Final Project

Frank Asto

Intro to Data Science (DS 210)

Introduction and Stating the Question

Can the outside temperature be estimated by the frequency of cricket chirps? That's a fascinating question! Let's embark on the data science life cycle to uncover the answer. To start, I have a dataset that will help me address this inquiry. The dataset comprises two features: 'Chirp15s' and 'TempFarenheight', containing 59 records. In my opinion, this dataset should be enough to make an interesting analysis to address this particular question. However, a quick scan revealed the presence of 2 null values. This emphasizes the need to thoroughly revise my data before proceeding with any analysis.

Exploratory Data Analysis

To ensure the integrity of my data before Exploratory Data Analysis, I will utilize Python/Pandas statements to clean it, identifying and addressing any potential errors within the dataset. I remember encountering a null value in the 'Chirps15s' column when I was doing a quick scan. To ensure I know how many null values I have, I will use the following statement:

```
print(sum(df['Chirps15s'].isnull()))
```

This statement counts all the null values I have in the 'Chirps15s' feature. After I ran this function, it returned one null value. To make things simpler, I will replace the null value with the mean (average) function in the dataset. For that, I will use the following Python/Pandas statement:

```
df.Chirps15s = df.Chirps15s.fillna(df.Chirps15s.mean())
```

By utilizing this method, I find it to be a more effective approach compared to simply deleting the null value. To verify the success of this operation, I will once again execute the "isnull()" function. Effectively, it reveals zero null values.

Moreover, I want to identify if there are any outliers in the dataset. For that, I will use the following statement:

```
print(df.sort_values('Chirps15s'))
```

This statement will sort the numerical values on the 'Chirps15s' feature, aiding in a more thorough inspection of the data. I chose this approach because outliers tend to manifest at the extreme ends. While browsing through my data I noticed an obvious outlier: '361'. Since I had identified only one outlier in the 'Chirps15s' feature, I will opt for deleting the entire record. To do this in Python/Pandas I will use the following statement:

```
df = df[df.Chirps15s < 47]
```

This statement filters the dataset based on a specific condition. It selects only the rows where the value in the 'Chirps15s' column is less than '47'. By doing so, it removes any values that are equal to or greater than '47', essentially eliminating the outlier '361' from the dataset. I chose number '47' because the next number to '361' is '46.4' in the dataset. When I ran the sort function again, the outlier '361' was no longer there.

Furthermore, I will now perform a similar approach on the 'TempFarenheight' feature. I will look for any null values in the dataset. To start, I will perform the following statement:

```
print(sum(df['TempFarenheight'].isnull()))
```

Again, this statement counts all the null values I have in the 'TempFarenheight' feature. After I ran this function, it returned one null value. As I did before, I will replace the null value with the mean (average) function in the dataset. For that, I will use the following Python/Pandas statement:

```
df.TempFarenheight = df.TempFarenheight.fillna(df.TempFarenheight.mean())
```

Once again, to verify that the statement worked, I will execute the "isnull()" function. Now, it shows zero null values.

Additionally, I want to know if there are any outliers in the dataset. To accomplish this, I will utilize the following Python/Pandas statement:

```
print(df.sort_values('TempFarenheight'))
```

Once again, this function will arrange the numerical values in order in the ‘TempFarenheight’ column, facilitating the examination of the data. While reviewing the dataset, I identified an obvious outlier: ‘6’. Therefore, I have decided to remove the entire record associated with it. In Python using Pandas, I'll employ the following statement:

```
df = df[df.TempFarenheight > 48]
```

This is a similar statement condition that I did with the ‘Chirps15s’ column. It selects only the rows where the value in the 'TempFarenheight' column is greater than ’48’. By doing so, it removes any values that are equal to or less than ‘48’, essentially eliminating the outlier ‘6’ from the dataset. I chose the number ‘48’ because the next number to ‘6’ is ’49.25’ in the dataset. When I ran the sort function again, the outlier ‘6’ was no longer there. Now that the dataset is clean, I am ready to do some Exploratory Data Analysis.

To begin, I will find the mean and median for both 'Chirps15s' and 'TempFarenheight'. Let's start by calculating the mean for 'Chirps15s' and 'TempFarenheight' with the following Python/Pandas statements:

```
print(df['Chirps15s'].mean())
print(df['TempFarenheight'].mean())
```

This function will return the mean value of 28.93 for the ‘Chirps15s’ feature and the mean value of 65.72 for the ‘TempFarenheight’ feature. Now, I will calculate the median for both features as well using the following statements:

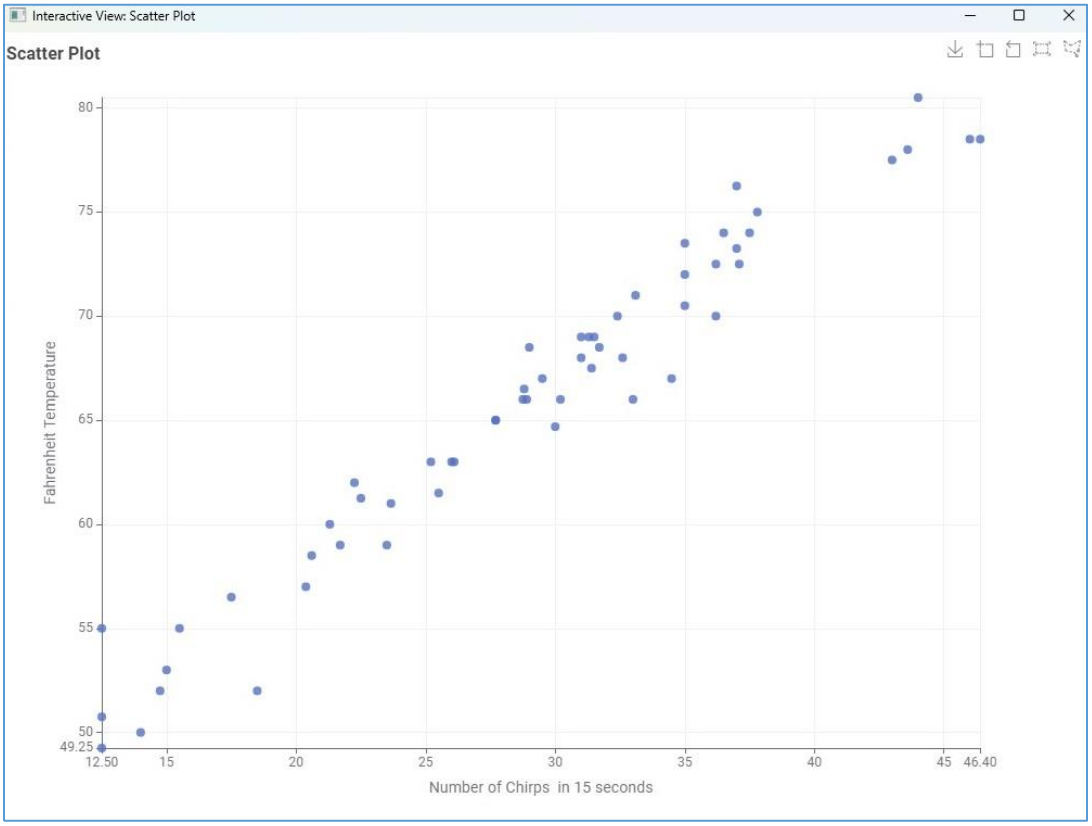
```
print(df['Chirps15s'].median())
print(df['TempFarenheight'].median())
```

This function will output the median values of 30.0 for 'Chirps15s' and 66.5 for 'TempFarenheight', respectively.

With my data now cleaned and prepared, I will proceed to save it. This will allow me to continue analyzing my dataset in KNIME without any disruptions. To save my dataset to a new dataset I will use the following statement:

```
df.to_csv('CleanDataProject.csv')
```

I will conduct further exploratory analysis, this time leveraging the capabilities of KNIME, a powerful analytical tool. Using my refined dataset, I will create a scatter plot to visually depict any noticeable patterns. KNIME simplifies analytic procedures through its node-based architecture, where each node serves a specific function. Beginning with the CSV Reader node to import my dataset, I will then connect it to the Scatter Plot node. This visualization will provide an intuitive representation of my data, helping me identify any underlying trends or relationships.

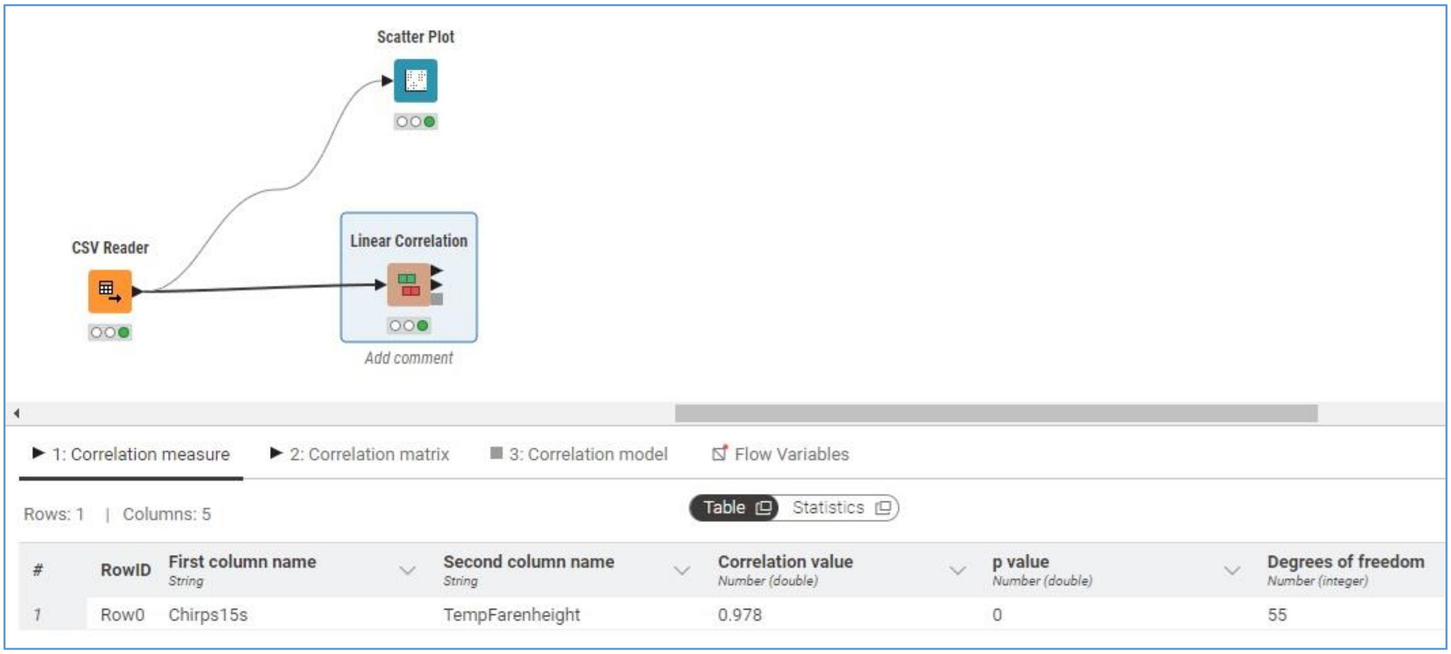


Refining the Question

Now that we have completed our Exploratory Data Analysis, let's dive in and examine our findings using the Scatter Plot visualization. Let's start by addressing the initial question: Can the outside temperature be estimated by the frequency of cricket chirps? Upon examining the visualization, a strong linear correlation between the two features becomes evident. As the number of chirps increases, so does the temperature. However, for complete assurance, I will conduct a calculation in KNIME to determine the Correlation Coefficient. Utilizing the Linear Correlation node, I obtain a Correlation value of 0.978. With this value nearing 1, it indicates a strong positive correlation. Thus, it corroborates my initial analysis, confirming the validity of this particular conclusion. Refining the question is an essential step in the Data Science Life Cycle as it ensures the accuracy of our analysis. It allows us to confirm our progress so far; otherwise, we might get stuck when moving on to the next phase of Model Building. There may be times when we need to adjust the original question to match our findings.

Model Building

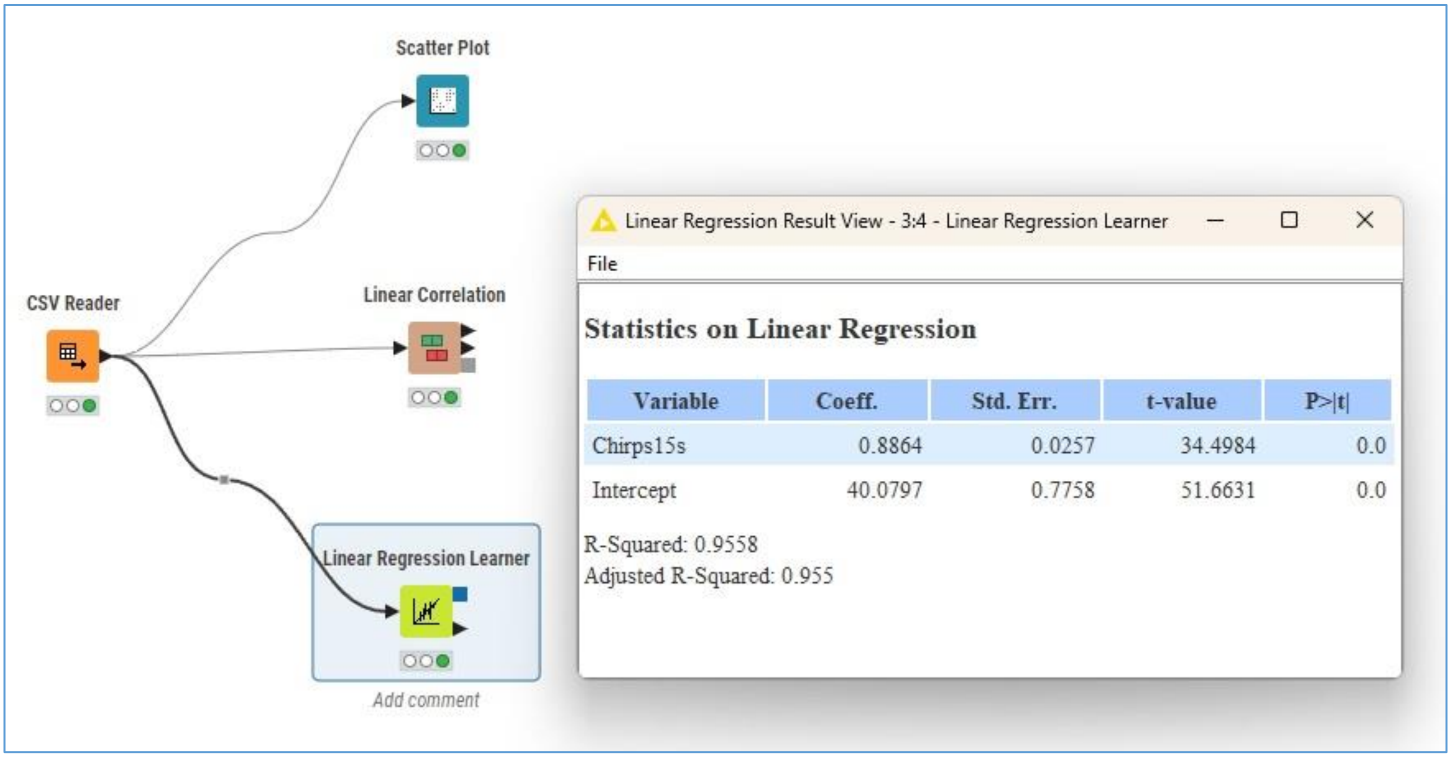
Now, we reach the exciting phase: Model Building. This step holds a special place in the Data Science Life Cycle for me because, with the insights from our exploratory data analysis, we can now begin to create a powerful model. This model will not only help us predict but also deepen our understanding of the analysis we are working on. To start, I'll utilize KNIME to construct a linear regression model for predicting temperature based on the number of cricket chirps heard in 15 seconds. This model falls under supervised learning as it involves using one feature to predict the value of another feature. Before building the model, let’s check again the Correlation Coefficient in the following picture:



Rows: 1 | Columns: 5

#	RowID	First column name String	Second column name String	Correlation value Number (double)	p value Number (double)	Degrees of freedom Number (integer)
1	Row0	Chirps15s	TempFarenheight	0.978	0	55

In this image, the Correlation value of 0.978 is visible, confirming a strong linear relationship between 'Chirps15s' and 'TempFarenheight'. Now, I will begin constructing the model by utilizing the Linear Regression Learner node in KNIME. Within this node, under configurations, I will designate 'Chirps15s' as my independent variable and 'TempFarenheight' as my target or dependent variable. This setup allows me to predict the temperature based on given values of 'Chirps15s'. Upon completion, I will obtain statistical insights on Linear Regression, as illustrated in the following picture:


Scatter Plot
CSV Reader
Linear Correlation
Linear Regression Learner
Add comment
Linear Regression Result View - 3:4 - Linear Regression Learner
File

Statistics on Linear Regression

Variable	Coeff.	Std. Err.	t-value	P>|t|
Chirps15s	0.8864	0.0257	34.4984	0.0
Intercept	40.0797	0.7758	51.6631	0.0

R-Squared: 0.9558
Adjusted R-Squared: 0.955

Now that I have these new calculations, I can start building our model. Let's begin by explaining what a regression line is and the equation we will use for the one we are creating. A regression line is a straight line that shows the relationship between two variables in a dataset. It helps predict the value of one variable (dependent variable) based on the value of another variable (independent variable). The equation for the regression line is the following:

$$\mathbf{y = mx + b}$$

'**y**' represents the dependent variable, which in this case is 'TempFarenheight'. '**x**' represents the independent variable, which corresponds to 'Chirps15s'. '**m**' stands for the slope, represented by the number 0.8864, and '**b**' represents the Intercept, which is the number 40.0797 as shown. So, using these two known numbers, we can form the following regression line equation:

$$\mathbf{y = 0.8864(x) + 40.0797}$$

Now, let's make some predictions! Let's consider a scenario where we want to determine the temperature on a night when you hear 40 cricket chirps in 15 seconds. Remember, 'Chirps15s' is our independent variable, so we can plug it into our equation as follows:

$$\mathbf{y = 0.8864(40) + 40.0797}$$

By solving this equation, we find the value to be 75.5. Therefore, if you hear 40 cricket chirps in 15 seconds, you can predict that the outside temperature could be approximately 75.5 degrees. This is very interesting, and now you can use this equation to predict the outside temperature by plugging in any number of 'x', or in this case, any number of cricket chirps in 15 seconds.

Interpretation/Summary

In conclusion, this journey has been remarkably compelling as I navigated through the data science life cycle to solve the mystery behind a simple yet intriguing question: "Can the outside temperature be estimated by the frequency of cricket chirps?" It all began with a dataset handed to me, prompting me to embark on a quest for answers.

First, I carefully cleaned the data, addressing missing values and outliers using Python/Pandas, ensuring the integrity of my analysis. Then, armed with insights from Exploratory Data Analysis using scatter plot in KNIME, I uncovered a compelling story: a strong positive correlation between the frequency of cricket chirps and outdoor temperature.

To confirm my findings, I calculated the correlation coefficient, affirming the strong relationship between the features. With this confirmation in hand, I ventured into model building using KNIME, leveraging its power to derive the ‘slope’ and ‘intercept’ necessary to construct a predictive equation. With all elements arranged, I confidently presented an effective equation, culminating in a temperature prediction for a scenario where cricket chirps per second reach 40. This project has been an exciting journey of discovery, showcasing the potential ability of data science to uncover nature's mysteries.