ECON3203 Group Assignment
10th November 2021

z5308483
z5310901
z5297724
z5248219

ATM Data Analysis Report

Min-Sik Son, Tony Min, Seoyeon Park, Joseph Chung

1. Executive Summary

1.1. Overview

Given a dataset, we were tasked to develop a model that predicted ATM cash demand for our bank - optimising the bank's cash management. In this report, you will find detailed analysis of various predictive models and a thorough comparison between each. Through extensive and articulate programming in Python, the optimal model with the smallest test error is derived.

1.2. Strategy

Our strategy involves 5 stages: initial analysis, modelling, testing and conclusion. We first explore the dataset and familiarise ourselves with each variable. Then we develop various models and train them with a subset of the dataset. These models are then tested with the remaining data to determine the effectiveness of them. The models we trialled were:

- Standard MLR
- MLR with interaction terms
- Lasso regression
- Ridge regression
- Elastic Net regression
- K-Nearest Neighbours

1.3. Conclusion

When analysing various methods for model selection, we found that the Kth-Nearest Neighbour model produced the lowest mean-squared error on our testing data subset. With the lowest MSE of 0.3156, the model with predictors Shops, Weekday, Center, High, ATMs, Downtown produced the most accurate predictions on our data.

2. Exploratory Data Analysis

2.1. Familiarisation of Variables

Our dataset came with 7 variables. The variable *Withdraw* was set as our independent response variable and *Shops*, *ATMs*, *Downtown*, *Weekday*, *Center* and *High* were our covariate variables. Below are descriptions of each variable in the dataset.

Variable	Description
Withdraw	The total cash withdrawn a day (in 1000 local currency)
Shops	Number of shops/restaurants within a walkable distance (in 100)
ATMs	Number of other ATMs within a walkable distance (in 10)
Downtown	= 1 if the ATM is in downtown, 0 if not
Weekday	= 1 if the day is weekday, 0 if not
Center	= 1 if the ATM is located in a center (shopping, airport, etc), 0 if not
High	= 1 if the ATM has a high cash demand in the last month, 0 if not

2.2. Data Quality

To check if the quality of the given dataset was acceptable, we perform several basic Python functions. We check characteristics such as null values, variable names and data types to determine the data quality.

a. We use the data.head( ) function to check the appropriateness of the variable names.

	Shops	ATMs	Downtown	Weekday	Center	High	Withdraw
0	10.18	10	1	0	0	0	72.750556
1	9.74	10	1	1	0	0	66.720482
2	0.96	2	0	0	0	1	19.189516
3	9.58	9	1	1	0	1	67.388669
4	1.03	4	0	1	0	1	15.813127

b. Null observations and uniform data types are checked by the data.info( ) function.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22000 entries, 0 to 21999
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Shops     22000 non-null  float64
 1   ATMs      22000 non-null  int64  
 2   Downtown  22000 non-null  int64  
 3   Weekday   22000 non-null  int64  
 4   Center    22000 non-null  int64  
 5   High      22000 non-null  int64  
 6   Withdraw  22000 non-null  float64
dtypes: float64(2), int64(5)
memory usage: 1.2 MB
```

Because the variables are appropriately named, have consistent data types and have no null observations, we conclude that our dataset is clean and ready for analysis.

2.3. Initial Analysis

An initial analysis of the data is utilised to identify basic information of different relationships between the variables. We calculate the correlations between each of the variables to identify strong influencers to *Withdraw*, and also find any sources of multicollinearity that may affect any of our findings.

a. We calculate the correlation coefficients using the data.corr( ) function.

	Shops	**ATMs**	**Downtown**	**Weekday**	**Center**	**High**	**Withdraw**
Shops	1.000000	0.872903	0.999131	0.013014	0.000004	0.001820	0.985797
ATMs	0.872903	1.000000	0.873726	0.009766	-0.003306	-0.002616	0.824030
Downtown	0.999131	0.873726	1.000000	0.012664	-0.000101	0.001782	0.983574
Weekday	0.013014	0.009766	0.012664	1.000000	-0.007153	-0.006793	-0.050470
Center	0.000004	-0.003306	-0.000101	-0.007153	1.000000	0.010521	0.088103
High	0.001820	-0.002616	0.001782	-0.006793	0.010521	1.000000	0.021275
Withdraw	0.985797	0.824030	0.983574	-0.050470	0.088103	0.021275	1.000000

Here, we notice that *Shops* and *ATMs* have a high correlation coefficient of 0.8729. Furthermore, we can see that *Shops* and *Downtown* also have a high correlation coefficient of 0.9991. Thus, we can keep in mind that statistically significant interaction terms should be created using these terms to produce a good model. Moreover, *Shops* and *Downtown* have a very high correlation very close to 1 which may indicate a multicollinearity issue.

2.4. Training and Testing Data

Given the csv file *"ATM_training.csv"*, the dataset was split into a training and testing set where 70% of the data was used for training and the remaining 30% was used for testing.

2.5. Interaction Terms

Adding interaction terms to our data creates a more complex relationship between the variables and allows more hypotheses to be tested. As such we will create interaction terms for further Forward-stepwise selection.

Consider predictors 'X' and 'Y'. We create interaction terms such as X^2 and XY. For X^2, we avoid using binary predictors as this will result in no changes. In addition, for XY, we create the product terms of highly correlated predictors to potentially improve the model. As such we create the interaction terms below for both the training and testing sets:

If we have a look at the correlation between 'Shops' and 'Downtown', there is a very high correlation between these two features. Similarly for 'Downtown' and 'ATMs'. Including these features could overfit our model and raise issues to multicollinearity. For now, Shops, Downtown, ATMs are highly correlated predictors so we will now attempt to create interaction terms with them.

X^2 Interaction Terms	XY Interaction Terms
Shops_SQ = Shops ✕ Shops	Shops_ATM = Shops ✕ ATMs
ATM_SQ = ATM ✕ ATM	Shops_Down = Shops ✕ Downtown
	Down_ATM = Downtown ✕ ATMs

2.6. Standardisation

We want to standardise predictors in our model so that we can put different variables on the same scale. This allows us to compare scores between different types of variables. This will require us to achieve zero mean and unit standard deviation after standardisation.

```
Mean for each numerical predictor:
Shops          6.057701e-15
ATMs          -7.033335e-17
Downtown       3.590202e-18
Weekday        4.481841e-16
Center         3.077120e-16
High           1.990471e-16
Shops_ATM      1.554817e-15
Shops_Down     4.213873e-16
Down_ATM       5.441535e-17
Shops_SQ      -4.303519e-14
ATM_SQ         8.380021e-17
```

```
Standard deviation for each numerical predictor:
Shops          1.0
ATMs           1.0
Downtown       1.0
Weekday        1.0
Center         1.0
High           1.0
Shops_ATM      1.0
Shops_Down     1.0
Down_ATM       1.0
Shops_SQ       1.0
ATM_SQ         1.0
```

As shown above, we have successfully standardised the data with zero mean and unit standard deviation for each numerical predictor.

3. Multiple Linear Regression

3.1. Full Model

The first model we produce is the full model including all predictors. This allows us to set a benchmark on all relevant values such as the R-squared value and p-values etc. Below is the summary output of the full model:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               Withdraw   R-squared:                       0.990
Model:                            OLS   Adj. R-squared:                  0.990
Method:                 Least Squares   F-statistic:                 2.542e+05
Date:                Thu, 11 Nov 2021   Prob (F-statistic):               0.00
Time:                        20:12:25   Log-Likelihood:                -36054.
No. Observations:               15400   AIC:                         7.212e+04
Df Residuals:                   15393   BIC:                         7.218e+04
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     54.5864      0.020   2692.881      0.000      54.547      54.626
Shops         44.8573      0.489     91.711      0.000      43.899      45.816
ATMs          -3.7446      0.042    -89.911      0.000      -3.826      -3.663
Downtown     -16.7871      0.491    -34.215      0.000     -17.749     -15.825
Weekday       -1.5951      0.020    -78.674      0.000      -1.635      -1.555
Center         2.1982      0.020    108.420      0.000       2.158       2.238
High           0.4355      0.020     21.482      0.000       0.396       0.475
==============================================================================
Omnibus:                    12276.887   Durbin-Watson:                   2.015
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           312019.589
Skew:                           3.724   Prob(JB):                         0.00
Kurtosis:                      23.756   Cond. No.                         57.5
==============================================================================
```

From the summary we see that the p-values for Shops, ATMs, Downtown, Weekday, Center, High are very close to zero. Hence, at a 0.05 level of significance, we reject the null hypothesis that the covariate variables had no significant influence on the amount withdrawn and choose the alternate hypothesis. We can then conclude that all variables are significant predictors.

Using the MLR Test MSE function built in the notebook, we computed a Test MSE value of 6.2527.

With this model, we observe that all predictors are significant at a 0.05 level of significance. We also observe that the R squared value of the model is 0.990 which could imply that our regression model first observed the data really well. However, a high R squared does not mean we can assume it is a good model as the r-squared value increases as model complexity increases. Considering the high correlation coefficients explored previously, this model may be overfitted or be including a different form of the same variable.

The second model is the full model containing all original covariates from the dataset as well as all the interaction terms built earlier.

```
                            OLS Regression Results                            
==============================================================================
Dep. Variable:               Withdraw   R-squared:                       0.990
Model:                            OLS   Adj. R-squared:                  0.990
Method:                 Least Squares   F-statistic:                 1.401e+05
Date:                Thu, 11 Nov 2021   Prob (F-statistic):               0.00
Time:                        20:12:25   Log-Likelihood:                -35975.
No. Observations:               15400   AIC:                         7.197e+04
Df Residuals:                   15388   BIC:                         7.207e+04
Df Model:                          11                                         
Covariance Type:            nonrobust                                         
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     54.5864      0.020   2706.308      0.000      54.547      54.626
Shops         10.7079      4.660      2.298      0.022       1.573      19.843
ATMs          -3.4032      0.268    -12.675      0.000      -3.929      -2.877
Downtown     -37.7052     19.239     -1.960      0.050     -75.415       0.005
Weekday       -1.5937      0.020    -78.980      0.000      -1.633      -1.554
Center         2.1982      0.020    108.956      0.000       2.159       2.238
High           0.4341      0.020     21.515      0.000       0.395       0.474
Shops_SQ     -13.8475     19.066     -0.726      0.468     -51.219      23.524
ATM_SQ        -0.1621      0.228     -0.711      0.477      -0.609       0.285
Shops_ATM     -1.9575      2.800     -0.699      0.485      -7.447       3.532
Shops_Down    68.8390     34.824      1.977      0.048       0.580     137.099
Down_ATM       1.8246      2.612      0.699      0.485      -3.295       6.944
==============================================================================
Omnibus:                    12419.528   Durbin-Watson:                   2.017
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           324917.302
Skew:                           3.779   Prob(JB):                         0.00
Kurtosis:                      24.195   Cond. No.                     5.99e+03
==============================================================================
```

From the summary, the model containing all covariates (including interaction terms) has insignificant features. We observe that the feature 'Shops_SQ' is irrelevant in the presence of all other predictors since the p-value = 0.468 > 0.05, where the level of significance is 0.05. Similarly, we can say the same for predictors 'Downtown', 'ATM_SQ', 'Shops_ATM' and 'Down_ATM'. This implies that we are most likely overfitting our model and at least one feature is irrelevant in the model.

3.2. Test MSE of MLR Models

When comparing various models, it is important that we do so on a common scale. This ensures that the predictors used are consistent throughout all trialled models, giving a fair and valid comparison. In particular, this is vital for comparing MLR models to those from Lasso, Ridge and Elastic Net because standard MLR models use original data points but the latter three utilise standardised data points. Thus, building the MLR models with standardised data ensures a valid comparison between the four different models.

Using Forward-stepwise selection, we added predictors to the model to improve Test MSE:

Model	Test MSE
Withdraw ~ Shops	16.9889
Withdraw ~ Shops + ATMs	13.7953
Withdraw ~ Shops + ATMs + Downtown	13.3329
Withdraw ~ Shops + ATMs + Downtown + Weekday	10.9041
Withdraw ~ Shops + ATMs + Downtown + Weekday + Center	6.2874
Withdraw ~ Shops + ATMs + Downtown + Weekday + Center + High	6.0872

Finally, using Backward-stepwise selection, we include all predictors and interaction terms in the model and remove predictors that will present us a low Test MSE model:

Model	Test MSE
Withdraw ~ Shops + ATMs + Downtown + Weekday + Center + High + Shops_SQ + ATM_SQ + Down_ATM + Shops_ATM + Shops_Down	6.0351
Withdraw ~ Shops + ATMs + Downtown + Weekday + Center + High + Shops_SQ + ATM_SQ + Down_ATM + Shops_ATM	6.0327

Removing any more predictors will increase our Test MSE and thus, we conclude that our best MLR model with the lowest Test MSE is:

Withdraw ~ Shops + ATMs + Downtown + Weekday + Center + High + Shops_SQ + ATM_SQ + Down_ATM + Shops_ATM

4. Lasso, Ridge and Elastic Net

To mitigate the risk of multicollinearity from including *Shops*, *ATMs* and *Downtown* we use the Lasso, Ridge and ElasticNet methods. By cross-validating each method, we determine the optimal model by comparing the MSEs of each.

4.1. Lasso Regression

Lasso allows the regularisation term to penalise the absolute value of the coefficients as well as setting irrelevant values to 0. To find the optimal lambda for our model, we can use the LassoCV function, which is the most convenient implementation with built in CV-based model selection for tuning parameter α.

With an optimal Lasso lambda value of 0.0248, we derive the following coefficients in a dataframe:

	Shops	ATMs	Down-town	Week-day	Center	High	Shops_ ATM	Shops_ Down	Down_ ATM	Shops_ SQ	ATM_ SQ
Coefficient	11.648	-2.614	0	-1.571	2.174	0.413	0.000	0.000	0.000	16.051	-0.781

Recall earlier when we suspected an issue with multicollinearity when our predictors 'Shops' and 'Downtown' were highly correlated to each other and were possibly involved in an issue with multicollinearity. As shown above, we observe that the feature 'Downtown' has been shrunk to 0. Furthermore, other features such as 'Down_ATM', 'Shops_ATM' and 'Shops_Down' have also been shrunk to 0 and are interpreted as irrelevant features in the model.

4.2. Ridge Regression

We do a similar process as above for the Ridge regression method. Ridge penalises the size (square of the magnitude) of the regression coefficients and enforces the beta (slope/partial slope) coefficients to be lower, but not 0. This means ridge regression shrinks the coefficients and reduces the model complexity.

In order to train the ridge regression model with our data, we trial 500 consecutive alpha values ranging from -10 to 20 and cross validate the results to determine the optimal ridge lambda of 0.2180. We use this range to ensure that we test a broad enough spread for a valid derivation of the optimal alpha value.

From this, we derive the following coefficients for our ridge regression model:

	Shops	ATMs	Down-town	Week-day	Center	High	Shops_ ATM	Shops_ Down	Down_ ATM	Shops_ SQ	ATM_ SQ
Coefficient	5.638	-3.602	-3.671	-1.594	2.198	0.435	0.650	8.478	-0.594	17.595	-0.160

As noted earlier, Ridge does not shrink irrelevant features to 0 but instead minimises impact. We will observe the Test MSE computations later on for findings.

4.3. Elastic Net Regression

We also derive an Elastic Net regression which is a combination of Lasso and Ridge regressions. This is beneficial as it removes all unnecessary coefficients but not the informative ones.

To train the elastic net regression model, we trial 99 consecutive values between 0.01 and 0.99. We use a CV score of 5 to find the optimal alpha value and ratio of 0.0251 and 0.99 respectively.

Using the optimal values for the elastic net regression, we calculate the following coefficients after storing the model as a standard ElasticNet regression object, using the CV selected parameters:

	Shops	ATMs	Down-town	Week-day	Center	High	Shops_ATM	Shops_Down	Down_ATM	Shops_SQ	ATM_SQ
Coefficient	11.137	-2.582	0	-1.570	2.173	0.413	0	0	0	16.540	-0.794

4.4. Cross Validation of Lasso, Ridge, ElasticNet

To compare the results of all three methods, we perform a cross-validation using the mean-squared errors of each built model as our metric. The models are all tested on the same test data sectioned off from the training data set.

	Test MSE	CV MSE
Ridge	144.570	6.285
Lasso	46.415	6.346
Elastic Net	48.898	6.342

When comparing the Test MSE and CV MSE for all models, we observe for the Ridge model, Test MSE > CV MSE significantly which indicates an overfitting model. We note that Ridge regression does not shrink irrelevant features to 0 but rather 'minimises' their impact. As such, we see a much larger test error in Ridge compared to Lasso and Elastic Net models.

After assessing the Test MSE and CV MSE values for each model, we can rank the methods from best to worst as follows:

1. Lasso
2. Elastic Net
3. Ridge

Thus, we conclude that the Lasso regression model is the best model for predicting the amount withdrawn among the above three models.

Recall earlier, we saw before that the MLR model with interaction covariates has a smaller MSE of 6.032715 than the Lasso model MSE. Therefore, we conclude that the MLR model with interaction covariates is currently the best model for prediction amount withdrawn.

5. K-Nearest Neighbours Model

5.1. Test/CV MSE for KNN Model

We use an unstandardised dataset for KNN for the training and testing, *train_copy* and *test_copy* as we set our metric to be mahalanobis. A function is initialised by inputting a list of covariates and a response variable. This goes through the entire process of selecting the number of neighbours through cross validation and obtaining the test results for the selected model and returning the relevant Test/CV MSE.

We find that the covariate 'Shops' returns the lowest test MSE in the single covariate model.

Chosen K	49
Test MSE	16.4783
CV MSE	17.7755

We now have, *Withdraw ~ Shops* as our best model currently using Forward-stepwise selection.

Adding the next covariate variable, we see that the best model with two predictors is when 'Weekday' was added.

Chosen K	47
Test MSE	14.1126
CV MSE	14.8041

We now have, *Withdraw ~ Shops + Weekday.*

Adding the next unused covariate, we see that the best model with three predictors is when 'Center' is added which improves our earlier model significantly.

Chosen K	49
Test MSE	3.7725
CV MSE	3.7577

We now have, *Withdraws ~ Shops + Weekday + Center* as our best model.

Similarly, to the process above, the next addition to the model that improves the Test MSE and CV MSE is 'ATMs'. We observe that this change to our model significantly decreases our mean squared errors for the Test and Cross validation shown below:

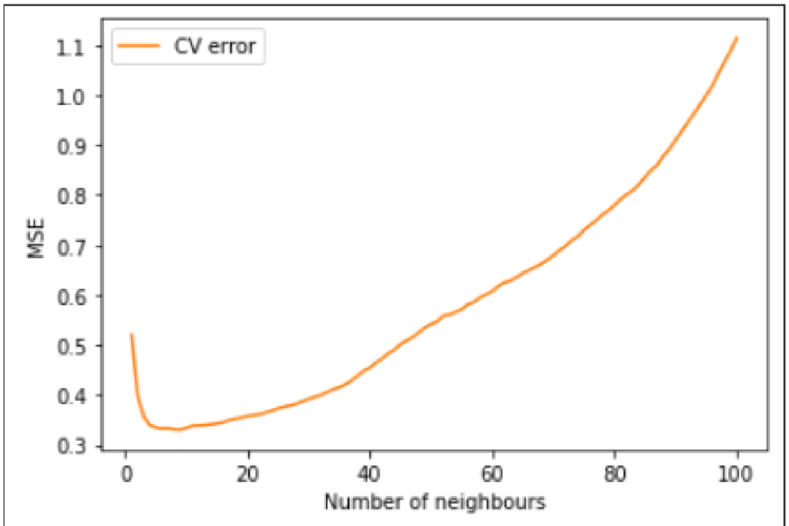
Chosen K	8
Test MSE	0.5937
CV MSE	0.5837

We now have, *Withdraw ~ Shops + Weekday + Center + ATMs.*
Continuing the Forward-stepwise selection, the best subset model containing 5 covariates is *Withdraw ~ Shops + Weekday + Center + ATMs + High*

Chosen K	4
Test MSE	0.3961
CV MSE	0.3998

Finally, we include all predictors for the model:
Withdraw ~ Shops + Weekday + Center + ATMs + High + Downtown.

Chosen K	9
Test MSE	0.3156
CV MSE	0.3292




```
Lowest CV error: K = 9
```

We conclude that from the original covariates from the dataset, the best subset model is the model with all predictors. Alternatively, Backward-stepwise selection could've been used to select the best subset model, however, the disadvantage is that this does not always result in the best subset model. Nonetheless, this is a much more time-efficient method than the best subset selection method which searches over the total 2^6 possible subsets of 6 covariates to find the best subset.

5.2. Testing Interaction Terms in K-Nearest Neighbours Model

Earlier, we found our best model was the model containing all predictors. Continuing with Forward-stepwise selection we will test whether the interaction covariates improve the model.

From the X^2 interaction covariates, we trial to see which terms improve the model. We add the predictor 'Shops_SQ' to the previous model and compute the Test/CV MSE's below:.

Chosen K	8
Test MSE	0.3298
CV MSE	0.3470

As shown above, this did not improve our model as our Test MSE nor our CV MSE decreased. Thus, we will not add 'Shops_SQ' to our previous model.

Similarly, we find that 'ATM_SQ' does not improve the model either as it returns a Test MSE of **0.3263** and a CV MSE of **0.3401**. Furthermore, we test the XY interaction covariates and derive:

- Shops_ATM does not improve the model as it returns a Test MSE of 0.3187 and a CV MSE of 0.3318.
- Shops_Down does not improve the model as it returns a Test MSE of 0.3280 and a CV MSE of 0.3431.
- Down_ATM does not improve the model as it returns a Test MSE of 0.3162 and a CV MSE of 0.3288.

If we use Backward-stepwise selection we observe that for the model:

Withdraw ~ Shops + Weekday + Center + ATMs + High + Downtown + Shops_SQ + ATM_SQ + Shops_ATM + Shops_Down + Down_ATM,

the computation of the Test MSE and CV MSE is significantly higher than our best model,

Withdraw ~ Shops + Weekday + Center + ATMs + High + Downtown.

Chosen K	2
Test MSE	0.4319
CV MSE	0.5405

Thus, our interaction terms did not improve the current model

Withdraw ~ Shops + Weekday + Center + ATMs + High + Downtown.

As such, we conclude the model above holds the lowest test errors for Test MSE and CV MSE.

6. Conclusion

After exploring various model selection methods, we find that the Kth Nearest Neighbours model with k = 9 and predictors: *Shops, Weekday, Center, High, ATMs* and *Downtown* performs the best in relation to the smallest test mean squared error.

The risk of multicollinearity due to the high correlation coefficients between predictors was our main concern during our investigations. But after trialling multiple interaction terms and exploring relevant statistics, we conclude that all covariate variables were in fact relevant to our predictions.

Model Selection Method	Test MSE
Multiple Linear Regression	6.0327
Ridge Regression	144.570
Lasso Regression	46.415
Elastic Net Regression	48.898
KNN Model	0.3156