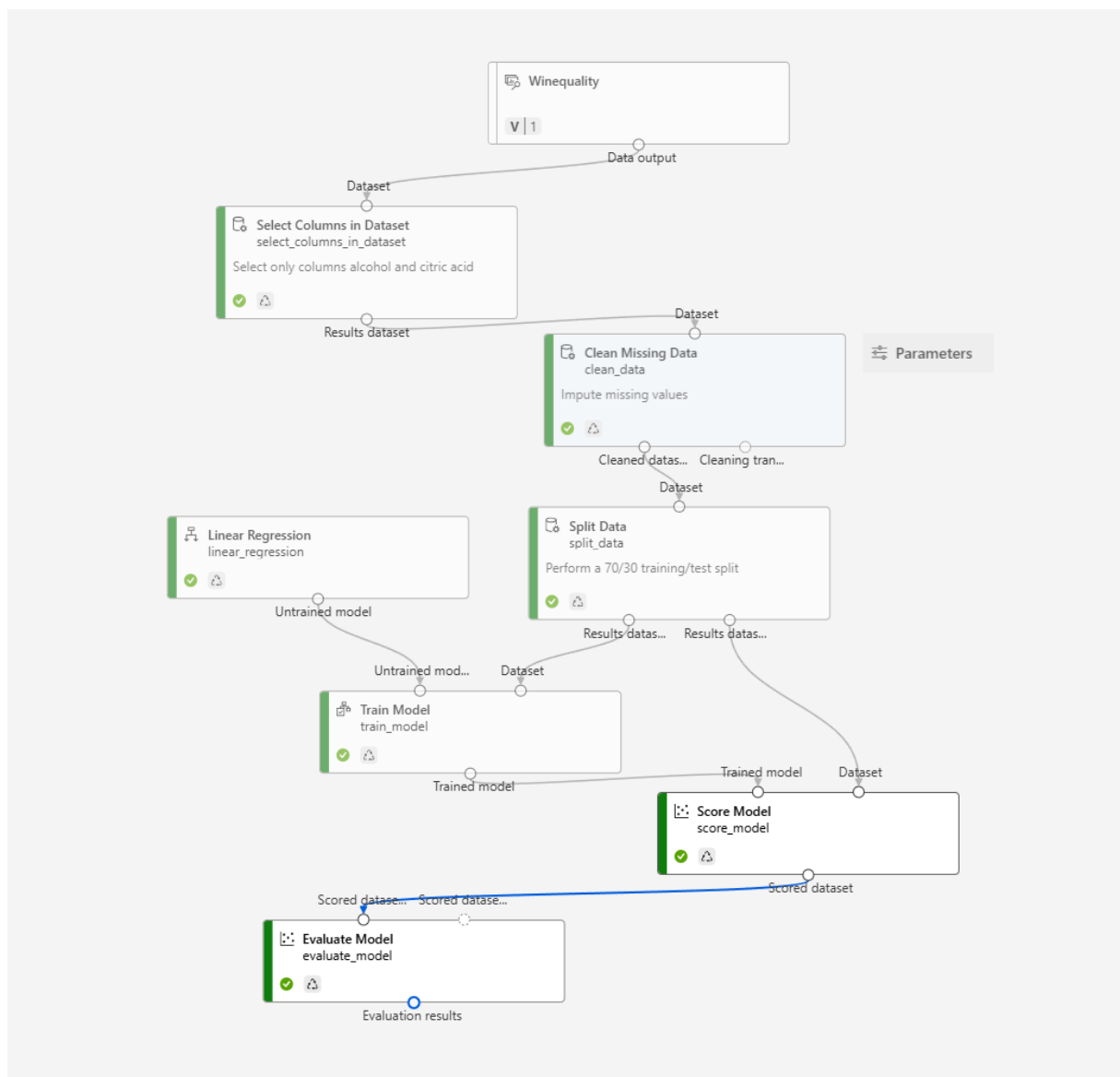
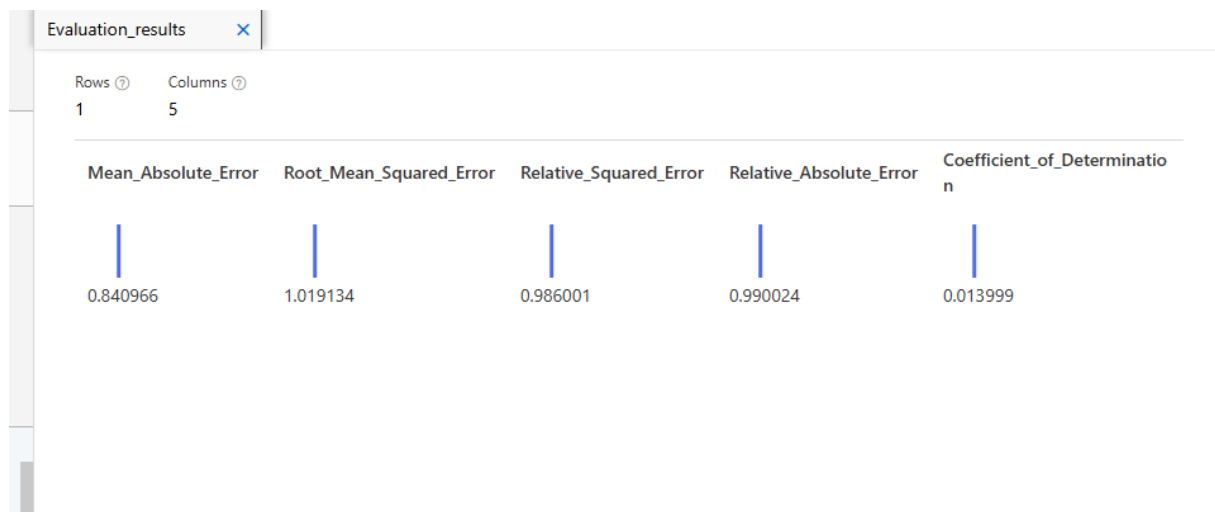


The predictive model which predicts the level of alcohol using citric acid:

1. I downloaded the wine csv file into Azure Machine Learning Studio (Workspace, Assets and Data). The csv file data is on the top of the Designer.
2. Select Column in Dataset: I selected only the columns alcohol and citric acid.
3. Clean Missing Data: Cleaned the data using cleaning mode: Custom substitution value only on columns alcohol and citric acid.
4. Split Data to perform a 70/30 training/test split.
5. Used Linear Regression as Machine Learning Algorithms
6. Trained the model by connecting the pipeline between Train Model, Linear Regression and Split Data
7. The Score Model to generate predictions from the trained model using the dataset as inputs to predict the values along with the actual values.
8. Evaluate the model to get the test results.



The test results from the pipeline model above:



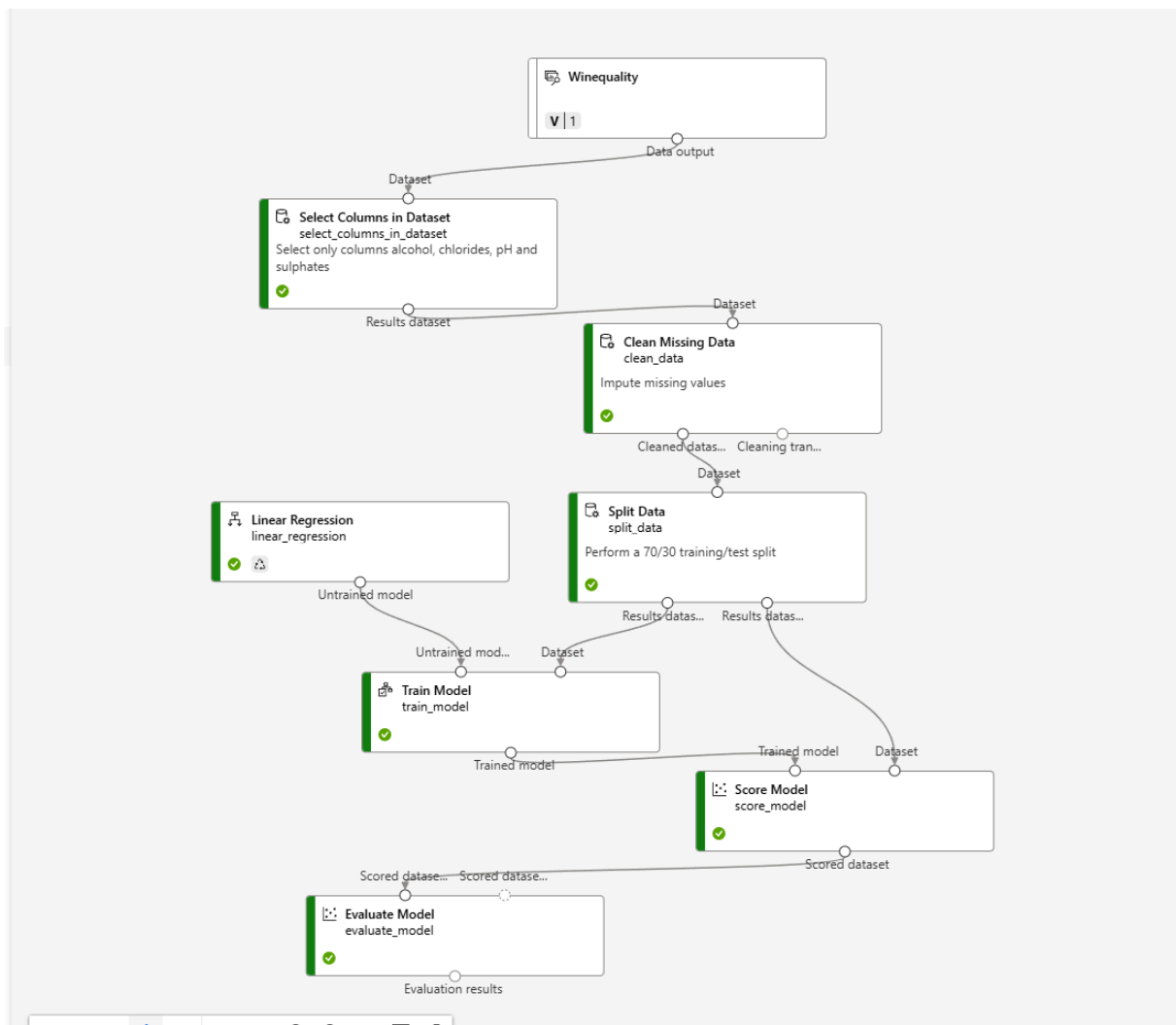
1. Mean Absolute Error (MAE): The models average error when predicting alcohol levels is around 0.84. A lower MAE indicates a better predictive performance.
2. Root Mean Squared Error (RMSE): The error is around 1.02 units of alcohol. RMSE is more sensitive to outlier than MAE.
3. Relative Squared Error (RSE): A value of around 0.99 indicates that the model preforms a little better than predicting the mean alcohol level.
4. Relative Absolute Error (RAE): A value of 0.99 could mean that the model performs almost the same as the mean, just a little better.
5. Coefficient of Determination: The model only explains 1.4 % of the variance in alcohol level. The value is very low.

Conclusion:

The model does not perform well. The relationship between citric acid and alcohol is weak and non-linear.

The predictive model which predicts the level of alcohol using chlorides, pH and sulphates:

1. I downloaded the wine csv file into Azure Machine Learning Studio (Workspace, Assets and Data). The csv file data is on the top of the Designer.
2. Select Column in Dataset: To predict level of alcohol, I added the columns chlorides, pH and sulphates.
3. Clean Missing Data: Cleaned the data using cleaning mode: Custom substitution value only on columns alcohol and citric acid.
4. Split Data to perform a 70/30 training/test split.
5. Used Linear Regression as Machine Learning Algorithms even though the model type is a multiple linear regression.
6. Trained the model by connecting the pipeline between Train Model, Linear Regression and Split Data
7. The Score Model to generate predictions from the trained model using the dataset as inputs to predict the values along with the actual values.
8. Evaluate the model to get the test results.
9. The pipeline model is shown below.



The test results from the pipeline model above:

Evaluation_results				
Mean_Absolute_Error	Root_Mean_Squared_Error	Relative_Squared_Error	Relative_Absolute_Error	Coefficient_of_Determination
0.796263	0.977384	0.906869	0.937398	0.093131

1. Mean Absolute Error (MAE): On average, the models prediction are off by 0.796 units of alcohol content.
2. Root Mean Squared Error (RMSE): RMSE of 0.977 could mean that the model is predicting around 0.977 ib average. A lower value in RMSE is better.
3. Relative Squared Error (RSE): A value of around 0.90 explains around 9% of the variance in the alcohol content.
4. Relative Absolute Error (RAE): A value of 0.937 means that the models error is around 93.7 % of what it would have been if I had always predicted the mean alcohol content. Again, a lower RAE is better.
5. Coefficient of Determination: The model only explains 9.31 % of the variation in alcohol level. This value is very weak.

Conclusion:

The model does not perform well. The low Coefficient of Determination and high errors could suggest that the predictors sulphates, chlorides and pH are not strong enough to predict alcohol content effectively. To make this model more efficient, it needs to explore additional columns to improve the perfmance.

A classification model which predicts wheter the wine is of a low or a high quality using logistic regression.

In this model I chose to use Python scripts to predict the last question:

```
#Importing all the necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Loading the dataset
file_path = "c:/Users/k_ekn/Documents/Data Analyst/Eksamen ML/winequality-red.csv"
data = pd.read_csv(file_path)

# Creating binary classification with a new column
data['quality_label'] = data['quality'].apply(lambda x: 'low' if x <= 5 else 'high')

# Select the columns for the model
columns = ['alcohol', 'chlorides', 'pH', 'sulphates', 'density']
X = data[columns]
y = data['quality_label']

# Splitting the data into 70 % training and 30 % testing (test size to 0.3)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)

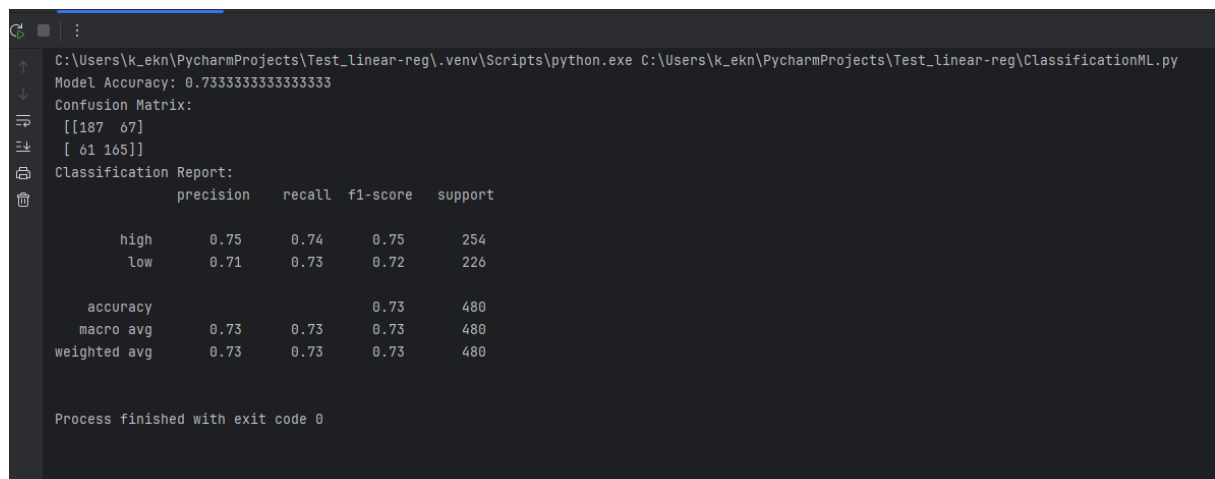
# Creating and training the logistic regression model
```

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

```
# Testing the model and calculating the accuracy, confusion matrix and classification report
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
confusion_matrix = confusion_matrix(y_test, y_pred)
classification_report = classification_report(y_test, y_pred)
```

```
# Print the accuracy of the model
print("Model Accuracy:", accuracy)
print("Confusion Matrix:\n", confusion_matrix)
print("Classification Report:\n", classification_report)
```

The result is below:



```
C:\Users\k_ekn\PycharmProjects\Test_linear-reg\.venv\Scripts\python.exe C:\Users\k_ekn\PycharmProjects\Test_linear-reg\ClassificationML.py
Model Accuracy: 0.7333333333333333
Confusion Matrix:
[[187  67]
 [ 61 165]]
Classification Report:
              precision    recall  f1-score   support

     high       0.75       0.74       0.75         254
     low        0.71       0.73       0.72         226

 accuracy              0.73              0.73         480
 macro avg              0.73              0.73         480
 weighted avg           0.73              0.73         480

Process finished with exit code 0
```

1. Model accuracy shows about 0.733. This could indicate that the model classified about 73.33 % of the sample in the test set.
2. The Confusion Matrix shows that:
 - 187 are true negatives correctly predicted as “low”.
 - 67 are false positives, incorrectly predicts as “high”.
 - 61 are false negatives, incorrectly predicts as “low”.
 - 165 are true positives that correctly predicts as “high”.
3. The classification report:
 - For high quality wines the precision is around 75 %. For the low quality wines, the precision is around 71 %.
 - I am not quite sure what the recall says, but I think it show that if there were 100 actual “high” quality wines, the model correctly identifies 74 of them. The same goes for the “low” quality wines of 73 %, that if there were 100 actual “low” quality wines, it correctly identifies 73 of them.
 - For “high” quality wines the F1-score shows: 75 %, and for the “low” quality wines, the F1-score shows 72 %.
 - Support shows 254 “high” quality wines in the dataset, and 226 “low” quality wines in the dataset.
 - The accuracy shows 73 % of the predictions were right.
 - The macro average and weighted average – I don’t know what that means.

- Overall, the accuracy of the model of 73 % does a decent prediction, but room for improvement.