Regression Analysis of Government Transfers Amount with Census Canada 2016 BC Dataset

CSIS 3290 – Term Project

Submitted to: Mr. Bambang Ali Basyah Sarif

Submitted by: Selim Ozdogan (300340479)

Date of Submission: August 2, 2021

The project "Regression Analysis of Government Transfers Amount with Census Canada 2016 for British Columbia Dataset" is aimed to make Government Transfers' regression using Census Canada 2016 BC dataset. This goal helps the government of BC to arrange the next years' budget for the transfers. For this purpose, the project can support all decision-makers of the government who are related to the budget. The dataset has almost 19 million data points and 14 features that are using in the project. My initial hypotheses are the dataset has education, household income and population of the place and they can help to find government transfers. The dataset has huge features and data points, clustering and feature selection are important for the models.

Jupyter Notebooks

The project has 2 jupyter notebooks which are Project_SOz40479-EDA and Project_SOz40479-FeatureEngineering. EDA is for preparing the data and it saves a file for the Feature Engineering file which is cleanedCensusData.csv. Feature engineering file opens the file and work on the cleaned data. All files are in the dataset folder, but if you want to run all process, the EDA file should run first.

Data Preparation

The dataset is multi-dimensional, 2247 features are in the rows and 14 features are in columns.
The data as shown as below.

GEO_LEVEL	GEO_NAME	GNR	GNR_LF	DATA_QUALITY_FLAG	ALT_GEO_CODE	DIM: Profile of Dissemination Areas (2247)	Member ID: Profile of Dissemination Areas (2247)	Notes: Profile of Dissemination Areas (2247)	Dim: Sex (3): Member ID: [1]: Total - Sex	Dim: Sex (3): Member ID: [2]: Male	Dim: Sex (3): Member ID: [3]: Female
0	Canada	4.0	5.1	20000	1	Population, 2016	1	1.0	35151728	...	...
0	Canada	4.0	5.1	20000	1	Population, 2011	2	2.0	33476688	...	...
0	Canada	4.0	5.1	20000	1	Population percentage change, 2011 to 2016	3	NaN	5.0	...	...
0	Canada	4.0	5.1	20000	1	Total private dwellings	4	3.0	15412443	...	...
0	Canada	4.0	5.1	20000	1	Private dwellings occupied by usual residents	5	4.0	14072079	...	...

Important columns of the dataset

- GEO_NAME has location of data
- GEO_LEVEL represent of size of location such as 0 means country, 1 is province
- DIM: Profile of Dissemination Areas (2247) has 2247 different features
- Member ID: Profile of Dissemination Areas (2247) has IDs of features
- Dim: Sex (3): Member ID: [1]: Total – Sex has value of features for all sex
- Dim: Sex (3): Member ID: [2]: Male has value of features for male
- Dim: Sex (3): Member ID: [3]: Female has value of features for female

```
CENSUS_YEAR                                            int64
GEO_CODE (POR)                                         int64
GEO_LEVEL                                              int64
GEO_NAME                                              object
GNR                                                  float64
GNR_LF                                               float64
DATA_QUALITY_FLAG                                      int64
ALT_GEO_CODE                                           int64
DIM: Profile of Dissemination Areas (2247)            object
Member ID: Profile of Dissemination Areas (2247)       int64
Notes: Profile of Dissemination Areas (2247)         float64
Dim: Sex (3): Member ID: [1]: Total - Sex             object
Dim: Sex (3): Member ID: [2]: Male                    object
Dim: Sex (3): Member ID: [3]: Female                  object
dtype: object
```

Figure 1: Types of columns

```
CENSUS_YEAR                                                0
GEO_CODE (POR)                                             0
GEO_LEVEL                                                  0
GEO_NAME                                                   0
GNR                                                  1343706
GNR_LF                                               1377411
DATA_QUALITY_FLAG                                          0
ALT_GEO_CODE                                               0
DIM: Profile of Dissemination Areas (2247)                 0
Member ID: Profile of Dissemination Areas (2247)           0
Notes: Profile of Dissemination Areas (2247)        16954470
Dim: Sex (3): Member ID: [1]: Total - Sex                  0
Dim: Sex (3): Member ID: [2]: Male                         0
Dim: Sex (3): Member ID: [3]: Female                       0
dtype: int64
```

Figure 2: Number of null values

Important columns do not have null value. Firstly, I need to drop unnecessary columns and rename the rest for better understandable. After that, the data should be transposed.

	RegionID	RegionLevel	FeatureID	FeatureValue
0	1	0	1	35151728
1	1	0	2	33476688
2	1	0	3	5.0
3	1	0	4	15412443
4	1	0	5	14072079

Figure 3: After drop and rename columns
I will use pivot function to transpose data

```
df.reset_index(inplace=True, drop=True)
df = df.pivot(index=['RegionLevel','RegionID'],
                   columns= 'FeatureID',
                   values='FeatureValue')
df.to_csv(r'.\dataset\temp.csv')
df =  pd.read_csv(r'.\dataset\temp.csv')
```

Figure 4: Pivot function and parameters

	RegionLevel	RegionID	1	2	3	4	5	6	7	8	...	2238	2239	2240	2241	2242	2
0	0	1	35151728	33476688	5.0	15412443	14072079	3.9	8965588.85	35151730	...	372475	32568565	20134760	12433805	6755630	5678
1	1	59	4648055	4400057	5.6	2063417	1881969	5.0	922503.01	4648055	...	68860	4339960	2490535	1849420	940895	908
2	2	5901	60439	56685	6.6	34197	25863	2.2	27541.84	60440	...	350	56175	35040	21130	9920	11
3	2	5903	59517	58441	1.8	30726	27016	2.7	22094.94	59515	...	340	55815	35855	19960	8835	11
4	2	5905	31447	31138	1.0	18321	14339	3.9	8084.52	31445	...	75	29405	19610	9795	3980	5

5 rows × 2249 columns

Figure 5: After transpose data

After transpose data of Census has 8385 data points and 2249 features. The data has invalid valu es those are below

- .. not available for a specific reference period
- ... not applicable
- F too unreliable to be published
- x suppressed to meet the confidentiality requirements of the Statistics Act

To Eliminate masked values, I change the value with np.nan and their data types will be change by float. After that I will apply some drop to rows and columns with threshold if they have lots of nan value, they will be removed from dataset.

```
# replace symbols to nan
df.replace('x', np.nan, inplace=True)
df.replace('F', np.nan, inplace=True)
df.replace('..', np.nan, inplace=True)
df.replace('...', np.nan, inplace=True)

#convert types to float
for column in df.columns:
    df[column] = df[column].astype(float)
```

```
# dropping some columns and rows whics has NaN value above thresh
df.dropna(thresh=5000, axis=1, inplace=True)
df.dropna(thresh=2000, axis=0, inplace=True)
```

Figure 6: Replacing data with np.nan and drop rows and columns

Finally, I will create "Government Transfers" column with "679 Number of government transfers recipients aged 15 years and over in private households - 25% sample data" and "680 Average government transfers in 2015 among recipients ($)". Their multiplication should be provided total amount of Government payment as a benefit.

```
df['GovernmentTransfers'] = df['679'] * df['680']
df.drop(columns=['679','680','RegionID'], axis=1, inplace= True)
```

Figure 7: Creating the response column (Government Transfers)

	RegionLevel	1	4	5	6	7	8
ount	7755.00	7755.00	7755.00	7755.00	7755.00	7755.00	7755.00
nean	3.92	6928.65	3050.73	2784.82	3460.89	1619.07	6928.65
std	0.28	403790.22	177042.30	161661.78	7309.85	102421.10	403790.25
min	0.00	40.00	12.00	12.00	0.00	0.00	40.00
25%	4.00	447.00	180.00	168.00	312.90	0.14	445.00
50%	4.00	551.00	237.00	219.00	2108.40	0.28	550.00
75%	4.00	738.00	341.00	310.00	4174.35	1.87	740.00
max	4.00	35151728.00	15412443.00	14072079.00	454782.60	8965588.85	35151730.00

Figure 8: Describe of the data frame

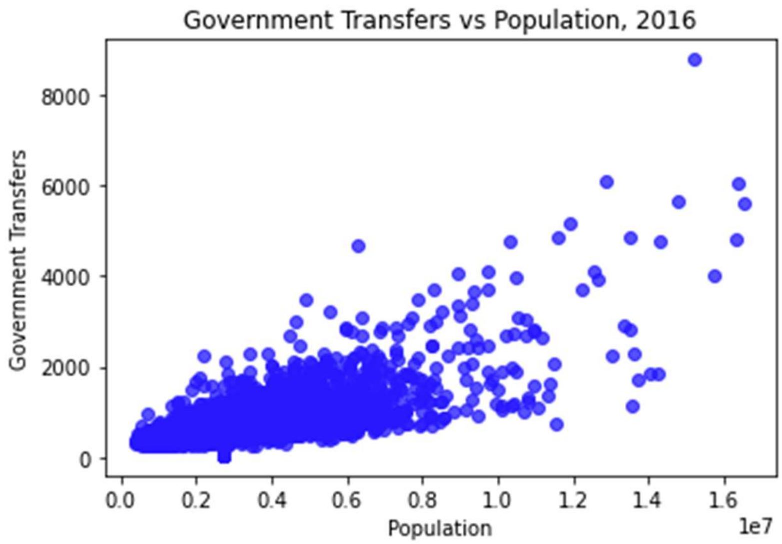

Figure 9: Government Transfers vs Population, 2016(There are good correlation)

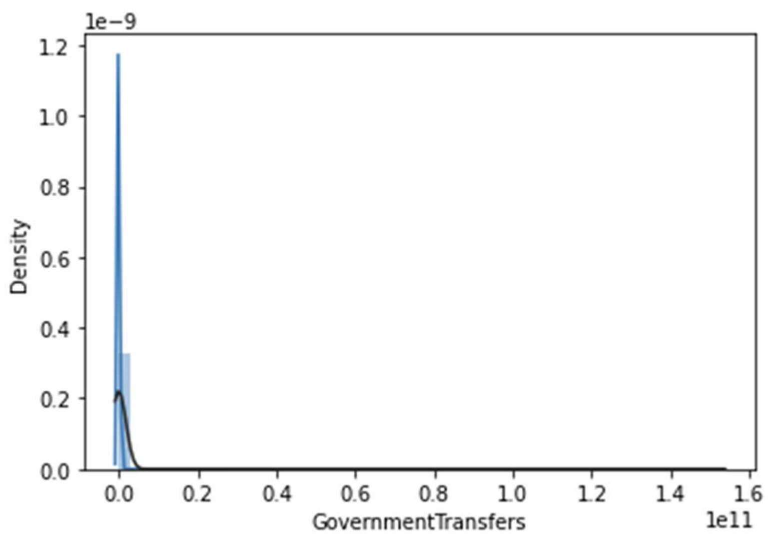

Figure 10: Government Transfers distribution before drop some Region Levels

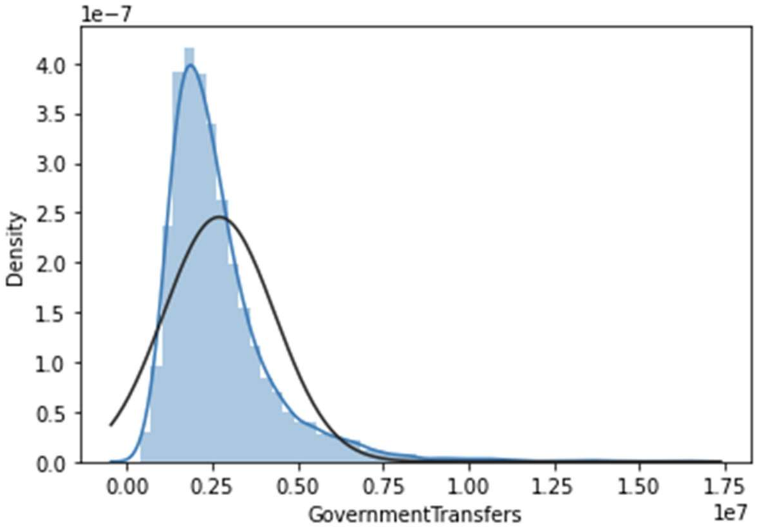

Figure 11: Government Transfers distribution. After drop if Region Level is greater than 4

Lastly, fill nan value with mean and than save file as a cleanedCensusData.csv to use for feature engineering.

Feature Engineering & Model

I will use StandardScaler, RobustScaler for scaling, for feature elimination I will use PCA and Lasso(it has own feature elimination algorithm). I will use LinearRegression, DecisionTreeRegressor, GradientBoostingRegressor, RandomForestRegressor, AdaBoostRegressor, XGBRegressor, Rigge, Lasso for regressor analysis. Therefore, I will split data 75% for train and 25% for test the model.

	Scaler	Regressor	R^2 score	RMSE
0	RobustScaler	LinearRegression	0.961	293920.175
1	RobustScaler	DecisionTreeRegressor	0.743	750272.276
2	RobustScaler	GradientBoostingRegressor	0.950	331782.744
3	RobustScaler	RandomForestRegressor	0.944	349999.939
4	RobustScaler	AdaBoostRegressor	0.883	506424.547
5	RobustScaler	XGBRegressor	0.940	363439.972
6	RobustScaler	Lasso	0.954	316371.552
7	RobustScaler	Ridge	0.964	282751.242
8	StandardScaler	LinearRegression	0.960	294325.339
9	StandardScaler	DecisionTreeRegressor	0.743	750272.276
10	StandardScaler	GradientBoostingRegressor	0.950	331871.941
11	StandardScaler	RandomForestRegressor	0.943	354589.134
12	StandardScaler	AdaBoostRegressor	0.882	507664.675
13	StandardScaler	XGBRegressor	0.940	363249.026
14	StandardScaler	Lasso	0.952	324185.603
15	StandardScaler	Ridge	0.963	285225.131
16	MinMaxScaler	LinearRegression	0.960	294325.339
17	MinMaxScaler	DecisionTreeRegressor	0.743	750272.276
18	MinMaxScaler	GradientBoostingRegressor	0.950	332229.531
19	MinMaxScaler	RandomForestRegressor	0.942	356476.941
20	MinMaxScaler	AdaBoostRegressor	0.880	512740.517
21	MinMaxScaler	XGBRegressor	0.940	363363.399
22	MinMaxScaler	Lasso	0.961	291662.910
23	MinMaxScaler	Ridge	0.962	288970.890

Figure 12: The best model Robust Scaler and Ridge Regressor (Max R^2cand Min RMSE)

According to https://scikit-learn.org/, Standard Scaler is recommended for Ridge Regressor.

Clustering

Elbow should be on the 3 when use elbow method (Figure 13), and when I check the Dendogram cluster should be 3 also according to dengram (Figure 14)

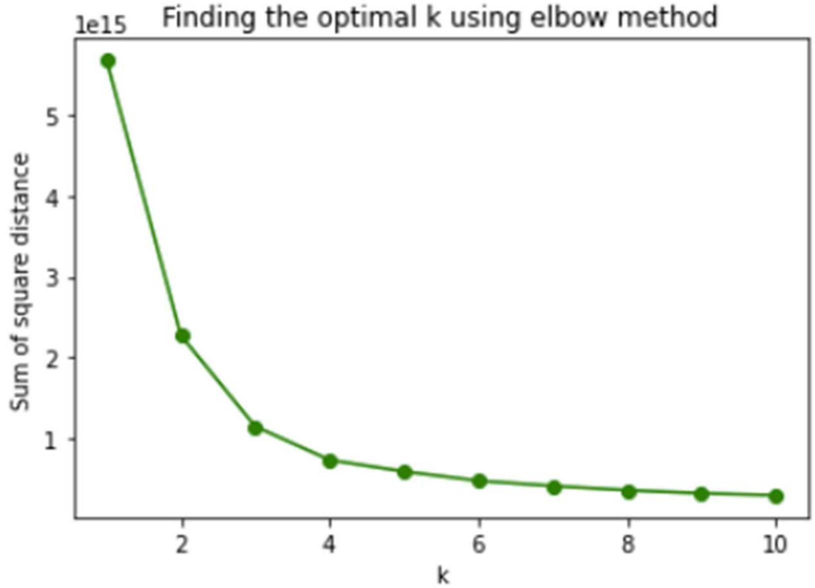

Figure 13: Elbow method

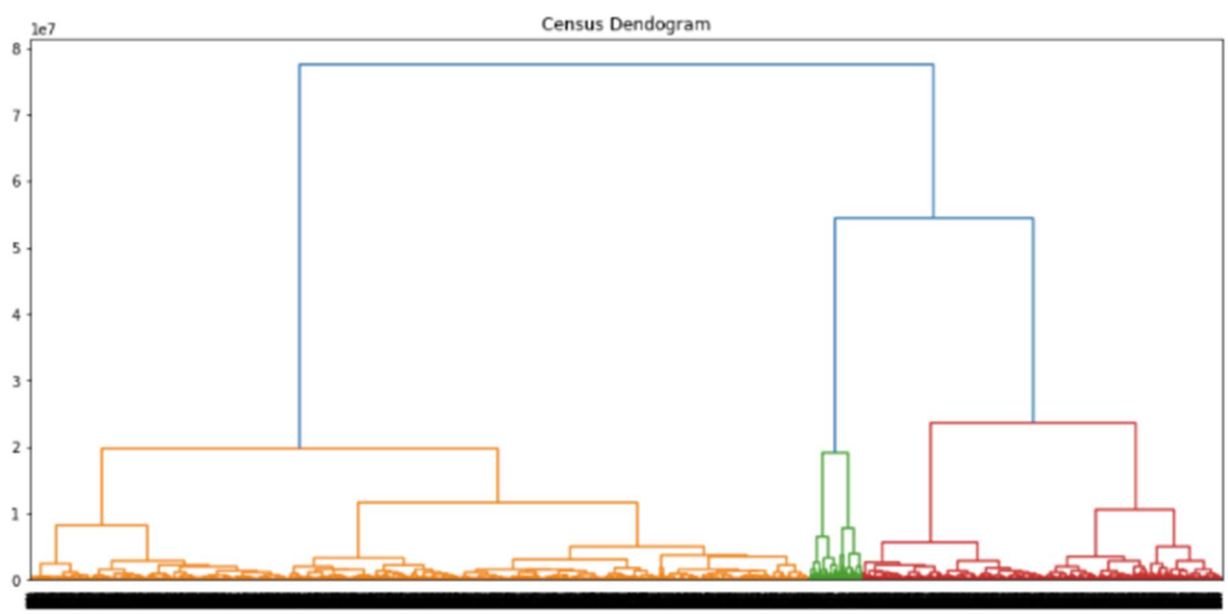

Figure 14: Dendogram

Grid Search

In Grid Search, I will check K-Means and Brich cluster with 3 clusters and solver with different parameters, also I will check scaler with Ridge's normalize feature and Standard scaler.

The result is the best score: 0.94 with the best parameters: {'norm': 'Stand', 'cluster': 'Birch ', 'solver': 'auto', 'alpha': '0.36', 'max_iter': 1000}

Tunning

I will tune the model with max_iter alpha, max_iter will be between 1000 and 100K with 10 different values. Alpha will be between 0.001 and 1000 with 50 different value. The cluster will be 3 and component of PCA will be 0.9. As a result, the best score is 0.94 and the best parameters: {'alpha': '59.64', 'max_iter': 1000.0}

Evaluation

The model is at Figure 15, Standard Scaler and PCA are as a preprocessing and Birch c luster with Ridge Regressor. The hyperparameters are number of clusters:3 for Birch, component s: 0.9 for PCA, max_iter: 100000 and alpha: 59.64 for Ridge. The scores of the model are R2: 0. 93 RMSE: 378735.45

```
n_clusters = 3
n_components = 0.9
alpha = 59.64
max_iter = 100000

preprocessing = Pipeline(
[
    ("scaler", StandardScaler()),
    ("pca", PCA(n_components=n_components))
]
)
regression = Pipeline(
[
    ('cluster', Birch(n_clusters= n_clusters)),
    ('Ridge',Ridge(max_iter= max_iter,
                 normalize = False,
                 alpha = alpha,
                 random_state = 42 )),
])
pipe = Pipeline(
[
    ("preprocessing", preprocessing),("regression", regression)
]
)

pipe.fit(X_train, y_train)
score = pipe.score(X_test, y_test)
pred = pipe.predict(X_test)

rmse = mean_squared_error(y_test, pred)**0.5

print("R2: {:.2f}".format(score))
print("RMSE: {:.2f}".format(rmse))
```

Figure 15: The Model

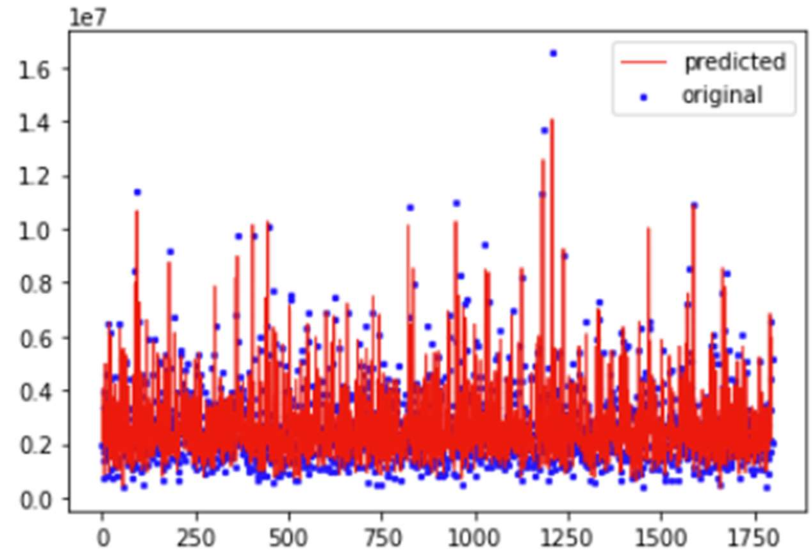

Figure 16: Prediction vs Actual Data

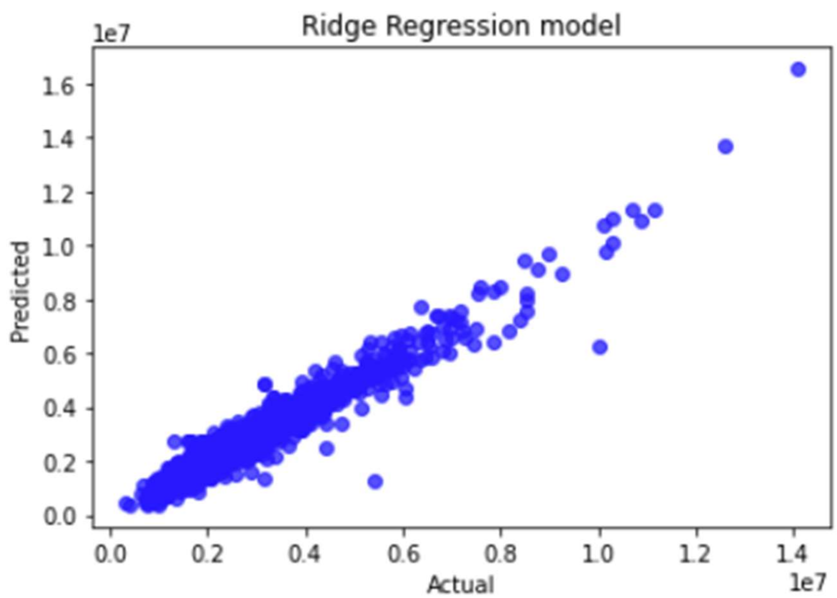

Figure 17: Prediction vs Actual Data

According to the plots (Figure 16 and Figure 17), the model works properly, and the regression is linear. The prediction and the actual value pretty similar for the Out of Sample data

Prediction: CAD 2699568.55

Actual: CAD 2708873.18

The Error is 0.0034

Dataset

- Name: Canada, provinces, territories, census divisions (CDs), census subdivisions (CSDs) and dissemination areas (DAs) - British Columbia only
- The page link: Census Profile 2016
- Dataset link: Census Canada 2016 for British Columbia