PYTHON MACHINE LEARNING LABS GROUP PROJECT

PREDICTING SLEEP VARIABLES IN MAMMALS

Group members:

Mohammed Danish Mustafa

(https://github.com/MOHAMMEDDANISHMUSTAFA/ML_Project_sleep)

Yamina Bait

(https://github.com/yaminabt1/Python-ML-project)

Phuong Anh Nguyen Ngoc

(https://github.com/lexinguyen/Python-ML-Project.git)

Paula Fonte

SUMMARY

INTRODUCTION

The objectives of this study entail constructing a model capable of predicting sleeping attributes, namely TotalSleep and Dreaming, by leveraging a combination of general, ecological, and biological factors. Additionally, the aim is to analyze the correlations between diet groups, endangerment status, or Genus and these sleeping characteristics. Further exploration will be conducted to understand the interrelations and regressions within biological and ecological attributes. To achieve this, feature engineering and selection methodologies will be employed. Subsequently, the model will undergo rigorous training, evaluation, and ultimately deployment to fulfill the outlined objectives.

The process involves several key steps.

Initially, comprehensive data analysis is conducted, encompassing exploratory analysis to gain insights, feature selection to identify relevant predictors, and meticulous data preparation to ensure quality and compatibility for further analysis.

Subsequently, machine learning algorithms are trained and tested using the prepared data, employing various techniques to evaluate their performance. This evaluation includes assessing their scoring metrics to determine effectiveness and reliability, ultimately leading to the selection of the most suitable algorithm for the task at hand.

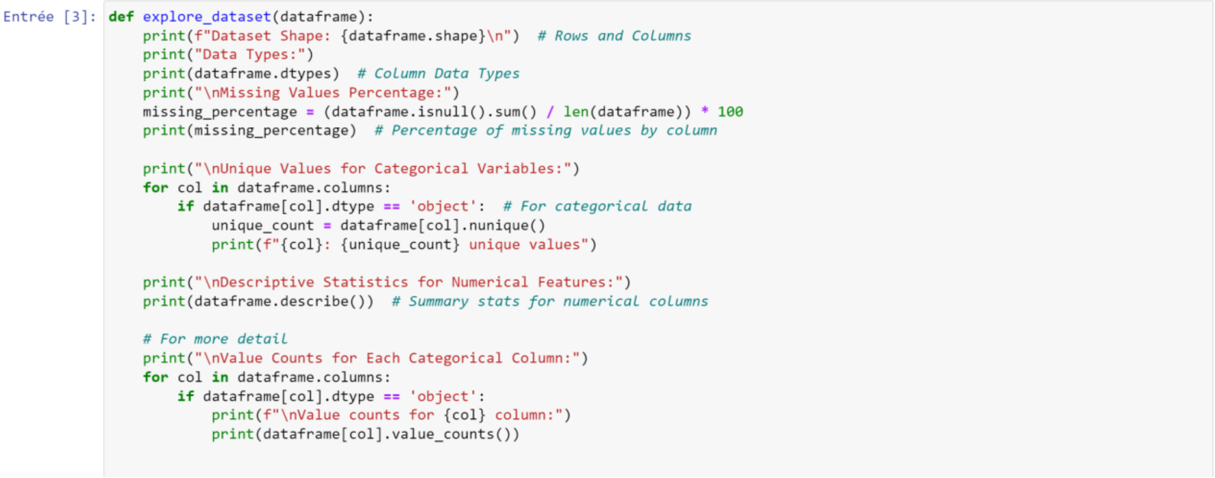
Finally, the chosen model undergoes deployment on GitHub, facilitating accessibility and potential collaboration within the community. These steps form a systematic approach towards building and deploying robust machine learning solutions.

DATA ANALYSIS

Exploratory analysis

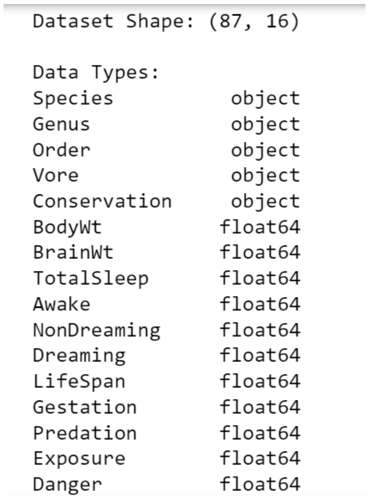
Data exploration is the first step in data analysis and typically involves summarizing the main characteristics of a data set, including its size, accuracy, initial patterns in the data and other attributes. It is commonly conducted by data analysts using visual analytics tools, but it can also be done in more advanced statistical software, Python. Before it can conduct analysis on data collected by multiple data sources and stored in data warehouses, an one must know how many cases are in a data set, what variables are included, how many missing values there are and what general hypotheses the data is likely to support. An initial exploration of the data set can help answer these questions by familiarizing analysts with the data with which they are working. We divided the data for Training and Testing purpose respectively.

The exploratory analysis function encompasses several key insights into the dataset:



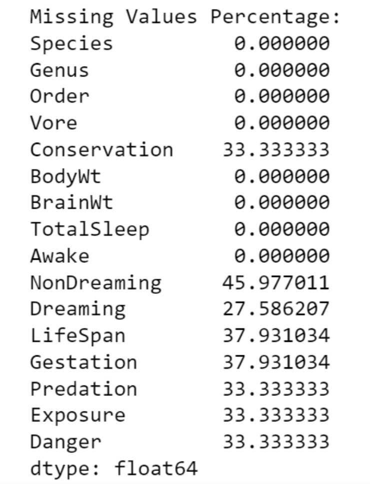
```
Entrée [3]: def explore_dataset(dataframe):
                print(f"Dataset Shape: {dataframe.shape}\n")  # Rows and Columns
                print("Data Types:")
                print(dataframe.dtypes)  # Column Data Types
                print("\nMissing Values Percentage:")
                missing_percentage = (dataframe.isnull().sum() / len(dataframe)) * 100
                print(missing_percentage)  # Percentage of missing values by column

                print("\nUnique Values for Categorical Variables:")
                for col in dataframe.columns:
                    if dataframe[col].dtype == 'object':  # For categorical data
                        unique_count = dataframe[col].nunique()
                        print(f"{col}: {unique_count} unique values")

                print("\nDescriptive Statistics for Numerical Features:")
                print(dataframe.describe())  # Summary stats for numerical columns

                # For more detail
                print("\nValue Counts for Each Categorical Column:")
                for col in dataframe.columns:
                    if dataframe[col].dtype == 'object':
                        print(f"\nValue counts for {col} column:")
                        print(dataframe[col].value_counts())
```

Dataset: provides 87 mammals with 16 attributes



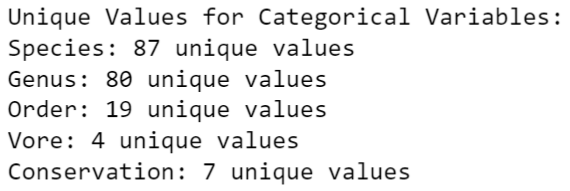
```
Dataset Shape: (87, 16)

Data Types:
Species          object
Genus            object
Order            object
Vore             object
Conservation     object
BodyWt          float64
BrainWt         float64
TotalSleep      float64
Awake           float64
NonDreaming     float64
Dreaming        float64
LifeSpan        float64
Gestation       float64
Predation       float64
Exposure        float64
Danger          float64
```

Missing Values Percentage: indicates variables with potentially problematic levels of missing data. Notable examples include NonDreaming (45.9%), Conservation (33.3%), Dreaming (27.5%), LifeSpan (37.9%), Gestation (37.9%), Predation (33.3%), Exposure (33.3%), and Danger (33.3%).



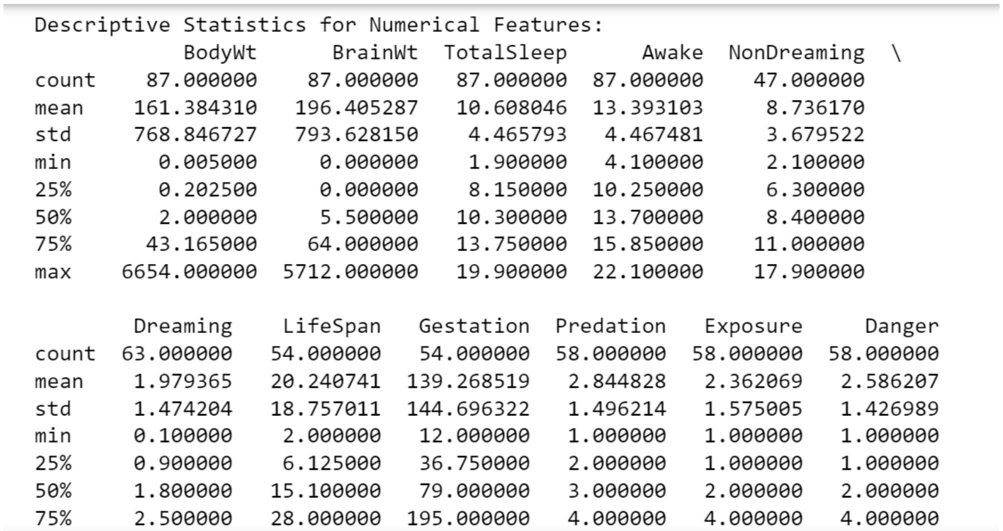
```
Missing Values Percentage:
Species          0.000000
Genus            0.000000
Order            0.000000
Vore             0.000000
Conservation    33.333333
BodyWt           0.000000
BrainWt          0.000000
TotalSleep       0.000000
Awake            0.000000
NonDreaming     45.977011
Dreaming        27.586207
LifeSpan        37.931034
Gestation       37.931034
Predation       33.333333
Exposure        33.333333
Danger          33.333333
dtype: float64
```

Unique Values for Categorical Variables: highlights variables like Species (with 87 unique values) and Genus (with 80 unique values), revealing the diversity within these categorical features.



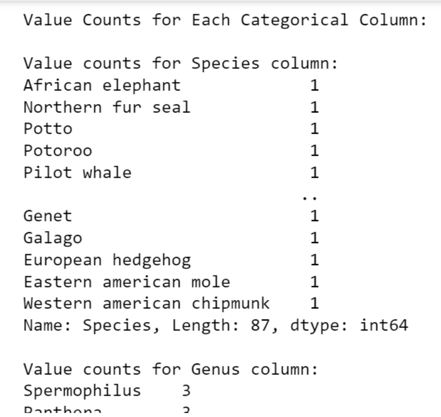
```
Unique Values for Categorical Variables:
Species: 87 unique values
Genus: 80 unique values
Order: 19 unique values
Vore: 4 unique values
Conservation: 7 unique values
```

Descriptive Statistics for Numerical Features: provides a summary of descriptive statistics for numerical columns in the DataFrame, encompassing metrics such as mean, standard deviation, minimum, and maximum values. This offers insights into the distribution and variability of numerical features.



```
Descriptive Statistics for Numerical Features:
            BodyWt      BrainWt  TotalSleep      Awake  NonDreaming  \
count    87.000000    87.000000   87.000000  87.000000    47.000000
mean    161.384310   196.405287   10.608046  13.393103     8.736170
std     768.846727   793.628150    4.465793   4.467481     3.679522
min       0.005000     0.000000    1.900000   4.100000     2.100000
25%       0.202500     0.000000    8.150000  10.250000     6.300000
50%       2.000000     5.500000   10.300000  13.700000     8.400000
75%      43.165000    64.000000   13.750000  15.850000    11.000000
max    6654.000000  5712.000000   19.900000  22.100000    17.900000

        Dreaming   LifeSpan   Gestation  Predation   Exposure     Danger
count  63.000000  54.000000   54.000000  58.000000  58.000000  58.000000
mean    1.979365  20.240741  139.268519   2.844828   2.362069   2.586207
std     1.474204  18.757011  144.696322   1.496214   1.575005   1.426989
min     0.100000   2.000000   12.000000   1.000000   1.000000   1.000000
25%     0.900000   6.125000   36.750000   2.000000   1.000000   1.000000
50%     1.800000  15.100000   79.000000   3.000000   2.000000   2.000000
75%     2.500000  28.000000  195.000000   4.000000   4.000000   4.000000
```

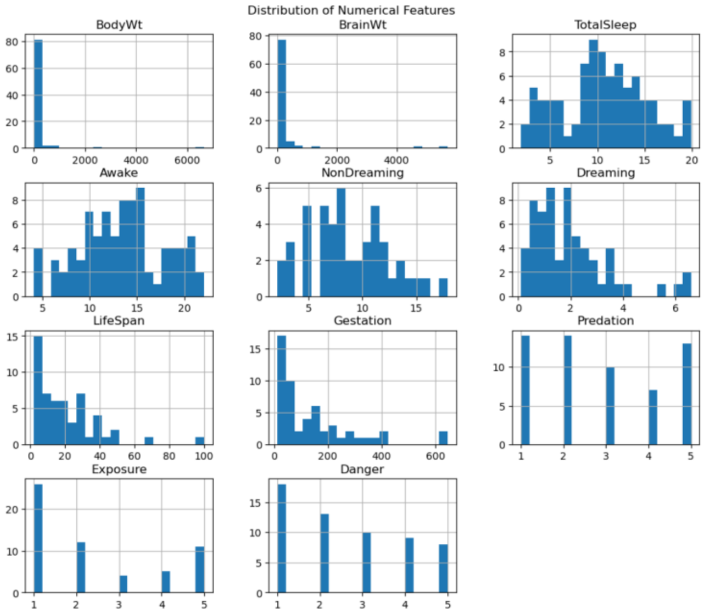
Value Counts for Each Categorical Column: As part of the exploratory analysis, this function prints the counts of unique entries for each categorical column. By identifying categorical columns based on their 'object' datatype, it presents the frequency of each unique value, aiding in the exploration of categorical variables and their distributions within the dataset.



```
Value Counts for Each Categorical Column:

Value counts for Species column:
African elephant             1
Northern fur seal            1
Potto                        1
Potoroo                      1
Pilot whale                  1
                            ..
Genet                        1
Galago                       1
European hedgehog            1
Eastern american mole        1
Western american chipmunk    1
Name: Species, Length: 87, dtype: int64

Value counts for Genus column:
Spermophilus    3
Panthera        3
```

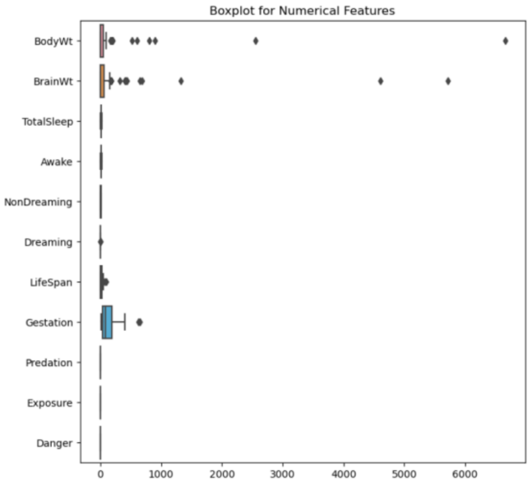
Data visualization

Data visualization is the graphical representation of information and data. Beside the describe() function, plots are more efficient to show the distribution of numerical features. . By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. In the world of Big Data, data visualization tools and technologies are essential to analyse massive amounts of information and make data-driven decisions.



Here we can see that some of the numerical features are not continuous but discrete. Like "Danger", "Exposure" and "Predation" who take discrete values from 1 to 5.

Also some of these numerical features (BodyWT and BrainWt) have outliers that obscure their true distribution on the graph. We utilize box plots as a method for detecting better outliers within the dataset.



Correlations

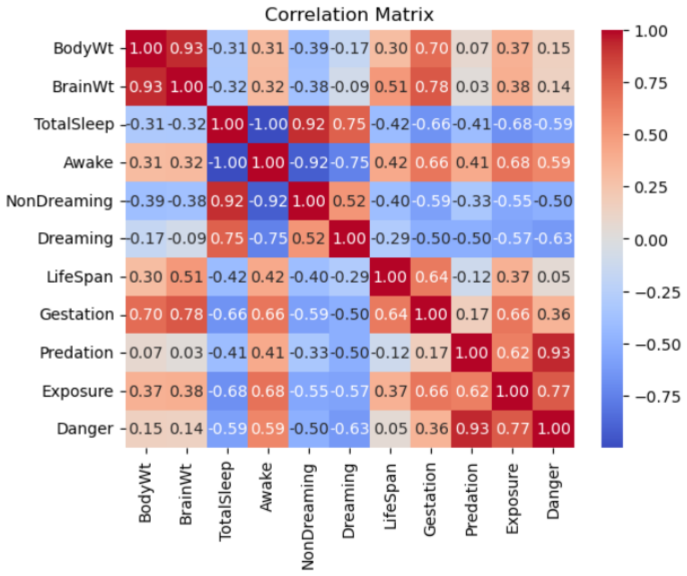
Correlation Matrix

Correlation is a statistical measure used to quantify the strength and direction of the relationship between two variables. It indicates how closely related two variables are, and whether they tend to change together.

A correlation coefficient of +1 indicates a perfect positive correlation, meaning that as one variable increases, the other variable also increases in a linear fashion. A correlation coefficient of -1 indicates a perfect negative correlation, where one variable decreases as the other increases. A correlation coefficient of 0 suggests no linear relationship between the variables.

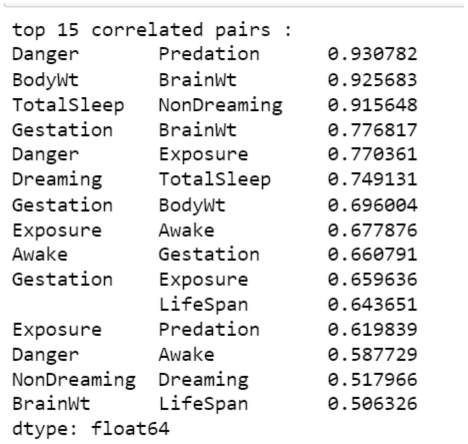
A positive correlation means that as one variable increases, the other tends to increase as well. In contrast, a negative correlation means that as one variable increases, the other tends to decrease.

The correlation matrix reveals significant relationships between various variables in the dataset.



Notably, TotalSleep demonstrates strong correlations with other sleep attributes, which is expected given their inherent relationship. However, beyond sleep attributes, notable correlations exist with Gestation (-0.66), exposure (-0.68), and Danger (0.59), although these correlations remain moderately strong. Similarly, Dreaming exhibits significant correlations with Gestation, exposure, and predation. Furthermore, biological and ecological attributes display noteworthy positive correlations, such as between predation and danger, BodyWt and BrainWt, as well as Gestation and BrainWt. The matrix also highlights both the least and most correlated pairs, offering valuable insights into the interrelationships within the dataset.

Top 3 most correlated pairs’s interpretations:



```
top 15 correlated pairs :
Danger      Predation     0.930782
BodyWt      BrainWt       0.925683
TotalSleep  NonDreaming   0.915648
Gestation   BrainWt       0.776817
Danger      Exposure      0.770361
Dreaming    TotalSleep    0.749131
Gestation   BodyWt        0.696004
Exposure    Awake         0.677876
Awake       Gestation     0.660791
Gestation   Exposure      0.659636
            LifeSpan      0.643651
Exposure    Predation     0.619839
Danger      Awake         0.587729
NonDreaming Dreaming      0.517966
BrainWt     LifeSpan      0.506326
dtype: float64
```

A positive correlation between "Danger" and "Predation" (0.930782) indicates that species facing higher predation risks tend to be perceived as more dangerous to other animals. This suggests an evolutionary adaptation where species vulnerable to predators develop defensive behaviors, potentially posing a greater threat to others.

A positive correlation between "Body Weight" and "Brain Weight" (0.92568) indicates that as the body weight of a species increases, so does the size of its brain. This suggests a relationship where larger-bodied species tend to have proportionally larger brains.

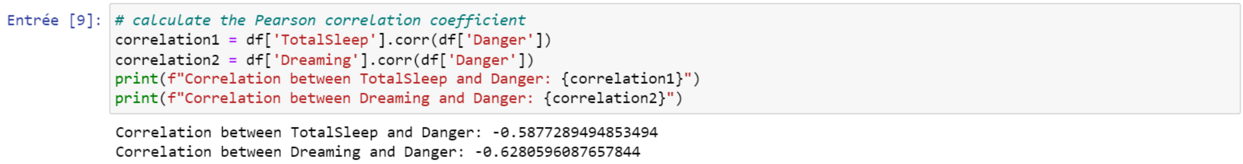
A positive correlation between "Total Sleep" and "Non-Dreaming" (0.915648) indicates that species sleeping for longer durations tend to spend more time in non-dreaming or slow-wave sleep phases.

Correlations between diet groups, danger status or Genus to the sleeping attributes

Since these three variables have different tytpes we will use different ways to assess the correlation between them and the sleep attributes (TotalSleep and Dreaming)

Danger and sleep attributes using the Pearson correlation coefficient

The Pearson correlation coefficient, denoted as "r," measures the strength and direction of the linear relationship between two continuous variables. It ranges from -1 to +1: +1 indicates a perfect positive linear relationship, -1 indicates a perfect negative linear relationship, and 0 indicates no linear relationship. It's widely used in various fields to assess relationships between variables, but it assumes linearity, normal distribution, and is sensitive to outliers.



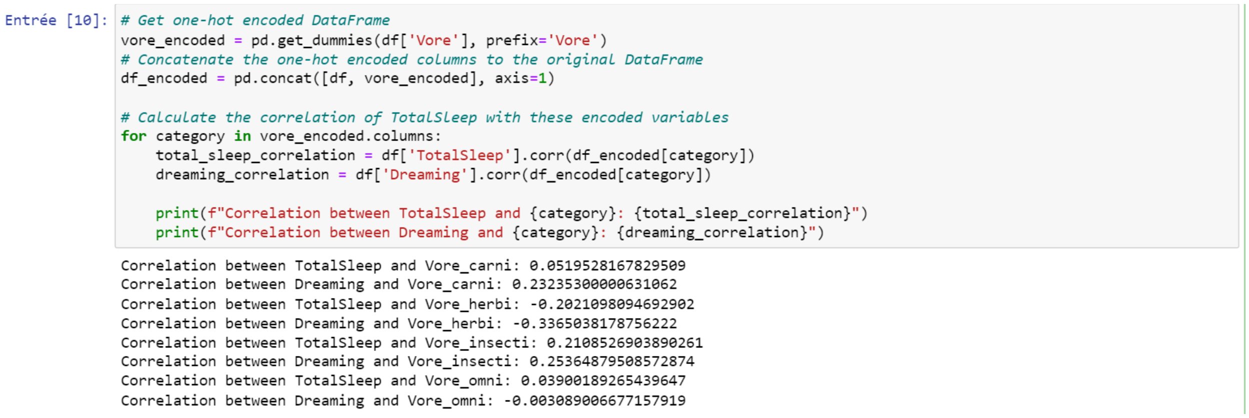
```
Entrée [9]: # calculate the Pearson correlation coefficient
            correlation1 = df['TotalSleep'].corr(df['Danger'])
            correlation2 = df['Dreaming'].corr(df['Danger'])
            print(f"Correlation between TotalSleep and Danger: {correlation1}")
            print(f"Correlation between Dreaming and Danger: {correlation2}")

            Correlation between TotalSleep and Danger: -0.5877289494853494
            Correlation between Dreaming and Danger: -0.6280596087657844
```

The correlation coefficients of -0.5877 between TotalSleep and Danger, and -0.6281 between Dreaming and Danger indicate a moderate negative linear relationship. This suggests that as the Danger rating increases, TotalSleep and Dreaming tend to decrease, or vice versa. While these correlations are significant, they're not strong enough to precisely predict TotalSleep based solely on Danger rating.

Vore and the sleep attributes using the Pearson correlation coefficient and one-hot encoding

One-hot encoding is a technique to convert categorical variables into a numerical format that can be used for model training.

In a one-hot encoded DataFrame, each categorical variable is converted into multiple binary columns, where each column represents a unique category in the original variable. For each observation, only one of these binary columns will have a value of 1, indicating the presence of that category, while the rest will be 0. This method allows machine learning algorithms to effectively interpret categorical data, but it can lead to an increase in the dimensionality of the dataset.



```
Entrée [10]: # Get one-hot encoded DataFrame
             vore_encoded = pd.get_dummies(df['Vore'], prefix='Vore')
             # Concatenate the one-hot encoded columns to the original DataFrame
             df_encoded = pd.concat([df, vore_encoded], axis=1)

             # Calculate the correlation of TotalSleep with these encoded variables
             for category in vore_encoded.columns:
                 total_sleep_correlation = df['TotalSleep'].corr(df_encoded[category])
                 dreaming_correlation = df['Dreaming'].corr(df_encoded[category])

                 print(f"Correlation between TotalSleep and {category}: {total_sleep_correlation}")
                 print(f"Correlation between Dreaming and {category}: {dreaming_correlation}")
```

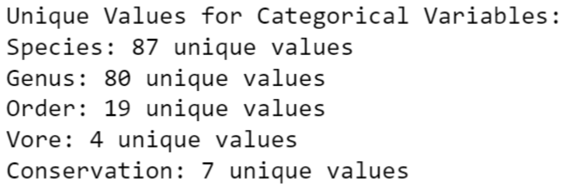
```
Correlation between TotalSleep and Vore_carni: 0.0519528167829509
Correlation between Dreaming and Vore_carni: 0.2323530000631062
Correlation between TotalSleep and Vore_herbi: -0.2021098094692902
Correlation between Dreaming and Vore_herbi: -0.3365038178756222
Correlation between TotalSleep and Vore_insecti: 0.2108526903890261
Correlation between Dreaming and Vore_insecti: 0.25364879508572874
Correlation between TotalSleep and Vore_omni: 0.03900189265439647
Correlation between Dreaming and Vore_omni: -0.003089006677157919
```

The sleep attributes do not exhibit strong linear relationships with the encoded Vore categories.

All correlation coefficients are relatively low, suggesting that any connection between an animal's diet (categorized as carnivore, herbivore, insectivore, or omnivore) and its sleep duration is likely weak and potentially influenced by other unaccounted factors.

It's important to note that correlation coefficients only capture linear relationships and may overlook non-linear associations. Therefore, these findings cannot be interpreted as evidence of direct cause-and-effect relationships without additional investigation. It's essential to remember that correlation does not imply causation.

Genus and the sleep attributes



```
Unique Values for Categorical Variables:
Species: 87 unique values
Genus: 80 unique values
Order: 19 unique values
Vore: 4 unique values
Conservation: 7 unique values
```

Examining the correlation between Genus and TotalSleep in a dataset containing 80 unique genera out of 87 observations presents challenges. The nearly one-to-one ratio of genera to data points reduces statistical power, increasing the likelihood that any observed correlation may be artificial and not reflective of a genuine relationship in the real world.

Conclusion on the most relevant attributes to predict the sleep attributes

Top-3 correlated attributes with TotalSleep: Exposure, Gestation and Danger.

Top-3 correlated attributes with Dreaming: Danger, Exposure and Predation.

Correlations within biological and ecological attributes

Biological Attributes

- Body Weight and Brain Weight (0.93): There is a very strong positive correlation between body weight and brain weight. This suggests that larger mammals tend to have larger brains. This could be associated with the principle of allometric scaling, where larger animals typically need larger brains to manage more complex bodily functions.
- Body Weight and Gestation (0.78): There is a strong positive correlation between body weight and the length of gestation. This indicates that larger mammals generally have longer gestation periods. A longer gestation period is often linked to the development of larger or more developed offspring.
- Life Span and Brain Weight (0.51): There is a moderate positive correlation between lifespan and brain weight. This could suggest that mammals with larger brains may live longer, potentially due to higher intelligence allowing for better survival strategies, or that larger brain mass is indicative of more prolonged development and maturity which is associated with longevity.
- Life Span and Gestation (0.42): There is a moderate positive correlation between lifespan and gestation period, indicating that species that have longer gestation periods tend to live longer. This could be due to the longer developmental period resulting in offspring that are more capable of survival, thereby contributing to the overall lifespan of the species.

Ecological Attributes

- Predation, Exposure, and Danger: All three indices (predation, exposure, and danger) are strongly correlated with each other (0.93 to 0.77). This implies that the mammals that are more likely to be preyed upon also tend to sleep in more exposed places and, as a result, have a higher overall danger index. These attributes are likely interlinked as animals at higher risk of predation may have evolved behaviors or characteristics that result in higher exposure and danger levels.

- Body Weight and Predation (-0.07): The weak correlation between body weight and predation suggests that body weight is not a strong predictor of the likelihood of being preyed upon. This might be due to various defense mechanisms or ecological niches occupied by mammals of different sizes that do not necessarily correlate with their weight.
- Brain Weight and Danger (0.14): There is a very weak correlation between brain weight and the overall danger index. This weak relationship might imply that brain size does not have a significant direct impact on the danger experienced from other animals, possibly because cognitive abilities (linked to brain weight) are only one of many factors affecting an animal's risk level.
- Gestation and Predation (0.41): A moderate correlation exists between the length of the gestation period and the predation index. This might suggest that species with longer gestation periods have evolved in environments with a moderate level of predation pressure, perhaps because such environments allow mothers to invest more time in developing offspring.

Additional Insights

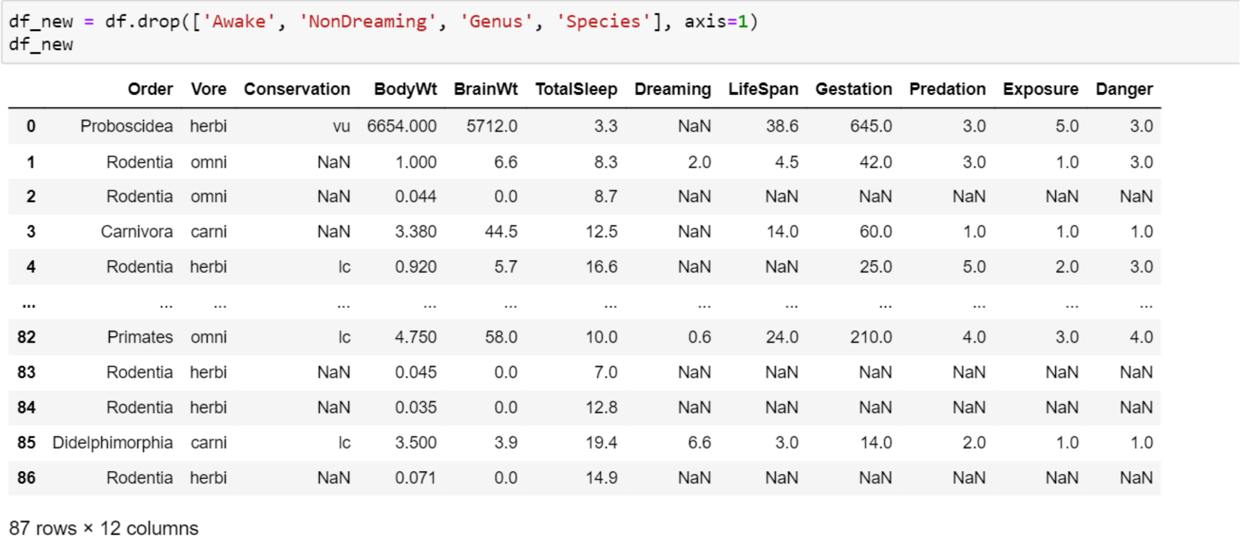
- Total Sleep, Awake, NonDreaming, Dreaming: These attributes show strong correlations with each other but are generally negatively correlated with body weight and brain weight. This could indicate that larger mammals tend to sleep less or have different sleep patterns compared to smaller mammals.
- Life Span and Total Sleep (-0.29): There is a negative correlation between lifespan and total sleep. This could suggest that species with longer lifespans may have shorter total sleep times, although the correlation is not very strong.

The correlation matrix provides a snapshot of how these attributes might be related in the biological and ecological context of these mammals. However, it's important to remember that correlation does not imply causation, and these relationships could be influenced by a variety of factors not captured in this dataset.

Feature selection and data preparation

Feature selection

Feature selection in machine learning is the process of choosing the most relevant variables for use in model construction. It aims to improve the model's performance by eliminating redundant or irrelevant data, thereby reducing overfitting, enhancing accuracy, and speeding up training. Methods for feature selection include filter, wrapper, and embedded techniques.



```
df_new = df.drop(['Awake', 'NonDreaming', 'Genus', 'Species'], axis=1)
df_new
```

	Order	Vore	Conservation	BodyWt	BrainWt	TotalSleep	Dreaming	LifeSpan	Gestation	Predation	Exposure	Danger
0	Proboscidea	herbi	vu	6654.000	5712.0	3.3	NaN	38.6	645.0	3.0	5.0	3.0
1	Rodentia	omni	NaN	1.000	6.6	8.3	2.0	4.5	42.0	3.0	1.0	3.0
2	Rodentia	omni	NaN	0.044	0.0	8.7	NaN	NaN	NaN	NaN	NaN	NaN
3	Carnivora	carni	NaN	3.380	44.5	12.5	NaN	14.0	60.0	1.0	1.0	1.0
4	Rodentia	herbi	lc	0.920	5.7	16.6	NaN	NaN	25.0	5.0	2.0	3.0
...	...	...	...	...	...	...	...	...	...	...	...	...
82	Primates	omni	lc	4.750	58.0	10.0	0.6	24.0	210.0	4.0	3.0	4.0
83	Rodentia	herbi	NaN	0.045	0.0	7.0	NaN	NaN	NaN	NaN	NaN	NaN
84	Rodentia	herbi	NaN	0.035	0.0	12.8	NaN	NaN	NaN	NaN	NaN	NaN
85	Didelphimorphia	carni	lc	3.500	3.9	19.4	6.6	3.0	14.0	2.0	1.0	1.0
86	Rodentia	herbi	NaN	0.071	0.0	14.9	NaN	NaN	NaN	NaN	NaN	NaN

87 rows × 12 columns

We need to streamline our dataset by removing redundant and irrelevant sleep attributes:

- "Awake" is redundant since it can be derived from the "TotalSleep" attribute using the formula Awake = 24 - TotalSleep.
- "NonDreaming" is redundant because it can be calculated as TotalSleep - Dreaming. Additionally, it has 47% missing values compared to Dreaming, which is detected by REM (Rapid Eye Movement).

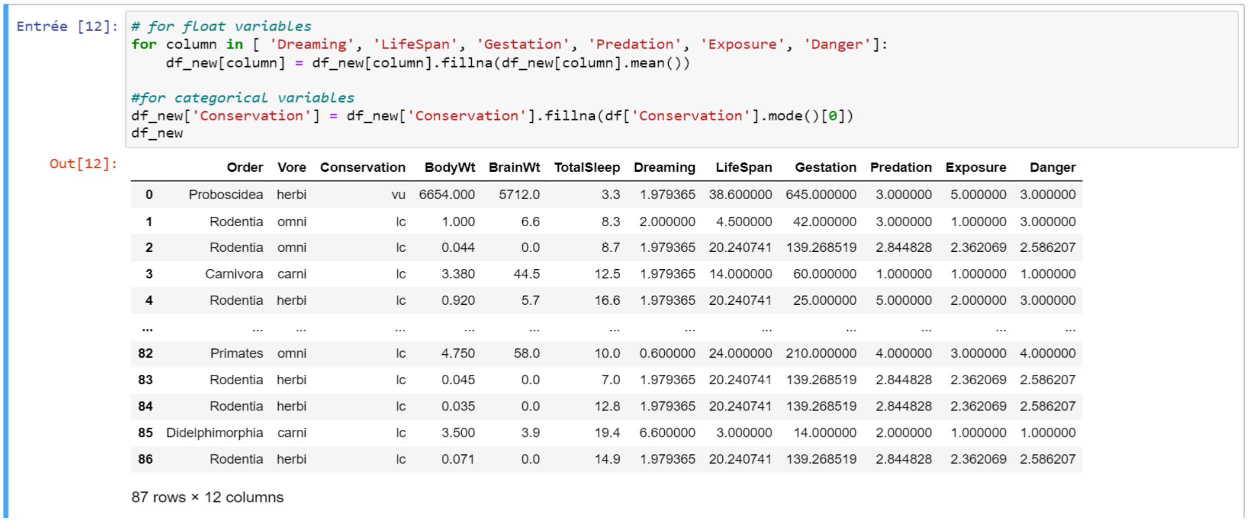
Given that Dreaming and TotalSleep are independent variables, and Awake and NonDreaming can be derived from them, we'll select Dreaming and TotalSleep as our target variables for building the predictive model.

- Eliminates "Genus" and "Species" columns from the dataset due to their high cardinality.

With 87 rows and 80 unique values in the Genus column, each category is represented by very few samples, sometimes only one. This increases the risk of overfitting, where the model may memorize specific samples rather than learning general patterns. Additionally, the sparse representation weakens the statistical power of any inference made by the model and undermines the reliability of its predictions. Removing the Genus column could mitigate these issues and lead to a more robust model capable of generalizing better to unseen data.

Data preparation

Fill null values with mean values



```
Entrée [12]: # for float variables
             for column in [ 'Dreaming', 'LifeSpan', 'Gestation', 'Predation', 'Exposure', 'Danger']:
                 df_new[column] = df_new[column].fillna(df_new[column].mean())

             #for categorical variables
             df_new['Conservation'] = df_new['Conservation'].fillna(df['Conservation'].mode()[0])
             df_new
```

Out[12]:

	Order	Vore	Conservation	BodyWt	BrainWt	TotalSleep	Dreaming	LifeSpan	Gestation	Predation	Exposure	Danger
0	Proboscidea	herbi	vu	6654.000	5712.0	3.3	1.979365	38.600000	645.000000	3.000000	5.000000	3.000000
1	Rodentia	omni	lc	1.000	6.6	8.3	2.000000	4.500000	42.000000	3.000000	1.000000	3.000000
2	Rodentia	omni	lc	0.044	0.0	8.7	1.979365	20.240741	139.268519	2.844828	2.362069	2.586207
3	Carnivora	carni	lc	3.380	44.5	12.5	1.979365	14.000000	60.000000	1.000000	1.000000	1.000000
4	Rodentia	herbi	lc	0.920	5.7	16.6	1.979365	20.240741	25.000000	5.000000	2.000000	3.000000
...	...	...	...	...	...	...	...	...	...	...	...	...
82	Primates	omni	lc	4.750	58.0	10.0	0.600000	24.000000	210.000000	4.000000	3.000000	4.000000
83	Rodentia	herbi	lc	0.045	0.0	7.0	1.979365	20.240741	139.268519	2.844828	2.362069	2.586207
84	Rodentia	herbi	lc	0.035	0.0	12.8	1.979365	20.240741	139.268519	2.844828	2.362069	2.586207
85	Didelphimorphia	carni	lc	3.500	3.9	19.4	6.600000	3.000000	14.000000	2.000000	1.000000	1.000000
86	Rodentia	herbi	lc	0.071	0.0	14.9	1.979365	20.240741	139.268519	2.844828	2.362069	2.586207

87 rows × 12 columns

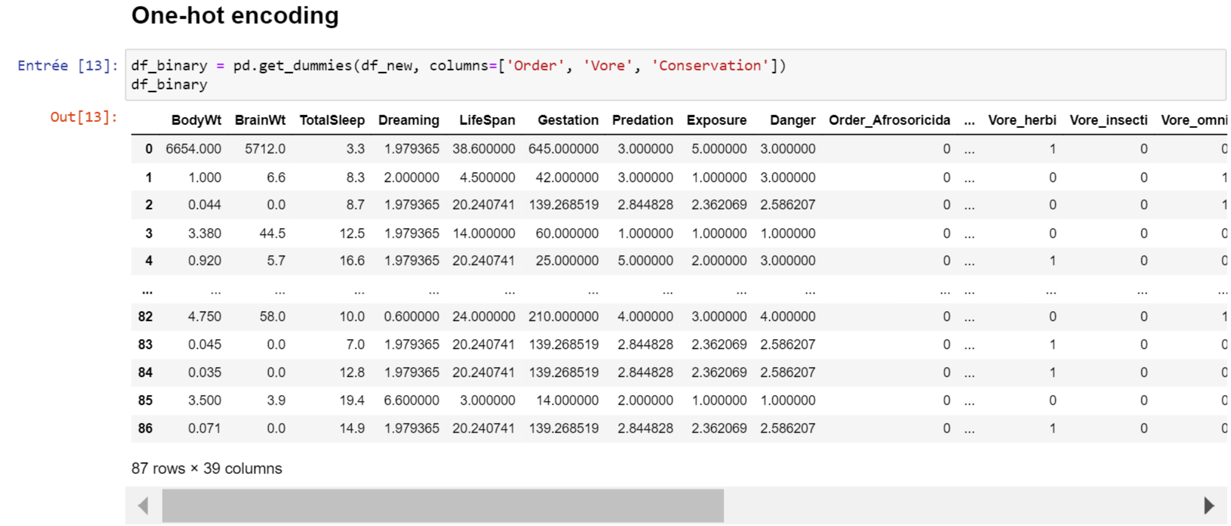
For each column (assumed to be numerical), missing values are replaced with the mean of the non-missing values in that column.

For the 'Conservation' column (assumed to be categorical), missing values are filled with the most frequent value of that column in the original DataFrame

One-hot encoding

By converting categorical variables into a numerical format, it allows machine learning algorithms to effectively interpret and learn from these features.

One-hot encoding ensures that each category within a categorical variable is treated independently, avoiding any implicit ordering or hierarchy that could bias the model's predictions.



One-hot encoding

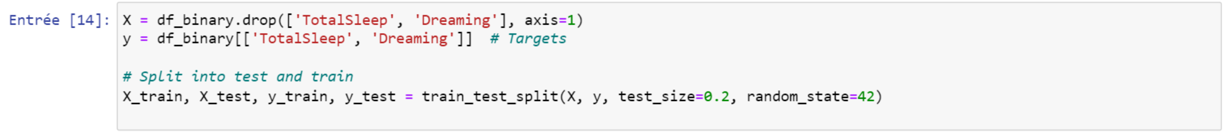
```
Entrée [13]: df_binary = pd.get_dummies(df_new, columns=['Order', 'Vore', 'Conservation'])
             df_binary
```

Out[13]:

	BodyWt	BrainWt	TotalSleep	Dreaming	LifeSpan	Gestation	Predation	Exposure	Danger	Order_Afrosoricida	...	Vore_herbi	Vore_insecti	Vore_omni
0	6654.000	5712.0	3.3	1.979365	38.600000	645.000000	3.000000	5.000000	3.000000	0	...	1	0	0
1	1.000	6.6	8.3	2.000000	4.500000	42.000000	3.000000	1.000000	3.000000	0	...	0	0	1
2	0.044	0.0	8.7	1.979365	20.240741	139.268519	2.844828	2.362069	2.586207	0	...	0	0	1
3	3.380	44.5	12.5	1.979365	14.000000	60.000000	1.000000	1.000000	1.000000	0	...	0	0	0
4	0.920	5.7	16.6	1.979365	20.240741	25.000000	5.000000	2.000000	3.000000	0	...	1	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
82	4.750	58.0	10.0	0.600000	24.000000	210.000000	4.000000	3.000000	4.000000	0	...	0	0	1
83	0.045	0.0	7.0	1.979365	20.240741	139.268519	2.844828	2.362069	2.586207	0	...	1	0	0
84	0.035	0.0	12.8	1.979365	20.240741	139.268519	2.844828	2.362069	2.586207	0	...	1	0	0
85	3.500	3.9	19.4	6.600000	3.000000	14.000000	2.000000	1.000000	1.000000	0	...	0	0	0
86	0.071	0.0	14.9	1.979365	20.240741	139.268519	2.844828	2.362069	2.586207	0	...	1	0	0

87 rows × 39 columns

Train test split



```
Entrée [14]: X = df_binary.drop(['TotalSleep', 'Dreaming'], axis=1)
             y = df_binary[['TotalSleep', 'Dreaming']]  # Targets

             # Split into test and train
             X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

1. Defining Features and Targets:

Selects the features (independent variables) for the model. It creates a new DataFrame ‘x’ by dropping the columns 'TotalSleep' and 'Dreaming' from the original DataFrame.

Selects the targets (dependent variables) for our model. It creates a new DataFrame `y` containing only the columns 'TotalSleep' and 'Dreaming'.

2. Splitting Data into Training and Testing Sets:

20% of the data will be used for testing, while the remaining 80% will be used for training.

MODEL TRAINING

Models building and testing

XGBoost

- XGBoost is a powerful and versatile ensemble learning algorithm that is known for its exceptional performance in both regression and classification tasks.
- It offers scalability, flexibility, and efficiency, making it suitable for large datasets and a wide range of applications.
- Its ability to handle complex relationships in the data, feature importance analysis, and robustness to overfitting make it a popular choice for predictive modeling.

Linear Regression

- Linear Regression is a machine learning algorithm based on supervised learning.
- It performs a regression task. Regression models a target prediction value based on independent variables.
- It is mostly used for finding out the relationship between variables and forecasting.

SVR

- SVR is a variant of Support Vector Machines (SVMs) that is used for regression tasks. It works by finding the hyperplane that best fits the data while maximizing the margin between the data points and the hyperplane.
- SVR is effective in capturing nonlinear relationships in the data through the use of kernel functions.
- It can handle high-dimensional data and is robust to outliers, making it suitable for datasets with complex patterns and noise.

Decision tree regressor

- Decision trees are intuitive and easy to interpret models that partition the feature space into regions and make predictions based on the average target value of the samples within each region.
- Decision trees are versatile and can handle both numerical and categorical data without requiring extensive preprocessing.
- They are robust to outliers and can capture nonlinear relationships in the data, making them suitable for a wide range of regression tasks.

KNeighbors regressor

- KNeighbors Regressor is a simple yet effective non-parametric algorithm for regression tasks. It predicts the target value for a new sample by averaging the target values of its k nearest neighbors in the feature space.
- It is easy to understand and implement, making it a good choice for baseline modeling and initial exploration of the data.

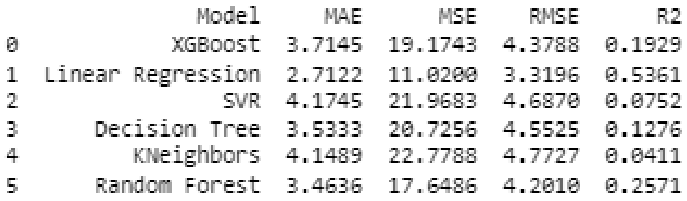
- KNeighbors Regressor is robust to noisy data and can capture complex relationships in the data, although it may struggle with high-dimensional datasets or datasets with many irrelevant features.

Random Forest Regressor

- A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees.
- Bagging, in the Random Forest method, involves training each decision tree on a different data sample where sampling is done with replacement.
- The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

Models Comparaison

We use somes metrics to compare models:



```
               Model     MAE      MSE    RMSE      R2
0            XGBoost  3.7145  19.1743  4.3788  0.1929
1  Linear Regression  2.7122  11.0200  3.3196  0.5361
2                SVR  4.1745  21.9683  4.6870  0.0752
3      Decision Tree  3.5333  20.7256  4.5525  0.1276
4         KNeighbors  4.1489  22.7788  4.7727  0.0411
5      Random Forest  3.4636  17.6486  4.2010  0.2571
```

1. **Mean Absolute Error (MAE):**

- MAE measures the average absolute difference between the predicted and actual values.
- It gives an indication of the average magnitude of errors in the predictions.
- For example, a MAE of KNeighbors of 4.15 means that, on average, the predicted TotalSleep values differ from the actual values by approximately 4.15 hours.

2. **Mean Squared Error (MSE):**

- MSE measures the average squared difference between the predicted and actual values.
- Squaring the errors penalizes larger errors more heavily than smaller ones.
- For example, a MSE of KNeighbors of 22.78 means that, on average, the squared differences between the predicted and actual TotalSleep values amount to approximately 22.78.

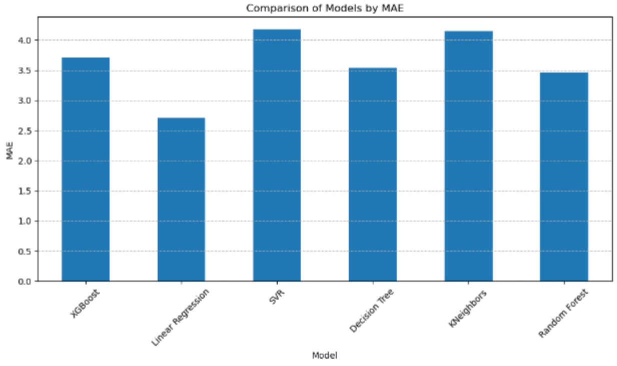
3. **Root Mean Squared Error (RMSE):**

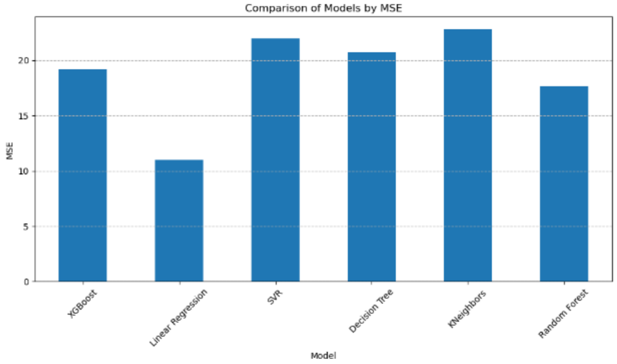
- RMSE is the square root of the MSE and provides a measure of the typical deviation of the predicted values from the actual values.
- It's in the same units as the target variable, making it easier to interpret.
- In this context, an RMSE of 4.77 means that, on average, the predicted TotalSleep values deviate from the actual values by approximately 4.77 hours.

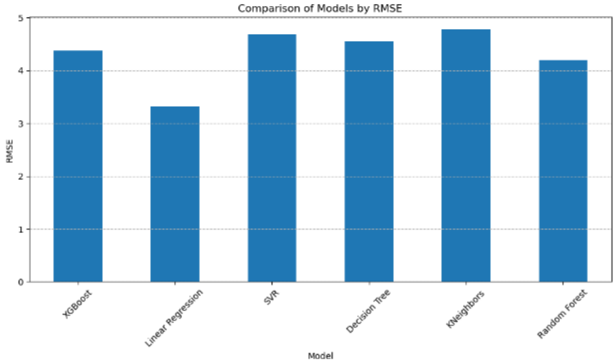
4. **R-squared (Coefficient of Determination):**

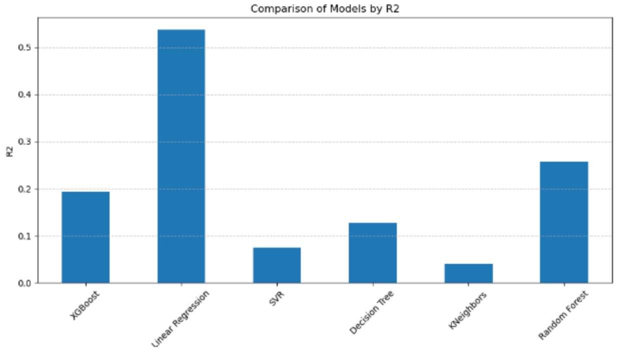
- R-squared measures the proportion of the variance in the dependent variable (TotalSleep) that is explained by the independent variables (features) in the model.
- It ranges from 0 to 1, where 0 indicates that the model explains none of the variance and 1 indicates that the model explains all of the variance.
- In this case, an R-squared of 0.041 means that the model explains only about 4.1% of the variance in TotalSleep, indicating that the model's predictive power is limited.

We use plots to compare different models by each metric to facilitate the study of the model's efficiency.









RESULT

To define the best model among multiple models based on these metrics, it must correspond to these criteria.

1. Minimization of Error Metrics: Look for models with lower values of error metrics such as MAE, MSE, and RMSE. A lower error indicates better performance in terms of accuracy and precision.
2. Maximization of R-squared: Seek models with higher values of R-squared. A higher R-squared value indicates that a larger proportion of the variance in the dependent variable is explained by the model, suggesting better overall fit.

By these criteria, the study showed that Linear Regression displayed the best performance for this Dataset and can be used for deploying purposes.

CONCLUSION

In conclusion, our machine learning project aimed to predict sleep variables (Total Sleep and Dreaming) in mammals. After evaluating various models, we found that linear regression performed the best based on the chosen metrics.

However, it's essential to note that Dreaming originally had 33% missing values, potentially affecting the accuracy of predictions for this variable. Therefore, while linear regression is the preferred model for predicting Total Sleep, caution should be exercised when interpreting predictions for Dreaming, as they may be less accurate due to the high percentage of missing values.

Further efforts to address missing data or alternative modeling approaches may be necessary to improve the prediction accuracy of Dreaming in future iterations of the project.