Machine Learning and Pattern Recognition

Team 18

AI_3rd year

Project Report

Milestone One:

1. Preprocessing:

After loading the data and viewing it:

1. we started with Removing the missing values using dropna(axis=0, how='any') where axis=0 means rows and any how='any' means any missing values.
2. Then we dropped duplicate values using drop_duplicates(inplace=true) where inplace=true means removing them only in the current dataframe not the original dataset.
3. We tried one hot encoding, but there were errors so we switched to get_dummies as it was easier to implement.
4. We dropped the original 'Holiday' and 'Functioning Day' columns after creating dummies.
5. We implemented the idea of one hot encoding manually into the column 'Seasons' dividing it into 'Spring' 'Summer' 'Autumn' 'Winter' for both the train and test datasets

    - First we initialized each with zero the created a for loop containing four if conditions (one for each of the four seasons) where it makes the value of the season 1 for the season in the original column seasons ex: if newdf.at[i,'Seasons']=='Spring': newdf.at[i,'Spring']= 1
    - We repeated the same steps for both the train and test datasets.

6. Drop the column Seasons
7. We concatenated the encoded dataframes with the train dataframe then we also performed the same step for the test dataframe.
8. We dropped the index column for both train and test because it was a useless bi-product of the concatenate process.
9. We shuffled both the train and test data followed by the function reset_index to reset the indices to their default values.
10. We created a new dataframe called x_train and another called y_train and transefered the train dataset to x_train without the column 'Rented Bike Count' which was put in the y_train.
11. We created a new dataframe called x_test which contained all the test data after dropping the column 'ID'(the column was dropped in the new dataframe only)
12. We created a new dataframe for each of the four models, each included the 'ID' column
13. We graphed the data using q-q plot and found that the data doesn't follow the Gaussian distribution so we chose normalization for both x_train and x_test using MinMaxScaler.

1. Analyzing the dataset:

After viewing and analyzing the dataset, we found:

1. The dew point temperature is always less than or equal to the air temperature.
2. The number of rented bikes increase as the temperature increases making the number of rented bikes the highest in summer followed by spring.
3. The temperature is affected by other factors such as the humidity and wind speed.

1. Regression Techniques:

- **Ridge Regression:** is a model tuning method that is used to analyze any data that suffers from multicollinearity(independent variables are highly correlated) with a regularization term called alpha where alpha can cause overfitting (equals zero) or underfitting (equals infinity) depending on its value.

- **Lasso Regression:** is a type of linear regression that uses shrinkage which is when data values are shrunk towards a central point, like the mean.

    1. It's also similar to ridge in the fact that it's suited for multicollinearity however they solve multicollinearity issue differently where in ridge regression, the coefficients of correlated predictors are similar, while In lasso, one of the correlated predictors has a larger coefficient, while the rest are (nearly) zeroed.
    2. The difference between lasso and ridge models is that lasso tends to make coefficients to absolute zero as compared to Ridge which never sets the value of coefficient to absolute zero.

- **Elastic net:** it first emerged as a result of critique on lasso, which can be unstable due to its variable selection being too dependent on data. The solution is to combine the penalties of ridge regression and lasso to get the best of both.

1. Comparing the Models:

Linear Regression:

- Error: 26381349846757300.0
- This model performed badly due to the absence of a regularization term.

Lasso Regression:

- Error: 817.28812
- This model performed a lot better than linear due to the presence of the regularization term alpha (which was given the value 0.005) however it still performed badly as it tends to perform well when only a few predictors actually influence the output

Ridge Regression:

- Error: 797.16545
- This model performed slightly better than lasso as it has a different regularization term (which was given the value 0.6)

Elastic Net:

- Error: 793.58041
- This model had the best performance out of the four this is mainly because it combines both penalties of ridge and lasso regression.

1. Feature Analysis:

All the features were used in the four models as we were unable to find dependencies or patterns between the feature in order to determine what features to keep and what to discard.

1. Size of training/testing sets:

We split the train dataset into test and train sets where the Test size was 2760/8760

This size was chosen to follow the distribution pattern of the original train/test sets where 8760 is the number of samples in both the original train and test data sets and 2760 is the number of samples in the original test set.

1. Conclusions:

- The first thing we noticed when viewing the data is the absence of any patterns or relationships between the features but after analyzing the data we found a few dependencies such as dew point temperature being always less than or equal to the air temperature.
- As a result of the previous point, we were unable to determine which features to keep and which features to discard causing the accuracy to be not as good as it should be.
- Which in the end caused our models to be unable to generalize (while training ridge had an error of about 409 and after submitting the error almost doubled and became 797)

Milestone Two:

2. Preprocessing:

After loading the data and viewing it:

1. We started with filling the null values using mode (which fills null values in each column with the most frequent values in this column) for the entire dataset.
2. Then we created new dataframes for both train and test containing all the data in the original dataframes and dropped the duplicates.
3. We performed label encoding for the columns 'Spending_Score', 'Var_1'and 'Segmentation'.
4. We performed get_dummies on 'Gender', 'Ever_Married', 'Graduated' and 'Profession' as label encoding was not possible due to these features not being ordinal.
5. Drop the columns for which we created dummies.
6. We concatenated the encoded dataframes with the train dataframe then we also performed the same step for the test dataframe.
7. We dropped the index column for both train and test because it was a useless bi-product of the concatenate process.
8. We shuffled both the train and test dataset.

9. We created a new dataframe called x_train and another called y_train and transefered the train dataset to x_train without the column 'Segmentation' and 'ID' and then put the column 'segmentation' in the y_train.
10. We created a new dataframe for each of the seven models, each included the 'ID' column
11. We graphed the data using q-q plot and found that the data doesn't follow the Gaussian distribution
12. We then applied normalization using 'MinMaxScaler' but we quickly noticed that the accuracy was not as good as we hoped, so we switched to standardization using 'Standard Scaler'and the accuracy gets much better, as shown in naïve bayes, SVM and Logistic regression models.

2. Analyzing the dataset:

After viewing and analyzing the dataset carefully, we were unable to find dependencies or patterns between the features.

2. Classification Techniques:

- **Random Forrest:** It belongs to the supervised learning technique of machine learning and it can be used for both classification and regression. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. It consists of a number of decision trees, each of them gets the average of a subset of the dataset to improve the predictive accuracy of this dataset. The more decision trees used, the better the accuracy and it also makes the possibility of overfitting.

- **Ada-boost:** is one of ensemble boosting classifier which combines multiple classifiers to increase the accuracy. The basic concept behind Ada-boost is to set the weights of the classifiers and then train the data in each iteration to ensure accurate predictions of unusual observations while trying to provide an excellent fit for these examples by minimizing training error. Any machine learning algorithm can be used as base classifier if it accepts weights on the training set.

- **Decision Tree:** it belongs to supervised learning algorithms but unlike other supervised learning algorithms, it can be used for both regression and classification. The goal is to create a model that can be used to predict the class by learning simple decision rules derived from the training data by starting from the root of the tree, We compare the values of the root with the sample and based on the result of this comparison, we follow the branch corresponding to that value and jump to the next node.

- **Naïve Bayes:** are a collection of classification algorithms all based on the same principle that every pair of features being classified is independent of each other. It performs well in multi class prediction and it also performs better when compared to other models of the assumption of the features being independence holds true which can be unrealistic as it's almost impossible that we get a set of predictors which are completely independent.

- **K-nearest neighbour:** is an easy and versatile and one of the top machine learning algorithms. It's non-parametric which means there is no assumption for underlying data distribution making the dataset the determining factor of model structure. It's also lazy meaning it does not need any training data points for model generation and all the training data is used in the testing phase making training faster and testing slower.

- **Support vector machines:** it's known as the discriminative classifier as it separates data points using a hyperplane with the largest amount of margin. It can be used for both classification and regression. In case of non-linear and inseparable planes, svm uses the kernel trick to transform the input space to a higher dimensional space in order to be able to separate the classes.

2. Comparing the Models:

Decision Tree:

- Accuracy: 0.38016
- This model performed the worst out of the seven due to the complex relationships among the features as in decision tree there's only two possibilities making it limited to the relationships it can learn.

Random Forest:

- Accuracy: 0.42889
- This model performed worse than the other models for the same reason decision tree did (as it consists of a number of decision trees) but still slightly better than decision tree as it depends on ensemble learning.

KNN:

- Accuracy: 0.42946
- This model performed similar to random forest but in knn the low accuracy is due to the fact that it basically works on the features similarity which is almost non-existent in this dataset.

Naive Bayes:

- Accuracy: 0.43739
- This model performed slightly better than the previous ones due to the fact that naïve bayes works better on large feature sets and big data.

Logistic regression:

- Accuracy: 0.46175

- This model performed better than the previous ones due to the fact that naive Bayes converges quicker but has a higher error than logistic regression and as your training set size grows, you will likely get better results with logistic regression.

SVM:

- Accuracy: 0.47535
- This model was the second runner and performed slightly better than logistic regression because SVM tries to maximize the margin between the closest support vectors whereas logistic regression maximize the posterior class probability.

Ada-boost:

- Accuracy: 0.47705
- This model performed the best out of all seven

2. Feature Analysis:

All the features were used in the seven models as we were unable to find dependencies or patterns between the feature in order to determine what features to keep and what to discard.

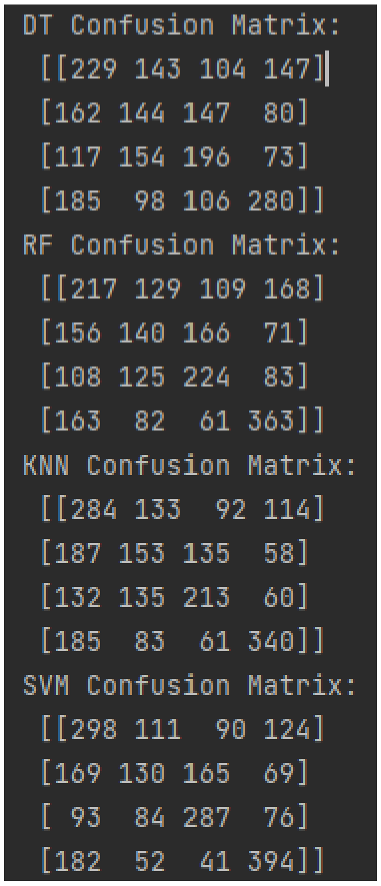
2. Size of training/testing sets:

We split the train dataset into test and train sets where the Test size was 3530/10,695

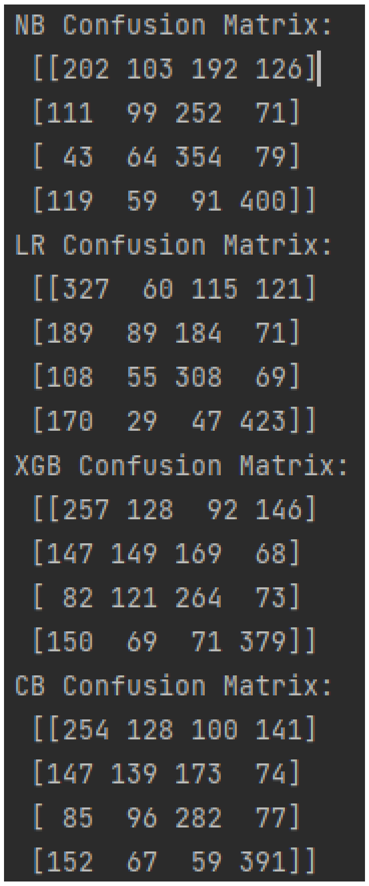
This size was chosen to follow the distribution pattern of the original train/test sets where 10,695 is the number of samples in both the original train and test data sets and 3530 is the number of samples in the original test set.

2. Confusion Matrices:

We split the train dataset into test and train sets where the Test size was 2760/8760



```
DT Confusion Matrix:
 [[229 143 104 147]
 [162 144 147  80]
 [117 154 196  73]
 [185  98 106 280]]
RF Confusion Matrix:
 [[217 129 109 168]
 [156 140 166  71]
 [108 125 224  83]
 [163  82  61 363]]
KNN Confusion Matrix:
 [[284 133  92 114]
 [187 153 135  58]
 [132 135 213  60]
 [185  83  61 340]]
SVM Confusion Matrix:
 [[298 111  90 124]
 [169 130 165  69]
 [ 93  84 287  76]
 [182  52  41 394]]
```



```
NB Confusion Matrix:
 [[202 103 192 126]
 [111  99 252  71]
 [ 43  64 354  79]
 [119  59  91 400]]
LR Confusion Matrix:
 [[327  60 115 121]
 [189  89 184  71]
 [108  55 308  69]
 [170  29  47 423]]
XGB Confusion Matrix:
 [[257 128  92 146]
 [147 149 169  68]
 [ 82 121 264  73]
 [150  69  71 379]]
CB Confusion Matrix:
 [[254 128 100 141]
 [147 139 173  74]
 [ 85  96 282  77]
 [152  67  59 391]]
```

2. Conclusions:

- The first thing we noticed when viewing the data is the absence of any patterns or relationships between the features as a result of the previous point, we were unable to determine which features to keep and which features to discard causing the accuracy to be not as good as it should be.
- After plotting the data using q-q plot, we noticed that it does not follow the Gaussian distribution, so we implemented some models such as naïve bayes , svm, and logistic regression that support working with data that does not follow the Gaussian distribution but the accuracy was still less than 50%, so we tried other models and noticed that even though they work with data that follow Gaussian distribution, they got higher accuracy such as Ada-boost.