0.82	0.89	-0.098	0.95	0.83	0.92	0.95	0.61	1	-0.46	-0.53	0.95	0.93	0.73	-0.8	0.035	0.9	0.94	0.9
job_leavers	0.44	-0.41	-0.13	-0.57	-0.41	-0.41	-0.095	-0.18	0.44	-0.35	-0.3	-0.41	-0.42	-0.15	-0.46	1	0.12	-0.42	-0.34	-0.32	0.54	0.11	-0.22	-0.39	-0.29
job_losers	0.5	-0.37	-0.44	-0.19	-0.37	-0.37	-0.42	-0.45	-0.0067	-0.47	-0.41	-0.38	-0.37	-0.63	-0.53	0.12	1	-0.4	-0.4	-0.34	0.28	-0.11	-0.41	-0.39	-0.54
GDP	-0.92	0.99	0.84	0.85	0.99	0.99	0.79	0.88	-0.2	0.96	0.88	0.97	0.99	0.55	0.95	-0.42	-0.4	1	0.98	0.82	-0.88	-0.02	0.91	0.97	0.84
CPI	-0.9	0.99	0.87	0.81	0.99	0.99	0.81	0.9	-0.16	0.97	0.9	0.96	0.98	0.59	0.93	-0.34	-0.4	0.98	1	0.81	-0.85	-0.096	0.93	0.98	0.85
GSPTSE	-0.76	0.78	0.68	0.65	0.78	0.78	0.65	0.7	-0.095	0.74	0.68	0.75	0.8	0.49	0.73	-0.32	-0.34	0.82	0.81	1	-0.7	0.081	0.71	0.74	0.63
no_degree	0.79	-0.89	-0.54	-0.97	-0.89	-0.89	-0.47	-0.62	0.59	-0.81	-0.82	-0.89	-0.89	-0.35	-0.8	0.54	0.28	-0.88	-0.85	-0.7	1	0.23	-0.74	-0.86	-0.64
high_school	0.033	-0.069	0.21	-0.33	-0.067	-0.072	0.27	0.14	0.61	-0.1	-0.045	-0.071	-0.042	-0.16	0.035	0.11	-0.11	-0.02	-0.096	0.081	0.23	1	0.016	-0.14	-0.04
diploma	-0.81	0.92	0.91	0.66	0.92	0.92	0.88	0.92	0.013	0.93	0.89	0.88	0.92	0.54	0.9	-0.22	-0.41	0.91	0.93	0.71	-0.74	0.016	1	0.92	0.83
uni	-0.86	0.99	0.86	0.82	0.99	0.99	0.81	0.9	-0.2	0.97	0.93	0.97	0.97	0.52	0.94	-0.39	-0.39	0.97	0.98	0.74	-0.86	-0.14	0.92	1	0.86
immigrants_5yrs	-0.86	0.84	0.84	0.6	0.84	0.84	0.81	0.85	-0.00029	0.89	0.76	0.81	0.84	0.68	0.9	-0.29	-0.54	0.84	0.85	0.63	-0.64	-0.04	0.83	0.86	1

From the figure 3 above, we can see that there is a multicollinearity happened between some features. Multicollinearity (linearly dependent) is an issue in polynomial linear models in Statistics theory. Therefore, we should beware of finding and removing these variables. As to be seen in the figure, there is high correlation between “total_pop”, “males_tpop” and “females_tpop” (r = 1), so we dropped males’ and female’s population and kept total population only. Since there is no point having 3 of them all in the model.

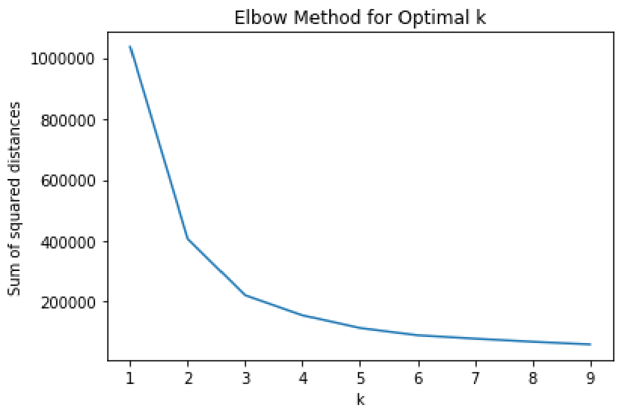
However, instead of dropping them directly, we decided to use these 2 set of data to generate a new feature called “tpop_sex_ratio”, which is using the males’ and female’s population of the same month to produce a ratio of sex differences (males : females).

Also, we see that the correlation is week between unemployment rate vs job vacancies (r=-0.76), job leavers(r=0.44) and job losers(r=0.5). Therefore, we decided to apply the K-means cluster algorithm to split instances to clusters.

K-means cluster:

1. Select K points as the initial centroids
2. Repeat from locating all points to the nearest centroid
3. Then recompute the centroid after each relocation
4. Finally getting K centroids that doesn't change after iteration

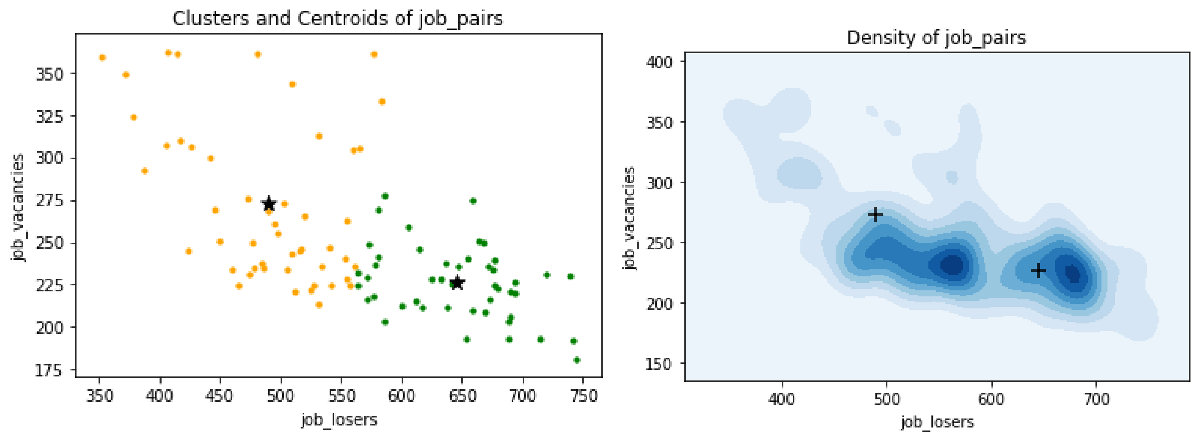
We first create a variable named "job_pairs" in paired form ("job_losers","job_vacancies"). Then we use the Elbow method to find the optimal K.



From the graph above, it shows the possible optimal point is K = 2. To verify the optimal solution is correct, we looked at the scatter plot of these pairs, and check whether they behave well in 2 separate groups. As can be seen in figure 4 below, the paired points clearly distributed into 2 groups, in green and orange colors.

In the figure, we can also see the 2 estimated centroids labeled as a Star in it. In order to further verifying the correctness of K=2, we looked at the density graph of the scatter plot. We can see that there is high density around the 2 estimated centroids. Therefore, we can move to the next step as we ensured K=2 is the optimal point. (estimated centroids: [489.14, 645.31])

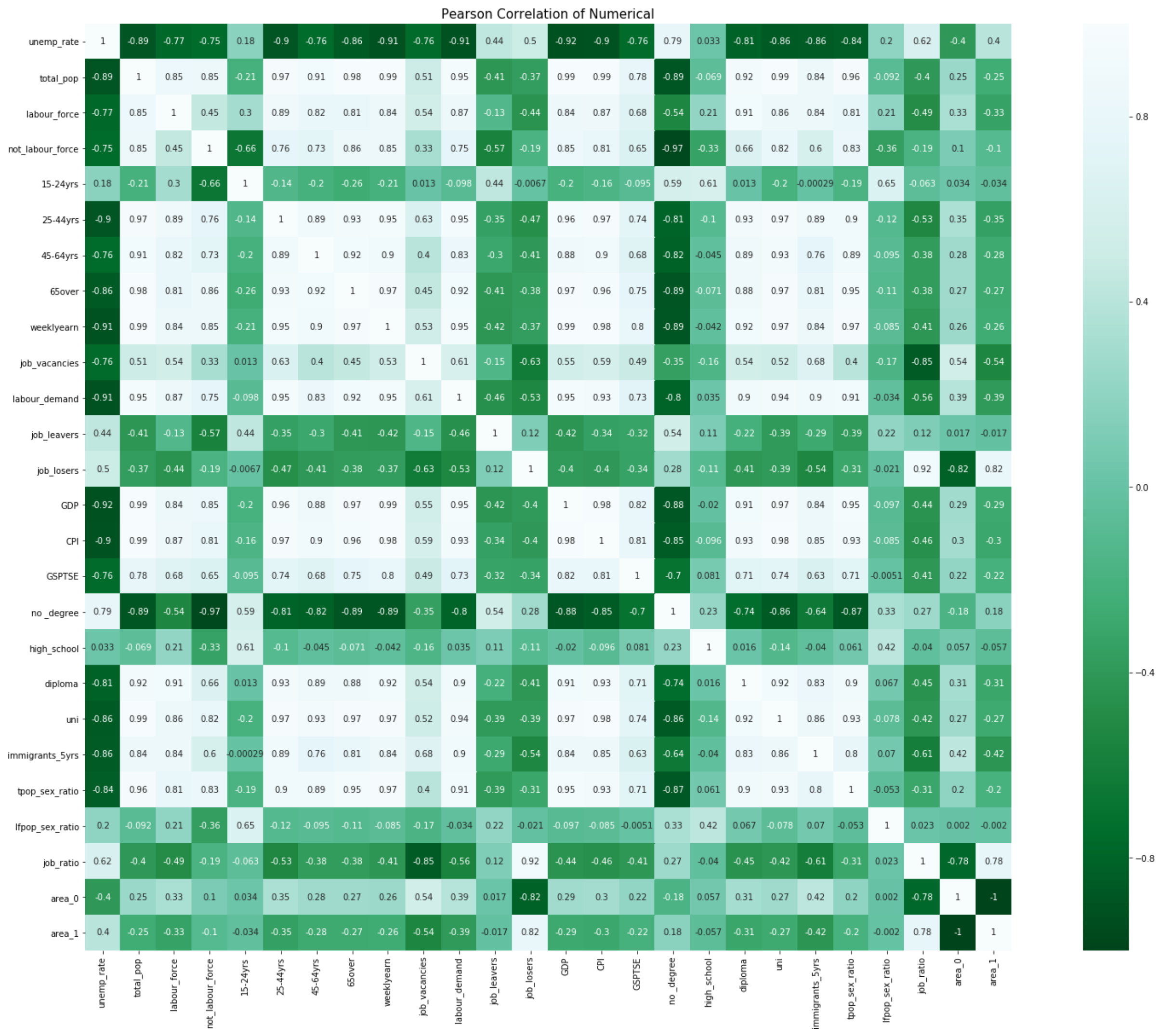
Figure 4: scatter plot of job pairs



The next step is adding the dummies variables into our data frame as “area_0” and “area_1”. We concatenate these 2 dummies variables to the original dataset. Then we implemented the Pearson Correlation once again to look at the new correlation between each variable after the feature selections.

From the figure 5 below, we can see that, although K-means cluster works well in splitting the pairs into 2 groups, but it does not show any improvement in terms of correlations. Therefore, in the next section, we will use different methods for selecting features to prevent from underfitting and overfitting.

Figure 5: Pearson Correlation after feature selection

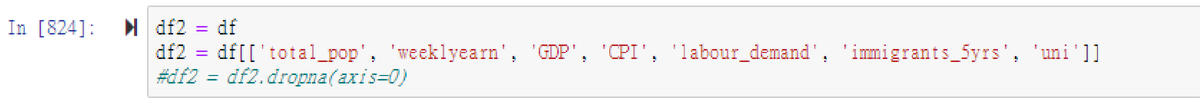


3.3 Underfitting and overfitting:

In this section, we will use 2 different methods in preventing underfitting and overfitting.

1. Manually select features by Pearson Correlation coefficients
2. Select by using SelectKBest API from sklearn.feature_selection

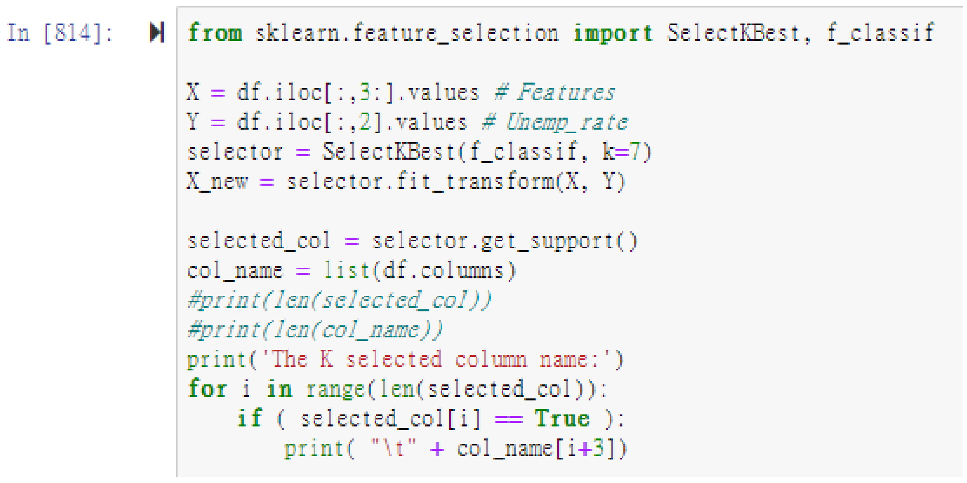
First part, we used the previous Pearson Correlation from figure 5 above. We manually picked top 7 features that has the highest correlation coefficients (closest to 1 or -1). Then we added them into a new data frame for data mining process later.



```
In [824]:  df2 = df
           df2 = df[['total_pop', 'weeklyearn', 'GDP', 'CPI', 'labour_demand', 'immigrants_5yrs', 'uni']]
           #df2 = df2.dropna(axis=0)
```

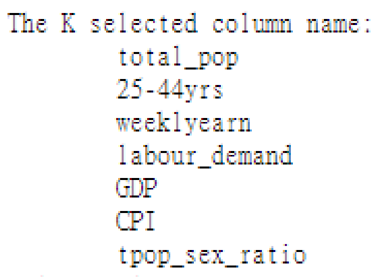
Second part, we implemented the "SelectKBest" which is imported from the library of "sklearn.feature_selection". This API is used for selecting features according to the k highest scores. This function will return the array-like of dataset with (n rows x k columns), which is the subset of X. In order to investigate what features are selected, we will use the function "get_support()" which is getting a mask, or integer index, of the features selected. It returns as a list of Boolean in the same dimension as the sequence of columns (features) we inputted.

Therefore, we used a for-loop to check through the list and print the corresponding column name. Implemented as follow:



```
In [814]:  from sklearn.feature_selection import SelectKBest, f_classif

           X = df.iloc[:,3:].values # Features
           Y = df.iloc[:,2].values # Unemp_rate
           selector = SelectKBest(f_classif, k=7)
           X_new = selector.fit_transform(X, Y)

           selected_col = selector.get_support()
           col_name = list(df.columns)
           #print(len(selected_col))
           #print(len(col_name))
           print('The K selected column name:')
           for i in range(len(selected_col)):
               if ( selected_col[i] == True ):
                   print( "\t" + col_name[i+3])
```

Output:



```
The K selected column name:
        total_pop
        25-44yrs
        weeklyearn
        labour_demand
        GDP
        CPI
        tpop_sex_ratio
```

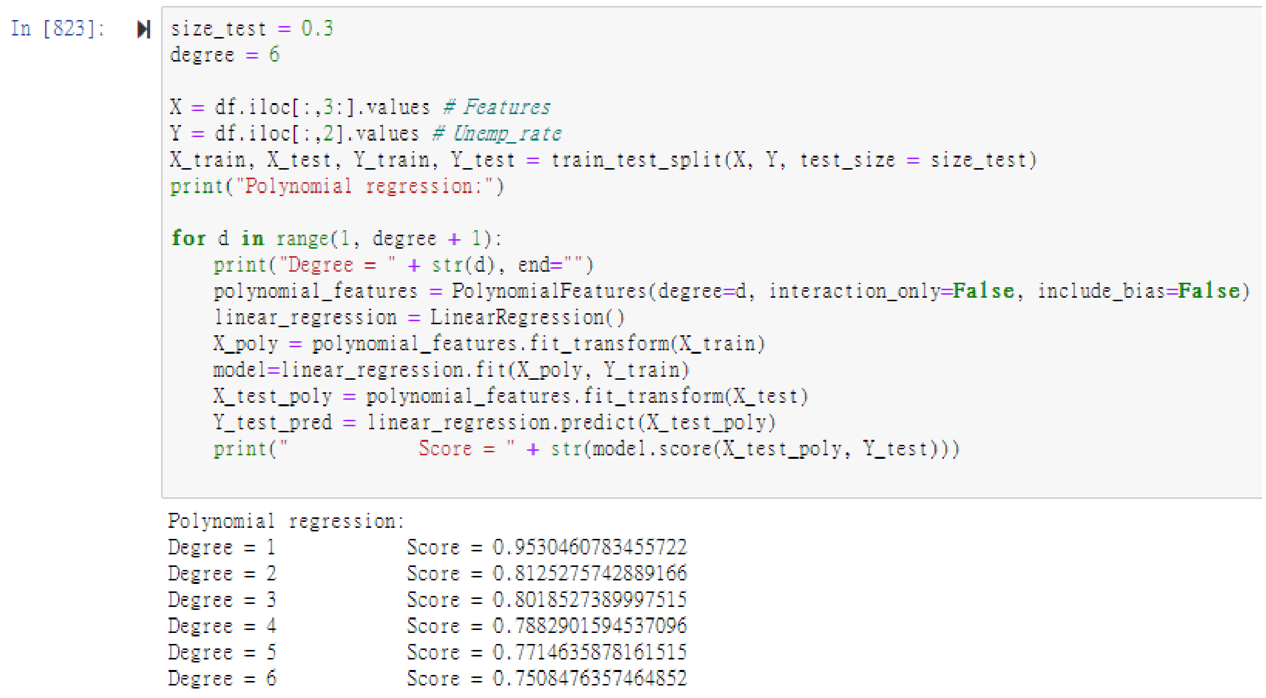
4. Data Mining:

Data mining is the process to turn raw data into useful information with specific algorithm and to predict outcomes. In this project, we used the dataset about unemployment rate, and we used the polynomial linear regression algorithm for out prediction model.

In order to set up the model, we have to split the data into 2 pieces as training set and validation set. We will use a ratio of 7:3 for our model, which is commonly suggested in the field of machine learning. As we investigated, this ratio and the randomness of splitting may affect our accuracy of the model, which will be discussed further in later section.

Part1:

The following is implemented with the dataset after the first step of features selection (without underfitting and overfitting process).

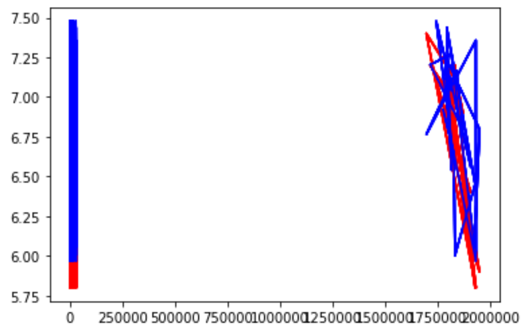


```
In [823]:  size_test = 0.3
           degree = 6

           X = df.iloc[:,3:].values # Features
           Y = df.iloc[:,2].values # Unemp_rate
           X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = size_test)
           print("Polynomial regression:")

           for d in range(1, degree + 1):
               print("Degree = " + str(d), end="")
               polynomial_features = PolynomialFeatures(degree=d, interaction_only=False, include_bias=False)
               linear_regression = LinearRegression()
               X_poly = polynomial_features.fit_transform(X_train)
               model=linear_regression.fit(X_poly, Y_train)
               X_test_poly = polynomial_features.fit_transform(X_test)
               Y_test_pred = linear_regression.predict(X_test_poly)
               print("           Score = " + str(model.score(X_test_poly, Y_test)))
```

```
Polynomial regression:
Degree = 1           Score = 0.9530460783455722
Degree = 2           Score = 0.8125275742889166
Degree = 3           Score = 0.8018527389997515
Degree = 4           Score = 0.7882901594537096
Degree = 5           Score = 0.7714635878161515
Degree = 6           Score = 0.7508476357464852
```

From the code we implemented above, we can see that we randomly picked a degree up to 6 and the output score is around 80% of accuracy.

From the scatter plot below, we observed that when we set the polynomial features highest degree to 6, linear regression produces a slightly overfitted line, but overall blue line is bounced within the red line. In the next part, we will try to implement the same algorithm but with the dataset of being manually selected or using SelectKBest as we discussed in the previous section.



Part2:

In this part, we will run the same algorithm with the selected features. As we discussed in section 3.3 part1, we implemented with degree of 7. However, the output scores show that the scores dropped rapidly in degree of 4 (score = 0.072). Therefore, we suspect the model is overfitting and the model performed better when degree is equal to 3 or lower, so we may try other method (selectKBest) to have more investigation.

```
In [824]:  df2 = df
           df2 = df[['total_pop', 'weeklyearn', 'GDP', 'CPI', 'labour_demand', 'immigrants_5yrs', 'uni']]
           #df2 = df2.dropna(axis=0)
```

```
In [831]:  size_test = 0.3
           degree = 7

           X = df2.iloc[:,3:].values # Features
           Y = df2.iloc[:,2].values # Unemp_rate
           X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = size_test)
           print("Polynomial regression:")

           for d in range(1, degree + 1):
               print("Degree = " + str(d), end="")
               polynomial_features = PolynomialFeatures(degree=d, interaction_only=False, include_bias=False)
               linear_regression = LinearRegression()
               X_poly = polynomial_features.fit_transform(X_train)
               model=linear_regression.fit(X_poly, Y_train)
               X_test_poly = polynomial_features.fit_transform(X_test)
               Y_test_pred = linear_regression.predict(X_test_poly)
               print("            Score = " + str(model.score(X_test_poly, Y_test)))
```

```
Polynomial regression:
Degree = 1            Score = 0.9626504840226857
Degree = 2            Score = 0.9628257890881936
Degree = 3            Score = 0.9289602963926248
Degree = 4            Score = 0.0721406635451366
Degree = 5            Score = -0.6276205535410182
Degree = 6            Score = -19.255345602966162
Degree = 7            Score = -24.09105071770711
```

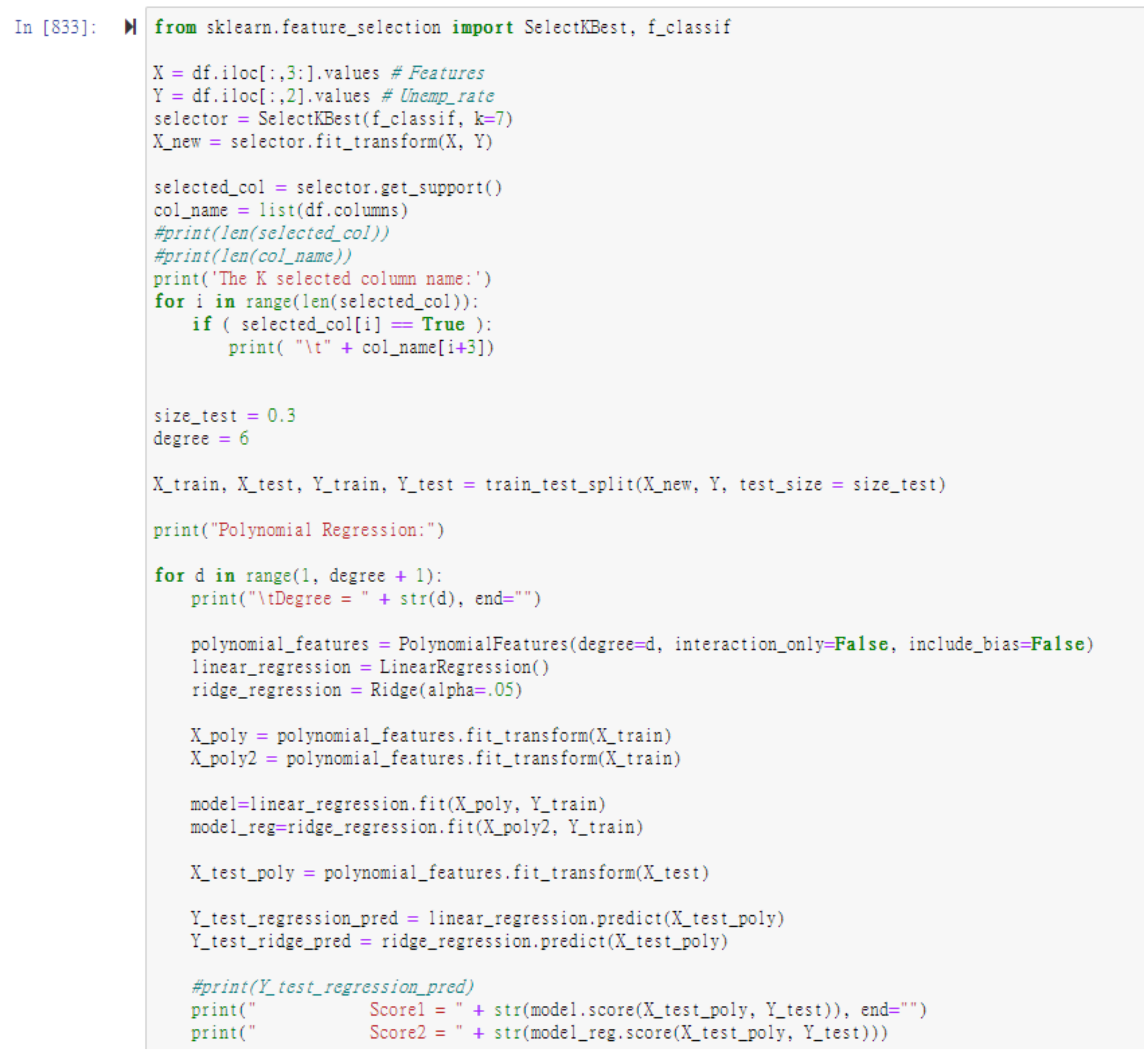
Part3:

In this part, we will implement the SelectKBest API with the same algorithm. As we discussed in section 3.3 part2, the following best 7 features are being selected: total_pop, 25-44yrs, weeklyearn, labour_demand, GDP, CPI and tpop_sex_ratio. After that, we fitted the features to linear regression (score1) and ridge regression (score2), in order to improve the accuracy.

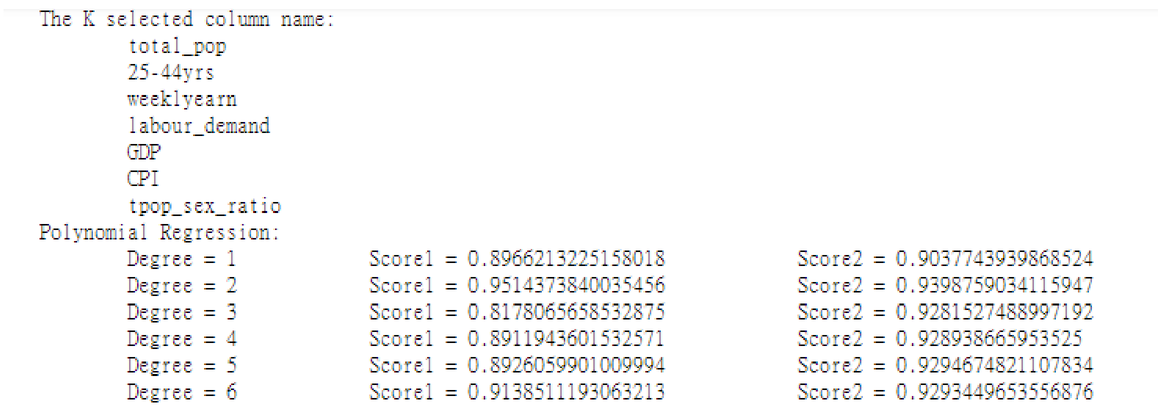
Ridge Regression:

Ridge Regression is a popular regression method with regularization. Ridge Regression is a technique for analyzing multiple regression data that suffer from multicollinearity. When multicollinearity occurs, least squares estimates are unbiased, but their variances are large so they may be far from the true value. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors.

From the output below, we can see that the scores for both regression methods became steadier and more consistent than the previous parts. In comparing between linear regression and ridge regression, it shows that ridge regression has a higher score in average. This may due to the reason of multicollinearity between features. Therefore, we have enough evidence that this is the better method in building our model.



```
In [833]:  from sklearn.feature_selection import SelectKBest, f_classif

           X = df.iloc[:,3:].values # Features
           Y = df.iloc[:,2].values # Unemp_rate
           selector = SelectKBest(f_classif, k=7)
           X_new = selector.fit_transform(X, Y)

           selected_col = selector.get_support()
           col_name = list(df.columns)
           #print(len(selected_col))
           #print(len(col_name))
           print('The K selected column name:')
           for i in range(len(selected_col)):
               if ( selected_col[i] == True ):
                   print( "\t" + col_name[i+3])


           size_test = 0.3
           degree = 6

           X_train, X_test, Y_train, Y_test = train_test_split(X_new, Y, test_size = size_test)

           print("Polynomial Regression:")

           for d in range(1, degree + 1):
               print("\tDegree = " + str(d), end="")

               polynomial_features = PolynomialFeatures(degree=d, interaction_only=False, include_bias=False)
               linear_regression = LinearRegression()
               ridge_regression = Ridge(alpha=.05)

               X_poly = polynomial_features.fit_transform(X_train)
               X_poly2 = polynomial_features.fit_transform(X_train)

               model=linear_regression.fit(X_poly, Y_train)
               model_reg=ridge_regression.fit(X_poly2, Y_train)

               X_test_poly = polynomial_features.fit_transform(X_test)

               Y_test_regression_pred = linear_regression.predict(X_test_poly)
               Y_test_ridge_pred = ridge_regression.predict(X_test_poly)

               #print(Y_test_regression_pred)
               print("            Score1 = " + str(model.score(X_test_poly, Y_test)), end="")
               print("            Score2 = " + str(model_reg.score(X_test_poly, Y_test)))
```



```
The K selected column name:
        total_pop
        25-44yrs
        weeklyearn
        labour_demand
        GDP
        CPI
        tpop_sex_ratio
Polynomial Regression:
        Degree = 1          Score1 = 0.8966213225158018          Score2 = 0.9037743939868524
        Degree = 2          Score1 = 0.9514373840035456          Score2 = 0.9398759034115947
        Degree = 3          Score1 = 0.8178065658532875          Score2 = 0.9281527488997192
        Degree = 4          Score1 = 0.891194360153257l          Score2 = 0.928938665953525
        Degree = 5          Score1 = 0.8926059901009994          Score2 = 0.9294674821107834
        Degree = 6          Score1 = 0.9138511193063213          Score2 = 0.9293449653556876
```

5. Evaluation:

As our model is built, in this section, we will work on predicting unemployment rate and check about the accuracy. We will use the training set to feed our model then use the test set for evaluation.

For accuracy, we used the relative squared error E:

$$E_i = \frac{\sum_{j=1}^{n} \left(P_{(ij)} - T_j\right)^2}{\sum_{j=1}^{n} \left(T_j - \bar{T}\right)^2}$$

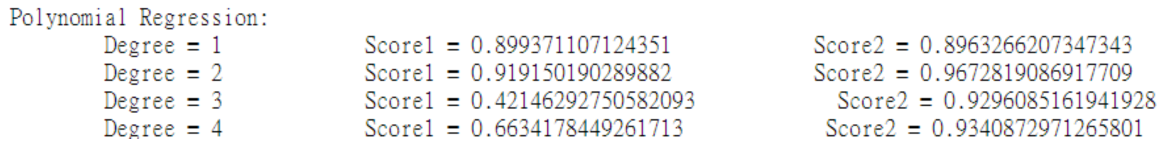
Determination of degree:

From the output set above, we can see that degree of 2 achieved the highest in both regression method. Therefore, we can conclude that degree of 2 is the optimal number of features.

Training set ratio:

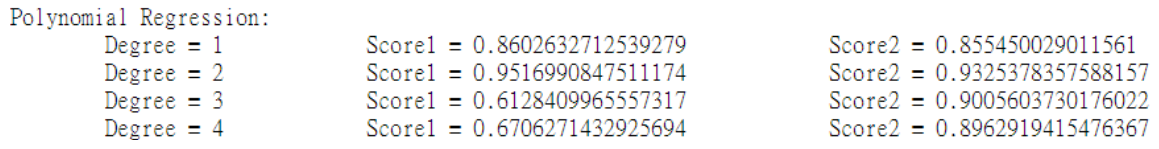
As we discussed in earlier section of data mining, we used a ratio of 7:3 for our model, which is commonly suggested in the field of machine learning. In this part, we will test on different ratio and how will they affect the accuracy.

Ratio 7:3



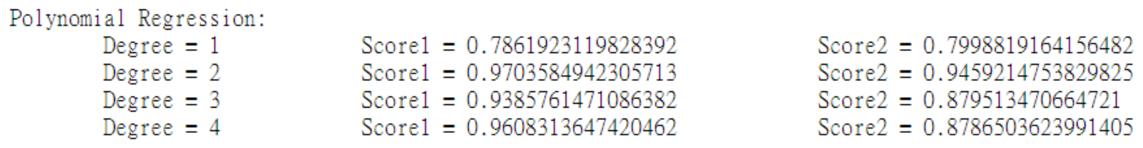
```
Polynomial Regression:
        Degree = 1           Score1 = 0.899371107124351           Score2 = 0.8963266207347343
        Degree = 2           Score1 = 0.919150190289882           Score2 = 0.9672819086917709
        Degree = 3           Score1 = 0.42146292750582093           Score2 = 0.9296085161941928
        Degree = 4           Score1 = 0.6634178449261713           Score2 = 0.9340872971265801
```

Ratio 8:2



```
Polynomial Regression:
        Degree = 1           Score1 = 0.8602632712539279           Score2 = 0.855450029011561
        Degree = 2           Score1 = 0.9516990847511174           Score2 = 0.9325378357588157
        Degree = 3           Score1 = 0.6128409965557317           Score2 = 0.9005603730176022
        Degree = 4           Score1 = 0.6706271432925694           Score2 = 0.8962919415476367
```

Ratio 9:1



```
Polynomial Regression:
        Degree = 1           Score1 = 0.7861923119828392           Score2 = 0.7998819164156482
        Degree = 2           Score1 = 0.9703584942305713           Score2 = 0.9459214753829825
        Degree = 3           Score1 = 0.9385761471086382           Score2 = 0.879513470664721
        Degree = 4           Score1 = 0.9608313647420462           Score2 = 0.8786503623991405
```

Overall, we can see that the highest accuracy for our model is the ratio 7:3, with an average around 93%.

Range of accuracy:

Since the splitting function for separating training and testing set is in random. Therefore, it is worth to run a few times.

Size of dataframe: 95
Degree: 2
Ratio: 7:3

Trial	Score1 (linear regression)	Score2 (ridge regression)
1	0.9793120271576372	0.9696575438690024
2	0.9236194386914384	0.9042197919579591
3	0.8870754311782825	0.8914708679917895
4	0.9049846538997381	0.9020524886185878
5	0.8362528072316602	0.8881815070491231

The mean of score1 is 90.6% and score2 is 91.1%

6. Conclusion:

The purpose of this project is predicting the unemployment rate. During this project, we used different techniques such as feature engineering, regression model etc. The most useful tools in the feature selection is the Pearson Correlation coefficient, which effectively point out the multicollinearity that may happened in our dataset. Furthermore, we also used ridge regression that helped us to obtain a more consistent result due to the issue of multicollinearity.

Overall, the result of this project shows that we can get the highest accuracy by building a model with ratio of 7:3 and use the degree of 2. Although the splitting of training and test set is random, we eventually achieved an average of 91.1% accuracy, which is significantly high.

7. References:

- Statistics Canada, sex and age group dataset, available at:
  https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=1410001701

- Statistics Canada, Job vacancies, labour demand dataset
  https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=1410022401

- Statistics Canada, educational degree dataset
  https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=1410011701

- Statistics Canada, Reason for leaving job dataset
  https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=1410012501

- Statistics Canada, immigrant status dataset
  https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=1410008201

- Wikipedia, Pearson correlation coefficient
  https://en.wikipedia.org/wiki/Pearson_correlation_coefficient

- SelectKBest:
  https://scikit-
  learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html#s
  klearn.feature_selection.SelectKBest.get_params

- Saedsayad.com. (2018). Model Evaluation. [online] Available at:
  https://www.saedsayad.com/model_evaluation_r.htm [Accessed 30 Nov. 2018].