Datamining and machine learning CA report

Nathan Cleary

Introduction

For my project I have selected two datasets to evaluate and to make a model that would accurately be able to predict future data.

Divorce Predictors data set available at:
https://archive.ics.uci.edu/ml/datasets/Divorce+Predictors+data+set

Metro Interstate Traffic Volume Data Set:
https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Volume

1 Regression Model

1.1 Business Understanding/Objective

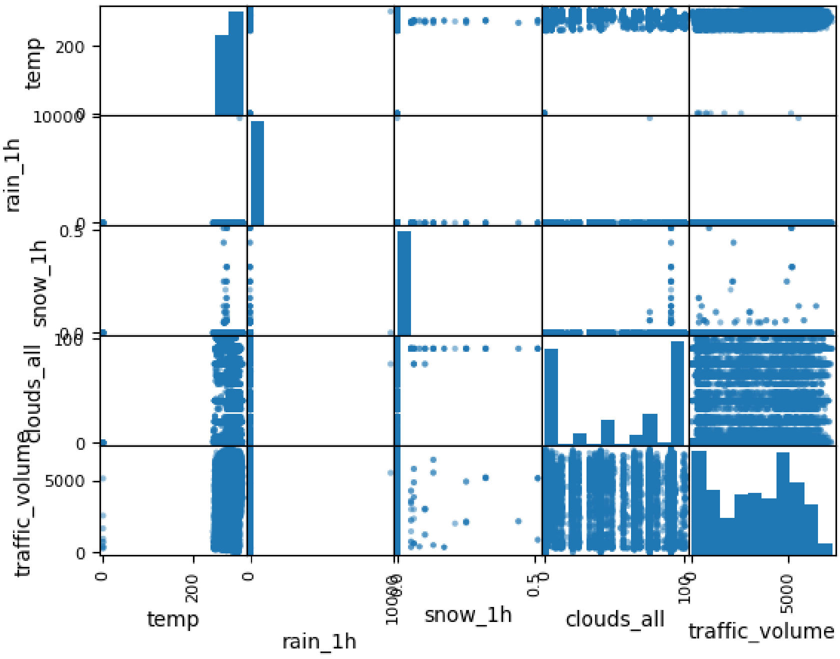
For the first dataset, it consisted of hourly readings on a section of road. These readings included weather information and the traffic volume. The objective of the model is to predict the traffic volume through the section of road. This can be used in the future to help prepare for increased flow of traffic along the road.

1.2 Data Exploration

From exploring the data, we can find information about the dataset that can help with building the model.

- Shape: 48204, 8.
- Instances: 48204,
- Attributes: 8.
- There is no missing data.
- The attributes all have different ranges.
- All attributes have a minimum value of 0.
- Temp maximum value: 310.
- rain maximum value: 9831.
- snow maximum value: 0.51.
- Clouds maximum value: 100.
- Traffic volume maximum value: 7280.
- we can find that the weather was mainly cloudy or clear. Having a more detailed
- description of the weather shows that the most common description was clear with misty and overcast clouds appearing half as of often.
- Snow shower was the least common only appearing once.
- Looking at the holiday, most readings were taken on days with no holiday, with only a couple readings on days of holidays.
- All the holidays have roughly the same frequency.
- The time is in 24-hour time and there are roughly equal values of all times with a maximum difference of 100

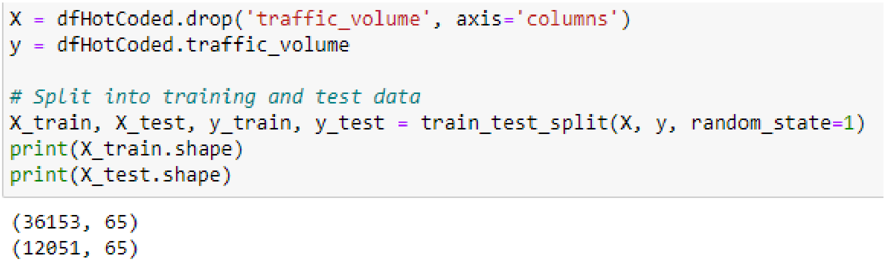
From the scatter plot, there does not seem to be any correlations between the attributes and the traffic volume. Although the scatterplot does not consider the weather main, weather description, holiday and the time attributes so there could still be a connection between those attributes and the traffic volume.



From the above information, there is no clear indication that it will be a good prediction model but there is still uncertainty about some of the attribute's affect to the model.

1.3 Modelling

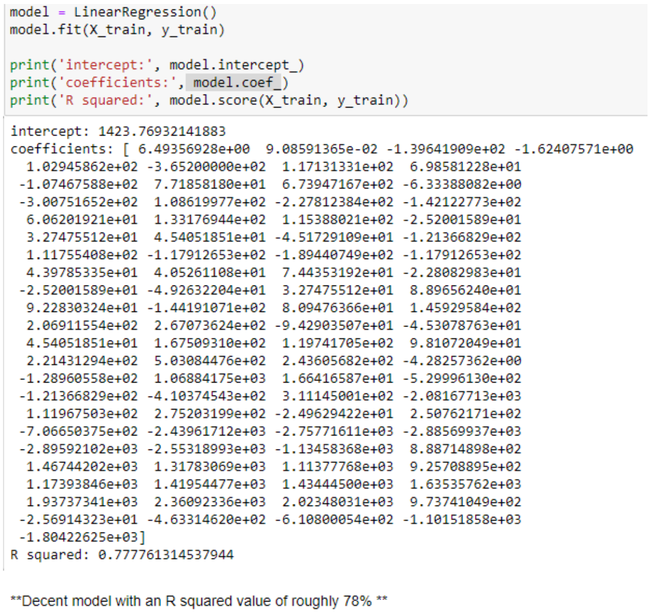
Firstly, the model is built by splitting the data into the parameters and results. The parameters are used to predict the results. The parameters are then split into the training data and the test data. By my model, there is 36,153 instances in the training data and 12,051 instances in the test data.



```
X = dfHotCoded.drop('traffic_volume', axis='columns')
y = dfHotCoded.traffic_volume

# Split into training and test data
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1)
print(X_train.shape)
print(X_test.shape)
```

```
(36153, 65)
(12051, 65)
```

The model is then built using the built-in features of the sklearn package and the training data is fitted to the model.



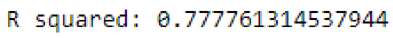
```
model = LinearRegression()
model.fit(X_train, y_train)

print('intercept:', model.intercept_)
print('coefficients:', model.coef_)
print('R squared:', model.score(X_train, y_train))
```

```
intercept: 1423.76932141883
coefficients: [ 6.49356928e+00  9.08591365e-02 -1.39641909e+02 -1.62407571e+00
  1.02945862e+02 -3.65200000e+02  1.17131331e+02  6.98581228e+01
 -1.07467588e+02  7.71858180e+01  6.73947167e+02 -6.33388082e+00
 -3.00751652e+02  1.08619977e+02 -2.27812384e+02 -1.42122773e+02
  6.06201921e+01  1.33176944e+02  1.15388021e+02 -2.52001589e+01
  3.27475512e+01  4.54051851e+01 -4.51729109e+01 -1.21366829e+02
  1.11755408e+02 -1.17912653e+02 -1.89440749e+02 -1.17912653e+02
  4.39785335e+01  4.05261108e+01  7.44353192e+01 -2.28082983e+01
 -2.52001589e+01 -4.92632204e+01  3.27475512e+01  8.89656240e+01
  9.22830324e+01 -1.44191071e+02  8.09476366e+01  1.45929584e+02
  2.06911554e+02  2.67073624e+02 -9.42903507e+01 -4.53078763e+01
  4.54051851e+01  1.67509310e+02  1.19741705e+02  9.81072049e+01
  2.21431294e+02  5.03084476e+02  2.43605682e+02 -4.28257362e+00
 -1.28960558e+02  1.06884175e+03  1.66416587e+01 -5.29996130e+02
 -1.21366829e+02 -4.10374543e+02  3.11145001e+02 -2.08167713e+03
  1.11967503e+02  2.75203199e+02 -2.49629422e+01  2.50762171e+02
 -7.06650375e+02 -2.43961712e+03 -2.75771611e+03 -2.88569937e+03
 -2.89592102e+03 -2.55318993e+03 -1.13458368e+03  8.88714898e+02
  1.46744202e+03  1.31783069e+03  1.11377768e+03  9.25708895e+02
  1.17393846e+03  1.41954477e+03  1.43444500e+03  1.63535762e+03
  1.93737341e+03  2.36092336e+03  2.02348031e+03  9.73741049e+02
 -2.56914323e+01 -4.63314620e+02 -6.10800054e+02 -1.10151858e+03
 -1.80422625e+03]
R squared: 0.777761314537944
```

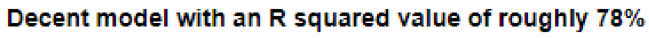
**Decent model with an R squared value of roughly 78% **

1.4 Evaluation

The model has a decent accuracy. The R squared value is roughly 78%.

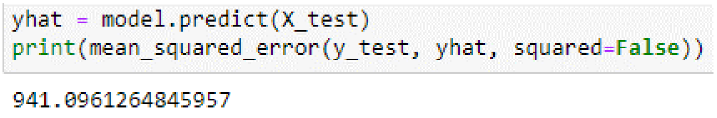


```
R squared: 0.777761314537944
```



Decent model with an R squared value of roughly 78%

The root mean square error was 941.



```
yhat = model.predict(X_test)
print(mean_squared_error(y_test, yhat, squared=False))
```

```
941.0961264845957
```

Based on the above information the model is decent and is somewhat accurate to predict future values.

1.5 Conclusion

This model can be used to predict the volume of traffic through a section of road. Although it should not be used as an absolute prediction of the traffic volume on the section of road it can be used to get a good estimation for how much traffic to expect when given the parameters. This model is also not geographically locked to this specific section of road as the parameters are independent of the location. Therefore, this model should be viable to be used in different locations.

2 Decision Tree

2.1 Business Understanding/Objective

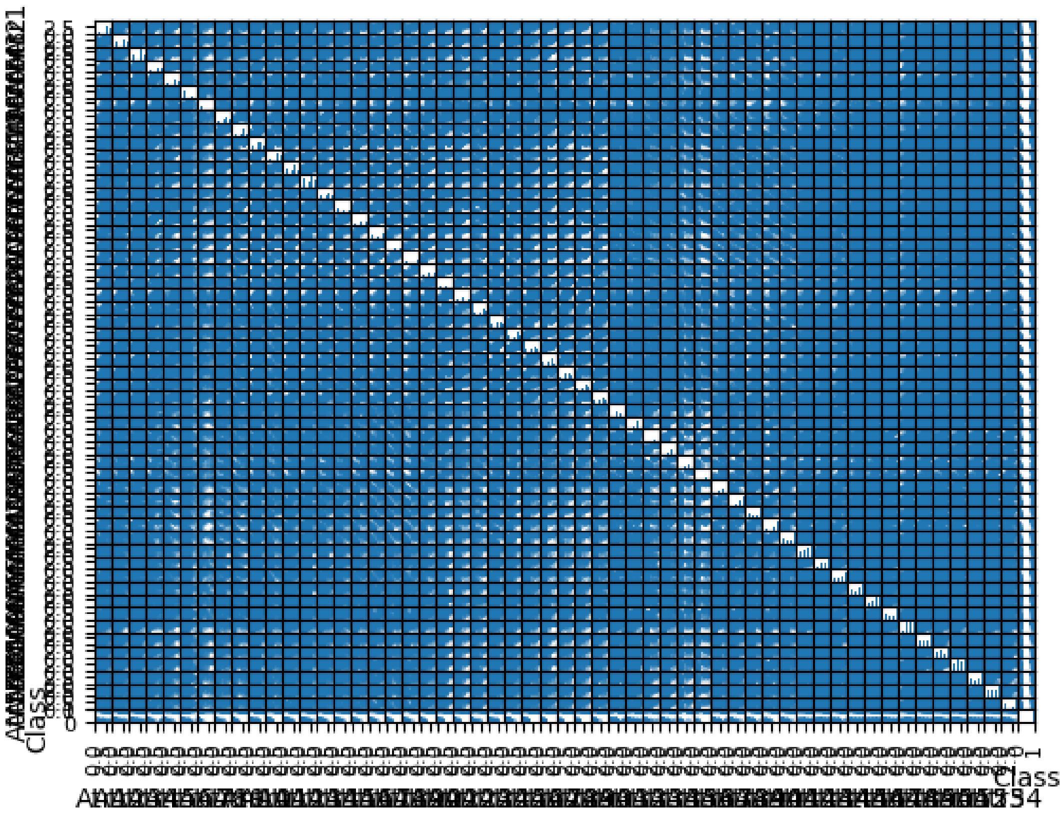
For the second dataset, is a survey consisting of 170 married and divorced individuals. This survey consisted of 54 questions related to their relationship with their partner. I will use this data to predict whether the person is married or divorced. This can be used in the future to help mitigate a long waiting list and waiting times for marriage counselling appointments.

2.2 Data Exploration

From exploring the data, we can find information about the dataset that can help with building the model.

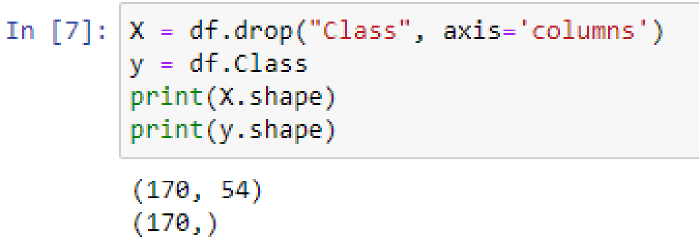
- Shape: 170,55
- Instances: 170
- Attributes: 55.
- There is no data missing.
- All attributes have the same range
- Attributes minimum value: 0
- Attributes maximum value: 4

Scatter Matrix



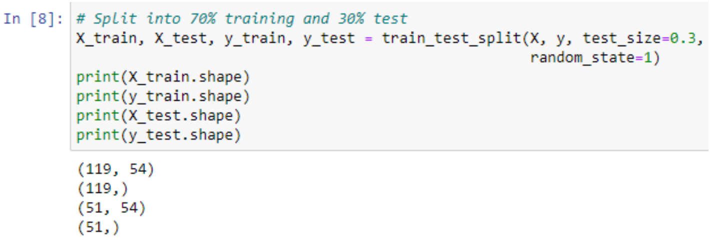
From the scatter matrix is very difficult to tell if the model will be accurate.

2.3 Modelling



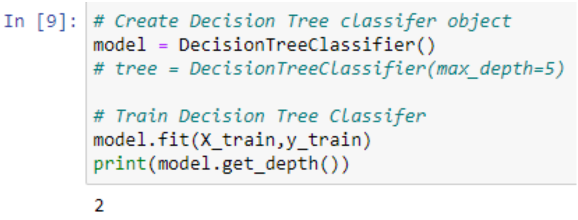
```
In [7]: X = df.drop("Class", axis='columns')
        y = df.Class
        print(X.shape)
        print(y.shape)

        (170, 54)
        (170,)
```

Firstly, the model is built by splitting the data into the parameters data and the results data.



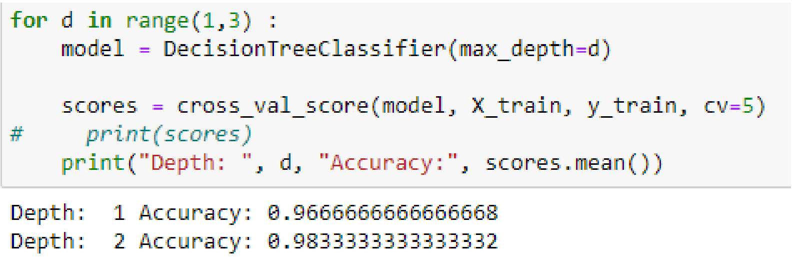
```
In [8]: # Split into 70% training and 30% test
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
                                                            random_state=1)
        print(X_train.shape)
        print(y_train.shape)
        print(X_test.shape)
        print(y_test.shape)

        (119, 54)
        (119,)
        (51, 54)
        (51,)
```

The data is then split again into test data and the training data. For this model we are going to use 70% of the data for training the model and 30% for testing the model to see how accurate it is.



```
In [9]: # Create Decision Tree classifer object
        model = DecisionTreeClassifier()
        # tree = DecisionTreeClassifier(max_depth=5)

        # Train Decision Tree Classifer
        model.fit(X_train,y_train)
        print(model.get_depth())

        2
```

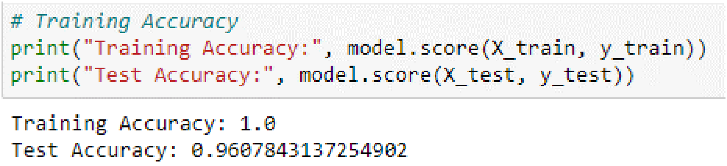
The model is then specified that it is a decision tree and we fit the training data to the model. From running the model, it automatically picks the highest depth which in this case is 2.



```
for d in range(1,3) :
    model = DecisionTreeClassifier(max_depth=d)

    scores = cross_val_score(model, X_train, y_train, cv=5)
#     print(scores)
    print("Depth: ", d, "Accuracy:", scores.mean())

Depth:  1 Accuracy: 0.9666666666666668
Depth:  2 Accuracy: 0.9833333333333332
```

Since the decision tree only has a depth of 2, it is highly unlikely that there is overfitting. But to be sure the model is cross validated using both depths of 1 and 2.

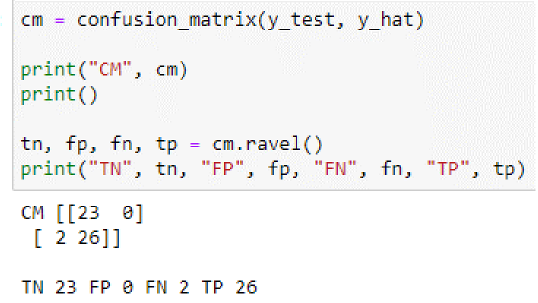
2.4 Evaluation



```
# Training Accuracy
print("Training Accuracy:", model.score(X_train, y_train))
print("Test Accuracy:", model.score(X_test, y_test))
```

```
Training Accuracy: 1.0
Test Accuracy: 0.9607843137254902
```

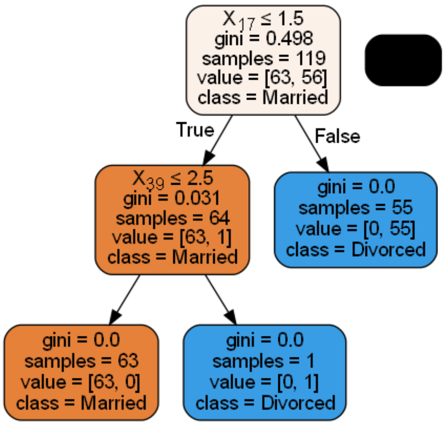
From the evaluation of the model, the test accuracy is 96%. Which makes this a very good model.



```
cm = confusion_matrix(y_test, y_hat)

print("CM", cm)
print()

tn, fp, fn, tp = cm.ravel()
print("TN", tn, "FP", fp, "FN", fn, "TP", tp)
```

```
CM [[23  0]
 [ 2 26]]

TN 23 FP 0 FN 2 TP 26
```

From the confusion matrix, the model correctly predicted all 26 positives. 23 were correctly identified as negatives and only 2 were incorrectly predicted as negatives. In this case positives are married, and negatives are divorced.



In the decision tree graph, we can see where the model makes the cuts in the data. Theses are on the 17th question at 1.5 and the 39th question at 2.5. It does successfully classify the married and divorced individuals at depth of 2.

2.5 Conclusion

The model is very accurate giving a test accuracy of 96%. The model can predict whether an individual is married or divorced. This becomes useful in all services in marriage. This will be very useful especially for marriage counselling where they have a huge number of clients and need to prioritise people who are very close to being divorced.