COMP 4522 - Advanced Databases Assignment 3

Rutu Karavadra, Andrew Walsh, Brandon Kern

Extraction

Describe 2 different ways of extracting the data to a csv from a MySQL database and a MongoDB database:

For MongoDB the first way would be to use the Aggregation Pipeline framework, second would be to use the MongoDB query language or MQL.

For MySQL the first method would be to use a python library like Pandas and the second would be to directly query the database using SQL.

Table / Relation	Attribute	Validation Required	Action
Department_Information	Department_ID	Uniqueness	Report Exception
Department_Information	Department_Name	Uniqueness	Report Exception
Department_Information	DOE	Year >= 1900	Report Exception
Department_Information	All	Missing values	Report Exception
Employee_Infomation	None	None	None
Student_Counceling_Information	Department_Admission	Missing Values	Report Exception
Student_Counceling_Information	Department_Admission	Department_Admission does not exist	Report Exception
Student_Performance_Data	Marks	Range: 0 to 100	Discard entries with issues, and report them
Student_Performance_Data	Hours	Min = 0 Max = any positive integer	Discard entries with issues, and report them
Student_Performance_Data	Student_ID and Paper_ID	A given Student_ID cannot have more than 1 mark per each Paper_ID	Report Exception
Student_Performance_Data	All	Missing values	Discard entries with issues, and report them

Completeness: is the data complete for the predictive data mining task at hand? Do you have null values or out-of-range values that could spoil the quality of your predictions?
2. Validity: here we have several aspects to consider:
Dates must be in range for day, month, and year.
Check for validity related to key values used to link relations.
3. Consistency: check you primary and foreign keys to ensure safe navigation among relations.
4. Uniqueness: you must check for duplicates and for non-existent values for important occurrences. For instance:

Are there any duplicate student IDs?
Are there duplicate Paper IDs?

Department_Information	Department_ID	Uniqueness	Report Exception

Keeping this type of data in the database where there are potential risks of having Department_ID duplicates can cause errors in indexing where two or more departments can be linked to the same ID, causing the confusion of which department does the ID actually reference. This causes ambiguity errors where sometimes the output of a query may be one department and another time, the same query may output a different result. This is especially true if the way it is reaching the ID in a table is not consistently in the same way (i.e. sometimes top to bottom, other times circular). This impacts consistency, which is typically a crucial aspect to have in a database. Deletion anomalies can also occur; for example, if there is one duplicate department_ID remaining after the other is deleted, this will change what references said department_ID. In simpler times, duplications especially in IDs can cause unwanted changes to the data, which then affects the queries and workflows that require this data.

We recognize that this data exception affects the predictive model is….

Department_Information	Department_Name	Uniqueness	Report Exception

Having duplicates of the Department_Name in the Department_Information database can cause certain redundancies in the database, where it can be possible to have two unique department ids pointing to the same thing. These duplicates in naming may also cause certain data to be stored/linked under one department name, and the rest to be stored/linked with the duplicate department name. This can cause a separation in data that should be linked under one identifier but is instead separated and potentially from each other. This significantly decreases the reliability of the data, because it enhances confusion and anomalies, which can then affect the predictive model's output. Having duplicates trains the model to think that there is higher frequency for a certain point, which skews the data graph, and hence lessens the accuracy of the prediction.

Department_Information	DOE	Year >= 1900	Report Exception

Ignoring data that should be checked for an in-range criteria can cause historical, and highly irrelevant discrepancies to exist, especially if considering situations where modern contents are critical. For instance, in this case, having a department that existed beyond any year range, can cause older departments to leak their impact on the database, which does not reflect the true nature of departments in the current scope. This hence impacts the model because it is using irrelevant data points. By implementing a date range check, this ensures that data falls within a

certain timeline, hence creating a limited set of data points. On the other hand, there may be situations where a company would rather have ‘all-data’ for their models, which is what will happen in this case, where we are not setting a date range, simply reporting it here.

Department_Information	All	Missing values	Report Exception

Having missing values in a data table decreases the reliability of the table in and of itself, but also the source. For instance, was missing values in the dataset a human error or is that missing data value simply applicable? Furthermore, the number of missing values in a dataset can highly limit the accuracy of the data set as a whole, because if a lot of values are missing, then that may imply that the data is not trust-worthy. A lot of missing data is highly likely to skew the results of the predictive model too because it may not be an accurate representation of reality. On the other hand, having fewer missing values, especially in a large data set, may be okay to ignore considering its impact is much less and that the remaining data values are enough to cover these missing values. In these cases, it should be okay to simply delete that row of data, but in the former situation, one may have to re-request the data.

Student_Counceling_Information	Department_Admission	Missing Values	Report Exception

Department admission happens to be the foreign key value for the student counseling information table, where each row represents a student who is admitted to a specific department (this is based on the ER-Diagram provided in the assignment). Therefore having missing values of admission means that it cannot match with its associated table, which in this case is the department. Based on the ER-Diagram, the business side seems to have decided that all students must have a department that they are associated with, hence having missing values for a student’s department admission column indicates that this is not a possibility and the student causes an exception to this rule. Students must always be associated with a departament. These missing values create a discrepancy based on admitted students and non-admitted students, which could then lead to data clarity. In a broader sense, it can be okay to have NULL values as foreign keys based on the context that a student could be pending admittance, but once again this is an assumption. In this specific case, while data cleaning, we could potentially remove the row entirely especially considering the low number of missing values in this specific column and table.

Student_Counceling_Information	Department_Admission	Department_Admission does not exist	Report Exception

The lack of validating the ‘existence’ criteria, especially for values that are foreign keys, causes dangling foreign key issues. Having a Department_Admission value that does not exist for a student means that this student is linked to a non-existing table or column. Removal of the table or column followed by improper clean up can cause this situation, which then leads to unreliable data connections and corrupt data too. There is a very easy resolution to this type of error which is to add constraints in your database management system where the table or column cannot be removed if it is being used somewhere else. This prevents the breakage of integrity and consistency by removing data that is needed someplace else. Good documentation highlighting the importance of being a foreign key is equally important to ensure the linkage does not break.

Student_Performance_Data	Student_ID and Paper_ID	A given Student_ID cannot have more than 1 mark per each Paper_ID	Report Exception

This validation is required because in the real-world context, it typically is not possible to have 2+ marks for the same paper. In the digital world of data however, having two marks associated with one paper can easily be designed and implemented, but would adversely affect the data model. From a technical standpoint, it may not recognize an issue, however this would definitely be considered a logical error, affecting data integrity. The way the data is being stored and handled should reflect reality, which is a goal that data cleaning strives to achieve. Hours worked and grade received might also not be properly reflective if two marks are given for the same paper, where one could be an initial hand in and the other would be a refined hand in with fewer hours put in. Not checking for this as validation during data cleaning can skew the predictive model’s results for sure.

Student_Performance_Data	Marks	Range: 0 to 100	Discard entries with issues, and report them

By discarding the data that does not fit into the range of 0 to 100, we set bias into the data that we do not believe that bonuses exist in the world, which is a fair assumption to make especially considering that getting 100%+ is a feat for the few. Another great factor of discarding this data is that it comes close to reflecting the true world marking system, where it is typically impossible to get a negative value on a paper/exam. This enhances the data’s integrity and reliability because it gets rid of outlier as well which would skew the data (ie. there was 999 in the Mark’s column which certainly does not reflect reality and is highly likely to be an incorrect value).

Student_Performance_Data	Hours	Min = 0 Max = any positive integer	Discard entries with issues, and report them

It is not possible to have hours put in as effort that could be showcased as a negative value, therefore once again, keeping this constraint enhances its accuracy in terms of matching with the reality of effort, marks and student performance. This would be considered a logical error, but even though it is technically possible to store a negative value as input under Effort_Hours, it would definitely be seen as erroneous. By checking for these values and discarding them from the dataset, the overall accuracy, reliability and integrity increases, which also means that the predictive model would be more successful at its predictions (assuming everything else is working correctly). It is a bit subjective to say that the max number of Effort_Hours could be any positive integer, especially because infinite could be possible input that is not tangibly true. However, that is more of a business decision that would have to be accounted for and then applied by the technical team. As of now, we have discarded the data values that do not fall within this constraint. We decided to discard them as there were very few of these entries especially considering how big the performance data file is. Removing these instances should logically only have a small impact on the predictive model's results.

Student_Performance_Data	All	Missing values	Discard entries with issues, and report them

Removing missing values from the Student_Performance_Data increases completeness which would enhance the predictive models results ideally, in terms of reliability and accuracy. The reason the cleaning of missing values would enhance the completeness of the data set is because we would have remaining instances in the data that have all of its values in place and hence we can get information out of it in its totality. Having a complete picture of frankly anything in the information one can get from the situation, so the same analogy goes for this complete data set too. It would be a stretch to say that the data set can be 100% complete but it does increase this quality. All in all, this validation provides a stronger comprehensive analysis and also limits the biases that missing values can bring.

Report

This report showcases the findings of the analysis conducted on the various datasets related to department information, employee information, student counseling information and student performance information. This report will share the insights from the data, what cleaning was

performed and the predictive model analysis to analyze the relationship between hours and marks. Please do note, that there is additional info in the jupyter file showcases the specifics of the removed data.

This analysis utilized 4 primary datasets:

1. Department Information (department_df)
2. Employee Information (employee_df)
3. Student Counseling Information (counceling_df)
4. Student Performance Data (performance_df)

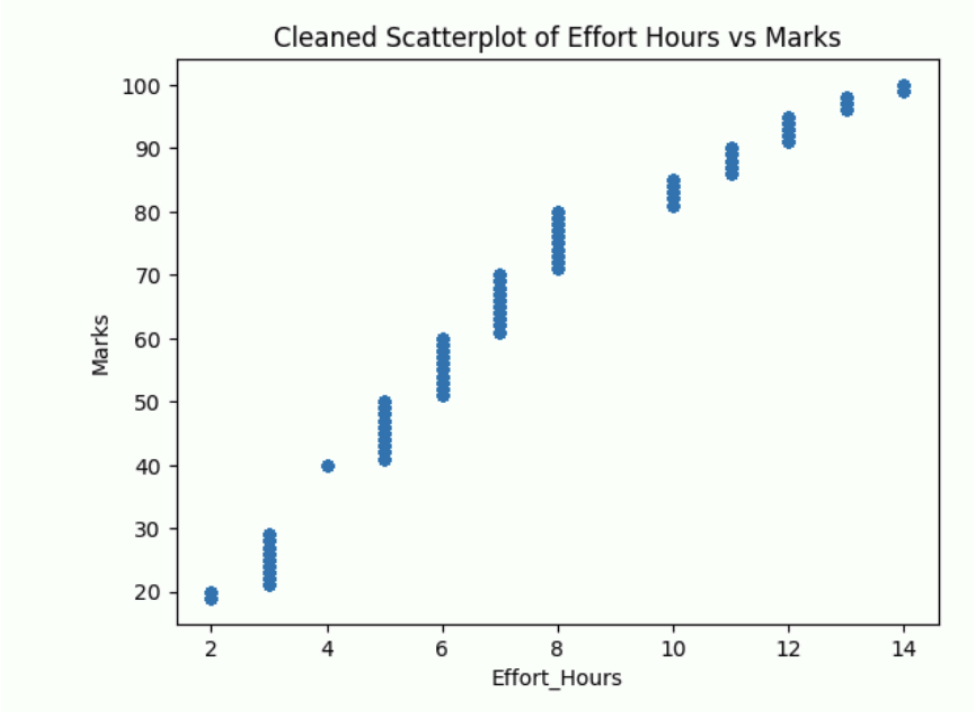
** Regarding Step 6 in the assignment, as noted in the step, it details how effort hours was an added attribute after the data of marks was received, relying on student memory and honesty to get a proper correlation of effort hours to marks earned relation. This could affect the validity and accuracy of the data due to the afterthought. **

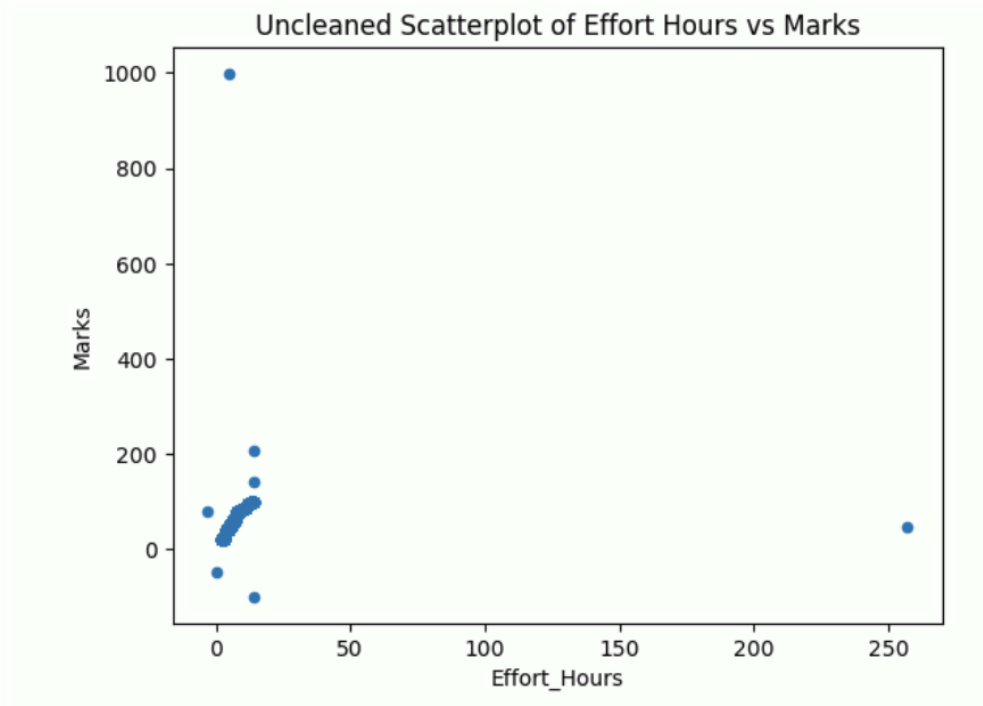
The first step taken was to explore the pure dataset that was given and understand what columns we had and the structure of the units as well. One way we did this was by using the head() feature to get an overview of the data. Also, via the assignment specification, we were told that the data has already been extracted hence we are only reading in the extracted data into the dataframes.

```
Department Information:
  Department_ID                Department_Name         DOE
0     IDEPT4670           Aerospace Engineering   5/31/1961
1     IDEPT5528  Biosciences and Bioengineering   6/28/1943
2     IDEPT3115            Chemical Engineering    5/1/1940
3     IDEPT5881                       Chemistry    6/8/2013
4     IDEPT4938              Civil Engineering  10/27/1941

Employee Information:
  Employee ID         DOB         DOJ Department_ID
0    IU196557   2/23/1983  10/31/2009     IDEPT4938
1    IU449901    9/2/1985    6/7/2009     IDEPT2357
2    IU206427   7/30/1971    5/9/2008     IDEPT4670
3    IU688905   7/20/1973   1/17/2002     IDEPT2601
4    IU634582  11/16/1991   2/13/2000     IDEPT7626

Student Counseling Information:
    Student_ID       DOA        DOB Department_Choices Department_Admission
0  SID20131143  7/1/2013   2/5/1996          IDEPT7783            IDEPT7783
1  SID20131151  7/1/2013  7/31/1995          IDEPT6347            IDEPT6347
2  SID20131171  7/1/2013   9/5/1995          IDEPT1836            IDEPT1836
3  SID20131176  7/1/2013  1/12/1996          IDEPT8473            IDEPT8473
4  SID20131177  7/1/2013  7/30/1995          IDEPT5528            IDEPT5528

Student Performance Data:
    Student_ID Semster_Name     Paper_ID Paper_Name  Marks  Effort_Hours
0  SID20131143        Sem_1  SEMI0012995    Paper 1   44.0           5.0
1  SID20131143        Sem_1  SEMI0015183    Paper 2   74.0           8.0
2  SID20131143        Sem_1  SEMI0018371    Paper 3   80.0           8.0
3  SID20131143        Sem_1  SEMI0015910    Paper 4   44.0           5.0
4  SID20131143        Sem_1  SEMI0016208    Paper 5   95.0          12.0
```

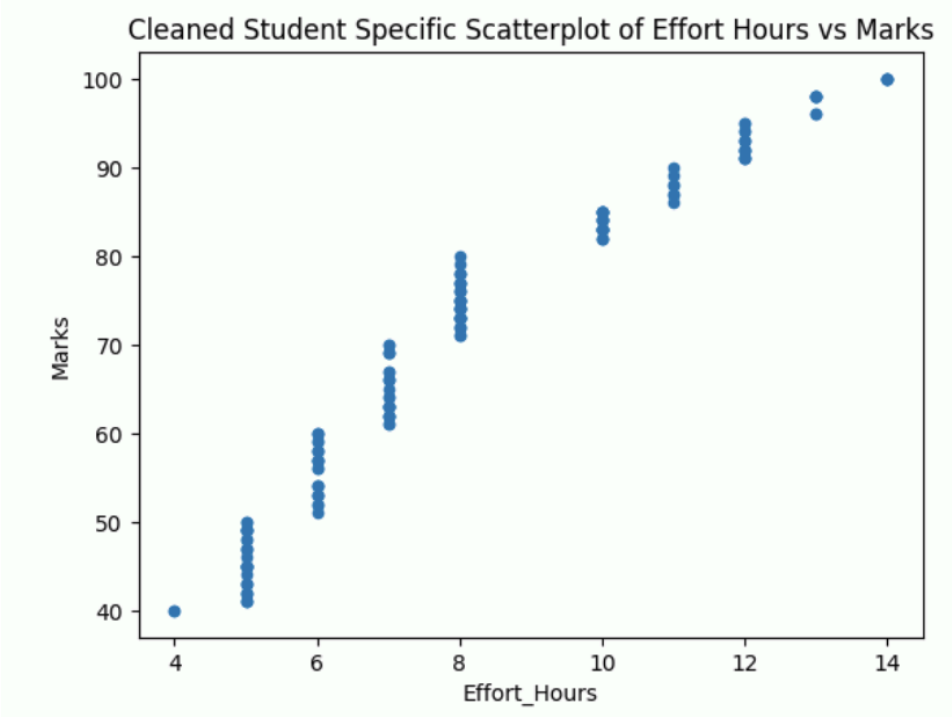
Thereafter, after examining the original dataset, we began the data cleaning. To be honest, in the beginning we misunderstood the assignment where we actually started coding the data cleaning process in Python and Jupyter, but when we had a chat with Orestes, it came to our attention that we are only supposed to code 3 validations that were listed in the table in the assignment and simply report on the others. Therefore, we limited our data cleaning to the following:

- Marks outside of the valid range 0-100 were identified and removed from the dataset. This helped eliminate outliers.
- Missing values in both the Marks and Effort_Hours columns were removed from the dataset. This helped limit the unknowns in the dataset and created a model based on the complete info instead.
- We removed outliers in the Effort_Hours by coming across a outlier that was significantly skewing the visualizations we were doing to better understand the data.
- Effort Hours that had negative values were also removed from the dataset to ensure that this not-possible scenarios were thrown away, limited the falsehood in the data





This is a step by step data cleaning approach, where incremental cleaning was done on the data set and then the more-cleaned data set was provided to the next layer of cleaning. Hence, step by step the data set continually gets cleaner.



We have two ‘cleaned’ datasets where one is called ‘student_specific’ and the other is called ‘cleaned_performance’ / full. The student specific is the data set based on the three provided student ids after cleaning the original dataset. The cleaned performance dataset in the overall data after cleaning, not student specific. Both datasets were fed into the linear regression into 2 separate occasions, but surprisingly the results were fairly similar.

Visualization techniques were used for the cleaned dataset thereafter using the following methods:

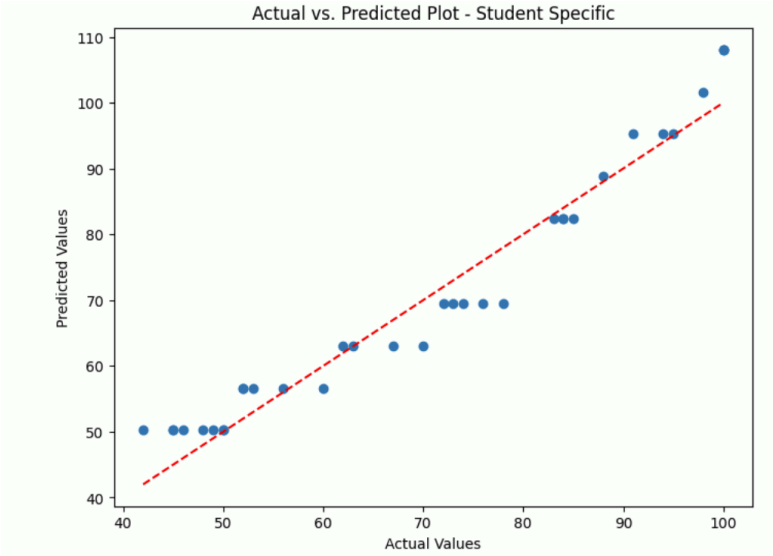
- Scatterplots: This helped visualize the relationship between Effort Hours and Marks and also helped give us a Before and After picture. It also helped to see that the data did follow a linear type of shape for the most part, which indicated that linear regression was a good fit.
- Histograms: This was used to visualize the distribution of Marks, which was fantastic to see that for the most part, every section possible for a mark did have a data point that would help. There was no section that was heavily missing data points indicating that the data was distributed quite evenly, covering vast ground.
- Box Plots: This was used to identify any outliers in the dataset and also helped to see where the middle was in terms of the 75th, 50th and 25th percentile too. This also helped to visualize the extremes including median and interquartile range.

Thereafter we immediately did a correlation analysis to understand the relationship between Effort Hours and Marks, and it was found that there is a strong positive correlation between the two variables, which indicates that the predictive model can rely on assuming marks based on effort hours.

The fact that we went with the predictive model of linear regression was due to it being specified in the assignment specification, however this model was a great choice due to the linear shape of the correlation that incremented in a straight way. The model was trained using the cleaned dataset and evaluated using mean absolute error (MAE), mean square error (MSE) and root mean squared error (RMSE) metrics.

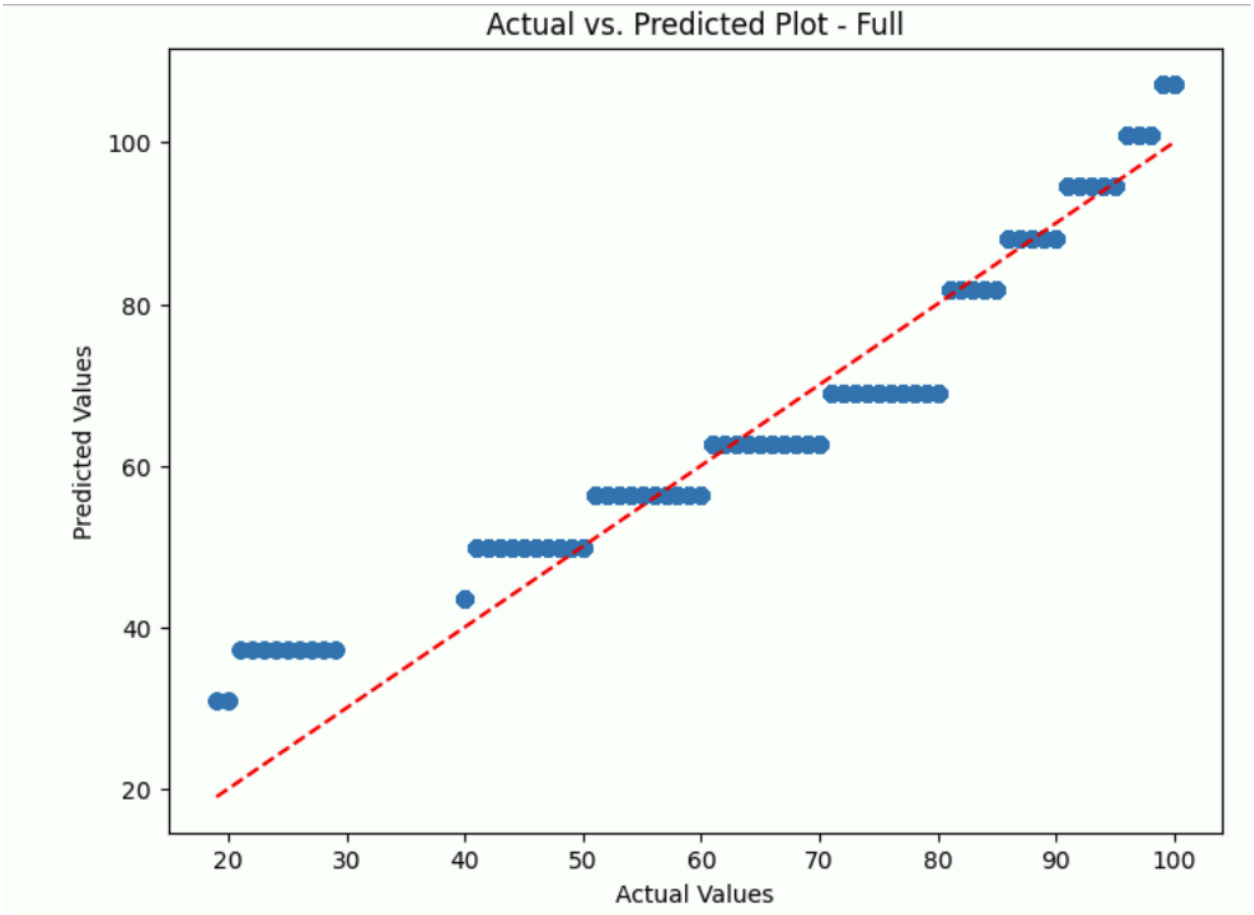
```
### EVALUATING the Model ###
#Taken directly from Orestes Appel's course notes on Linear Regression - COMP 4522 Winter 2024
from sklearn.metrics import mean_absolute_error, mean_squared_error

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse) # Root Mean Squared Error
print(f"mae = {mae:.2f}")
print(f"mse = {mse:.2f}")
print(f"rmse = {rmse:.2f}")

r2 = r2_score(y_test, y_pred)
print("R-squared (R2):", r2)
```

```
mae = 3.59
mse = 19.83
rmse = 4.45
R-squared (R2): 0.9440660552430018
```




```
### EVALUATING the Model ###
#Taken directly from Orestes Appel's course notes on Linear Regression - COMP 4522 Winter 2024
from sklearn.metrics import mean_absolute_error, mean_squared_error

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse) # Root Mean Squared Error
print(f"mae = {mae:.2f}")
print(f"mse = {mse:.2f}")
print(f"rmse = {rmse:.2f}")

r2 = r2_score(y_test, y_pred)
print("R-squared (R2):", r2)
```

```
mae = 3.73
mse = 22.19
rmse = 4.71
R-squared (R2): 0.9324671926544668
```



The following is what was analyzed from these 3 measurements:

- mae = 3.73 means the predictions are off by 3.73 units
- mse = 22.19 adds more weight on larger errors compared to MAE
- rmse = 4.71 which is a lower value, hence indicates good performance!
- R-squared = 0.93 shows the variability in the target variable, and it suggests a good fit for the data!

Overall these 3 evaluation mechanisms indicated that the predictive model was looking good and could be depended on.

Once again we also used visualization to help see the predictive model's reliability:

- Residual Plot helps look into the actual and predicted values against the predicted values. Ideally residuals should be randomly scattered against the horizontal line of y = 0, as that would indicate unbiased predictions. That is what our visualization does show with limited outliers and clusters, hence a good indication that the performance and assumptions of our linear regressor is decent.

- Actual VS Predicted plot helped visualize how deviated the actual points are from the predictions, and it was a pleasure to see that the deviations were relatively very very small. I would even assume that the deviations are more human related, rather than a reflection of the predictive model. For instance, a dehydrated student who puts in effort of 10 hours may get 79% compared to a healthy, hydrated individual who would get 82%. It is fair to say that because this analysis is so human-related, deviations are bound to happen, but once again I am glad that it was small.
- A heatmap helps with seeing which attributes in the analysis are more closely related with each other, but in this case, because we were only working with two columns 0 Effort Hours and Marks - this heat map feature did reveal much apart from the fact that we can say that the number of effort hours a student puts does have a high correlation with the mark that they will get. This heatmap correlation feature may be more useful for datasets and analysis that have more attributes and connections.

Using multiple students' data onto a singular data effort hours would skew the results of that singular student in order to achieve a proper model for a single student, you would need to build a model on that specific student. For example, for SID20131151, 2 hours of effort resulted in only a mark of 19 whereas for SID20182516, 12 hours of effort resulted in a mark of 92. The marks between the higher students are quite large, which is why the predictive model estimates that approximately 30 should be given as a mark for students who put in 2 hours, when in actuality the extracted database shows only a mark of 19 for the former student.

Since we used two datasets where one was student specific and the other had the full dataset (note both were cleaned), it was interesting to see that despite the mass difference between the entity instances, the overarching shape of the visualizations resembled one another. Also, the model evaluations where we calculated the mae, mse, rmse and r2 were similar as well, indicating a model that was strikingly alike.

Below is a clear example of how for effort hours of 10, the student specific predicted a mark of 82.37 and the full dataset predicted a 81.79%.
Student Specific:

```
#Taken directly from Orestes Appel's course notes on Linear Regression - COMP 4522 Winter 2024
# Student Specific
score = regressor.predict([[10]]) # We are passing 10 in double brackets to have a 2-dimensional array
print(score)
```

```
[[82.37295137]]
```

Full/Cleaned Dataset:

```
[311]: #Taken directly from Orestes Appel's course notes on Linear Regression - COMP 4522 Winter 2024
score = regressor.predict([[10]]) # We are passing 10 in double brackets to have a 2-dimensional array
print(score)
```

```
[[81.79639027]]
```

All in all, based on the analysis of the predictive model, I would say that it is trustworthy. I would say that the analysis showed that there is a strong relation between student's effort hours and marks, and the predictive model of a linear regression demonstrated promising results in predicting what those marks could be based on the effort hours given. Further refinement and evaluations would be helpful especially in data cleaning and data mining. In terms of future outlook some areas that could be looked into further are what other attributes may help in determining the marks of a student other than effort hours. Refinement of the predictive model would also be a place for improvement too.

6. Take note that the Professors members of the Student Council, whom are in control of assigning funding, asked all students to provide the number of hours they spent in each paper that they did write. The Council requested to the Database Department that the (new) information, related to Effort (hours invested in each paper), were added to the Database. As the request was urgent and everyone was busy, the Database Administrator (DBA), added an attribute (column) to the "Student Performance Data" table, as the last attribute to the right. Hence, now the aforementioned relation looks like this:

Student_ID	Semester_Name	Paper_ID	Paper_Name	Marks	Effort_Hours
SID12345678	Sem_x	SEMI1234567	Paperx	XX	XX

Note: these activities are very common in the industry and are usually grouped together as ETL or ELT, combined with data warehousing and data mining. Data quality is an underlying layer to all the above. Think about the implications.