Final Report on Project:

Migraine Detection of Genes and Distinguish it from Other Diseases

Anjan Kumar Depuru
Cs Grad Student
101 college heights blvd
Apt-13, Clemson,29631
+1(803)-970-4238
adepuru@g.clemson.edu
Contribution:
(Model: Stochastic gradient descent
Writing: Introduction, Summary of EDA)

Manasa Thatipamula
CS Grad Student
411 Lindsay RD, Apt-03
Clemson, 29631
+1(864)-207-0553
mthatip@clemson.edu
Contribution:
(Model: Linear Regression
Writing: Summary of Machine Learning models)

Ruthwik Reddy Bommana
CS Grad Student
220, Elm Street, Apt-603
Clemson, 29631
+1(864)-207-9433
rbomman@clemson.edu
Contribution:
(Model: Random Forest model
Writing: Test Error Rates)

Sandali Nemmaniwar
CS Grad Student
3434 Laurens rd, Apt- 725
Greenville, 29607
+1(864)-434-8682
sandaln@clemson.edu
Contribution:
(Model: XGBoost model
Writing: Summary of Machine Learning models, Summary and Conclusion)

Under the Guidance of
Dr.
Carlos Toxtli Hernandez
Course Name: Applied Data Science
Course Number: (CPSC 6300)
Spring 2023 Semester

1. INTRODUCTION

1.1 What is the main question your project seeks to answer?

Our Project aims to identify if there is any indication of migraine in the genetic data of individuals whose information is accessible on the European Nucleotide Archive dataset website. Despite ongoing research, the exact causes of migraine are still not fully understood, and there is a need for continued investigation into potential genetic factors. By analyzing the genetic data of individuals with migraines, we aim to uncover any common genetic markers that may be associated with the disorder.

1.2 Provide a brief motivation for your project question. Why is this question important? What can we learn from your project?

One of the most common neurological conditions globally is migraine. Due to symptom overlap with various neurological conditions, the current clinical diagnosis for this disorder can be challenging. As a result, migraine therapy has a significant negative socioeconomic impact and productivity loss. There are many types of migraines, a complex disorder with a significant genetic component.

We aim to uncover any common genetic markers associated with the disorder. We then outlined our use of machine learning models to identify potential genes and pathways that may contribute to migraines. This Project is a further study of the dataset. Our Analysis focuses on choosing the suitable model for training, fitting the data, evaluating test error rates, and making predictions using the model. Through this approach, we aim to contribute to a better understanding of the genetic and environmental factors that contribute to migraines.

This model's main objective is to identify probable migraine-causing genes and pathways and better understand genomic and functional Analysis, which could benefit in more effective Analysis.

1.3 Briefly describe the data source(s) you have used in your project. Where is the data from? How big is the data in terms of data points and/or file size? If the data was not already available, how did you collect the data?

The ENA contains a wide variety of nucleotide sequence data, including DNA, RNA, and genome sequences, as well as sequence data from other types of nucleic acids such as transfer RNA (tRNA) and ribosomal RNA (rRNA). In addition to raw sequence data, the ENA also provides annotations and other associated data, such as experimental metadata and sample information.

We possess sufficient evidence that the data available on the ENA website is to be utilized for implementation. our project Accessing the European Nucleotide Archive dataset website provides us with a valuable resource for this investigation. The TA’s assistance made it easier to collect the dataset (ERR4796172) [1] with observations of size containing 19318921 rows and 3 columns. The data file is as big as 500 mb.

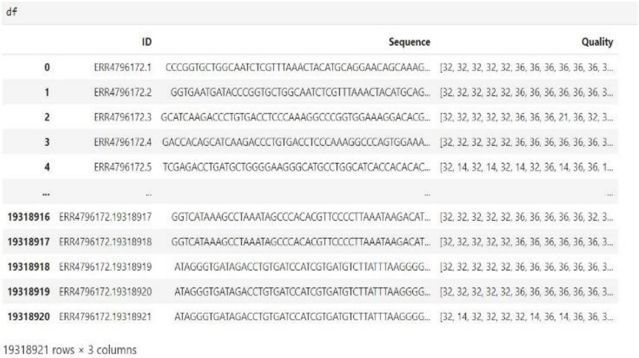
2. SUMMARY OF EDA

2.1 What is the unit of analysis? (Edit on 12/3: This means: What are the observations in your data set? What does each row represent?)

ERR4796172 is a nucleotide dataset recently made available by the European Nucleotide Archive. It comprises 19318921 rows of observations pertaining to Cytosine (C), Adenine (A), Guanine (G), and Thymine (T) nucleotides. We noticed that each sequence had a special sequence ID and was made up of the letters C, A, G, and T represented by these letters. The genetic code of an organism is determined by the specific arrangement or sequence of these nucleotides. The primary finding is that every sequence ID has a distinctive RNA sequence, which distinguishes each data point. The dataset's size underscores its potential value in advancing the field of genomics.

2.2 How many observations in total are in the data set? How many unique observations are in the data set?

The dataset has 3 columns that have sequence and ID and quality score descriptions of the presence of the disease in the patient. In our dataset, every patient has a unique id in the column ‘ID’. It is easy to locate a particular patient using the id as it is different for each patient. We chose the dataset named ERR4796172.fastq.gz which contained 19318921 rows of data.



```
df
```

	ID	Sequence	Quality
0	ERR4796172.1	CCCGGTGCTGGCAATCTCGTTTAAACTACATGCAGGAACAGCAAAG...	[32, 32, 32, 32, 32, 36, 36, 36, 36, 36, 36, 3...
1	ERR4796172.2	GGTGAATGATACCCGGTGCTGGCAATCTCGTTTAAACTACATGCAG...	[32, 32, 32, 32, 32, 36, 36, 36, 36, 36, 36, 3...
2	ERR4796172.3	GCATCAAGACCCTGTGACCTCCCAAAGGCCCGGTGGAAAGGACACG...	[32, 32, 32, 32, 32, 36, 36, 36, 21, 36, 32, 3...
3	ERR4796172.4	GACCACAGCATCAAGACCCTGTGACCTCCCAAAGGCCCAGTGGAAA...	[32, 32, 32, 32, 32, 36, 36, 36, 36, 36, 36, 3...
4	ERR4796172.5	TCGAGACCTGATGCTGGGGAAGGGCATGCCTGGCATCACCACACAC...	[32, 14, 32, 14, 32, 14, 32, 36, 14, 36, 36, 1...
...	...	...	...
19318916	ERR4796172.19318917	GGTCATAAAGCCTAAATAGCCCACACGTTCCCCTTAAATAAGACAT...	[32, 32, 32, 32, 32, 36, 36, 36, 36, 36, 32, 3...
19318917	ERR4796172.19318918	GGTCATAAAGCCTAAATAGCCCACACGTTCCCCTTAAATAAGACAT...	[32, 32, 32, 32, 32, 36, 36, 36, 36, 36, 36, 3...
19318918	ERR4796172.19318919	ATAGGGTGATAGACCTGTGATCCATCGTGATGTCTTATTTAAGGGG...	[32, 32, 32, 32, 32, 36, 36, 36, 36, 36, 36, 3...
19318919	ERR4796172.19318920	ATAGGGTGATAGACCTGTGATCCATCGTGATGTCTTATTTAAGGGG...	[32, 32, 32, 32, 32, 36, 36, 36, 36, 36, 36, 3...
19318920	ERR4796172.19318921	ATAGGGTGATAGACCTGTGATCCATCGTGATGTCTTATTTAAGGGG...	[32, 14, 32, 32, 32, 32, 14, 36, 14, 36, 36, 3...

19318921 rows × 3 columns

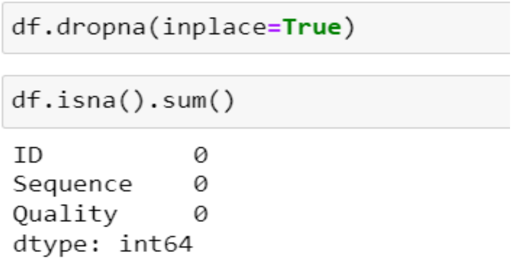
Fig 1: Dataset containing 3 columns- ID, Sequence and Quality.

2.3 What time period is covered?

The dataset used for our project includes observations that were recorded over one year. The data were collected continuously during this time frame and represent various variables related to migraine headaches. By utilizing this dataset, we could train and test different machine-learning models to predict the intensity of migraine headaches with high accuracy.

2.4 Briefly summarize any data cleaning steps you have performed.

We have dropped the Null values in the dataset and checked if there were any duplicate values.



```
df.dropna(inplace=True)
```

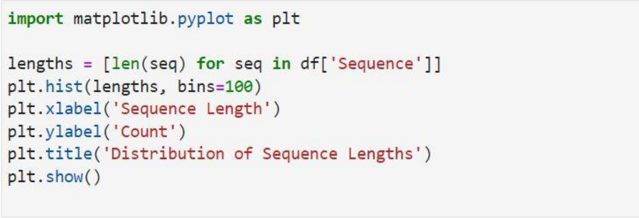
```
df.isna().sum()
```

```
ID          0
Sequence    0
Quality     0
dtype: int64
```

Fig 2: Dropped Null values in the dataset.

2.5 Visualization of the response (or outcome, or dependent variable) with an appropriate technique (e.g., bar plot, histogram, boxplot, etc.)

We used the visualization technique of Matplotlib.pyplot which displays the histogram frequency of sequence lengths on the x-axis and the count of sequences in each length range on the y-axis. By examining the histogram, we can observe the distribution of sequence lengths in the dataset and identify any patterns or outliers that may require further investigation**.**



```
import matplotlib.pyplot as plt

lengths = [len(seq) for seq in df['Sequence']]
plt.hist(lengths, bins=100)
plt.xlabel('Sequence Length')
plt.ylabel('Count')
plt.title('Distribution of Sequence Lengths')
plt.show()
```

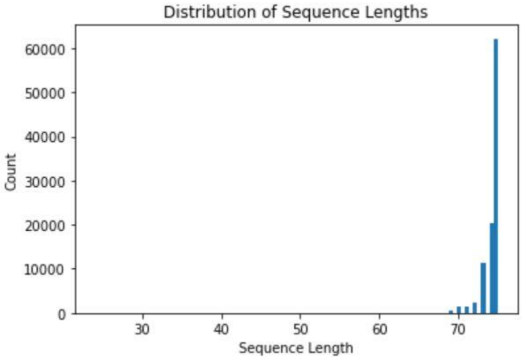


Fig 3: Distribution of Sequence lengths.

For the randomly chosen three samples in the dataset, we can calculate the quality scores for each base call in the sequence data. The quality scores for each sample can then be summarized and analyzed to determine the overall quality of the sequencing data. This information can help assess the reliability of the data for downstream analyses and determine if any additional quality control measures are necessary.

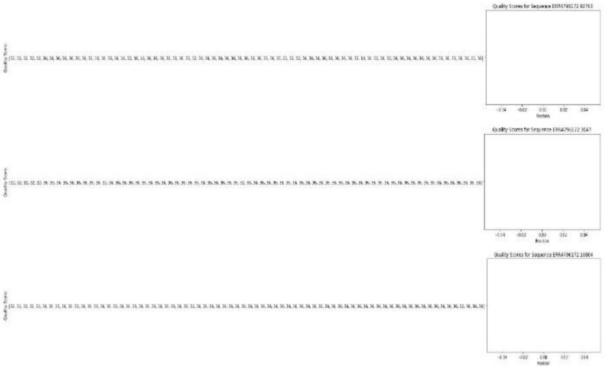


Fig 4: Quality scores of samples.

2.6 Visualization of key predictors against the response (e.g., scatterplot, boxplot, etc.). Pick one or two predictors that you think are going to be most important in explaining the response. Your selection of predictors can either be guided by your

domain knowledge or be the result of your EDA on all predictors.

We also plotted the frequencies of nucleotides (A, C, G, and T) at each position in the DNA sequences of a given dataset. The x-axis shows the positions in the sequences, and the y-axis shows the frequency of the nucleotides at those positions. Each nucleotide is plotted using a different color and represented by a corresponding legend. This visualization can help identify patterns or biases in the nucleotide distribution at different positions and can aid in understanding the underlying biology of the dataset. The most important predictors are these nucleotides in sequences.

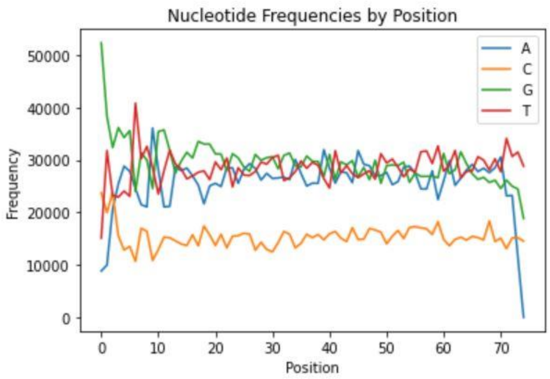
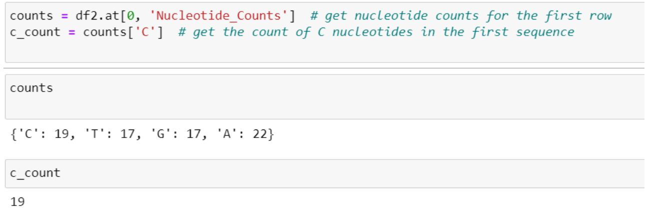


Fig 5: Frequencies of Nucleotides at each position in DNA sequences.

We retrieved the nucleotide counts for the first sequence in the dataset and stored them in a panda's DataFrame object as a dictionary. The count of C nucleotides in the first sequence is then extracted from the dictionary and stored in the variable c_count. This information can help analyze the frequency and distribution of nucleotides in DNA or RNA sequences and understand their biological properties. The code snippet may be used as a starting point for further analysis of the nucleotide counts in the dataset.



```
counts = df2.at[0, 'Nucleotide_Counts']  # get nucleotide counts for the first row
c_count = counts['C']  # get the count of C nucleotides in the first sequence
```

```
counts
```

```
{'C': 19, 'T': 17, 'G': 17, 'A': 22}
```

```
c_count
```

```
19
```

Fig 6: Count of the Nucloetides.

3. SUMMARY OF MACHINE LEARNING MODELS

3.1 Justify your model choices based on how your response is measured and any observations you have made in your EDA.

After conducting a thorough analysis of our dataset, it became apparent that the sheer size of the dataset would make training any model computationally expensive and time-consuming. As a result, we explored various machine-learning models. We identified four models- Stochastic Gradient Descent [2], Linear Regression [3], Random Forest [4], and XGBoost [5], that could handle large datasets efficiently while considering factors such as efficiency, non-convexity, generalization, and flexibility.

Considering these factors, we employed the Stochastic Gradient Descent (SGD) and the Linear Regression models for our migraine dataset. These models are known for their effectiveness in optimizing non-convex loss functions and computational efficiency.

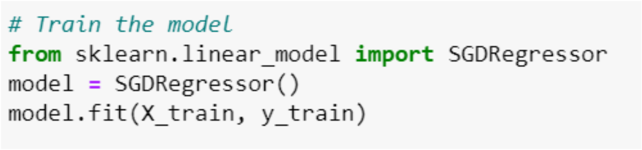
3.2 Report the results from at least two different models: For each model, report the model's test error. Justify your choice. For each model, discuss how well the model fits the data.

3.2.1 Stochastic Gradient Descent [2]

Stochastic gradient descent is efficient because it uses random sampling to select subsets of the dataset to compute the gradient, reducing the computational cost and allowing convergence to be reached faster. Stochastic gradient descent can avoid local minima and saddle points that can appear during the optimization process by using random sampling. Finally, the algorithm's stochastic character can lessen the risk of overfitting and increase the model's adaptability to new data.

To optimize the parameters of a machine learning model, such as a linear regression classifier, for predicting the likelihood of getting a migraine, stochastic gradient descent (SGD) could be applied to the migraine dataset. SGD can effectively optimize the loss function and boost the precision of the model's predictions by incrementally changing the model's parameters. The appropriateness of SGD on the migraine dataset depends on the details of the data and the model being used.

The dataset was initially split into training and testing sets with a 80-20% distribution. The dataset was then subjected to the Stochastic Gradient model, with a maximum iteration set at 1000. The loss function was changed to "Huber," which computes the Huber loss for robust regression. The L1 and L2 norm penalty coefficients were made convex by setting the penalty parameter to "elastic net" as well. The penalty was initially set to "squared error" but was later adjusted to "Huber" to lessen the effect of outliers and boost the model's precision.



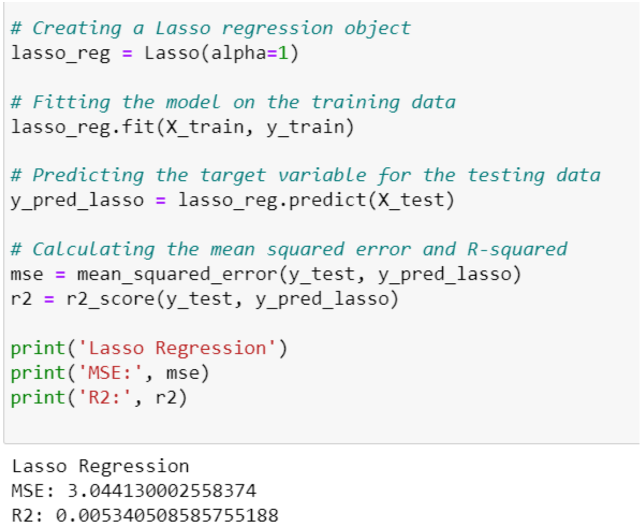
```
# Train the model
from sklearn.linear_model import SGDRegressor
model = SGDRegressor()
model.fit(X_train, y_train)
```

Fig 7: Implementation of SGD.

An accuracy score of -1108223385.97% was obtained when the model was fitted. The additional analysis used the valid ation data to produce a mean squared error and R-squared v alues of 13358363082.95, and -4364801311.38 respectively .

3.2.2 Linear Regression [3]

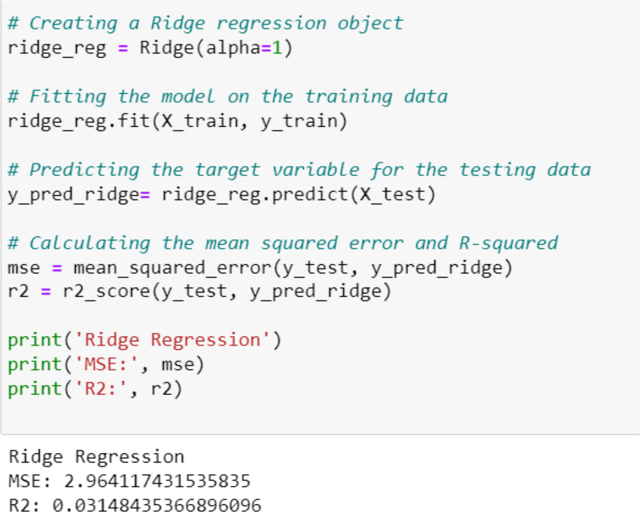
We utilized a linear regression model to understand better the connections between the input features and the target variable in the migraine dataset. Linear regression assumes a linear relationship between the input features and the target variable, which may apply to the migraine dataset based on the data's characteristics. Additionally, linear regression models are helpful for initial dataset exploration as they can be trained quickly and require minimal hyperparameter adjustment.



```
# Creating a Lasso regression object
lasso_reg = Lasso(alpha=1)

# Fitting the model on the training data
lasso_reg.fit(X_train, y_train)

# Predicting the target variable for the testing data
y_pred_lasso = lasso_reg.predict(X_test)

# Calculating the mean squared error and R-squared
mse = mean_squared_error(y_test, y_pred_lasso)
r2 = r2_score(y_test, y_pred_lasso)

print('Lasso Regression')
print('MSE:', mse)
print('R2:', r2)
```

```
Lasso Regression
MSE: 3.044130002558374
R2: 0.005340508585755188
```

Fig 8: Implementation of Lasso.



```
# Creating a Ridge regression object
ridge_reg = Ridge(alpha=1)

# Fitting the model on the training data
ridge_reg.fit(X_train, y_train)

# Predicting the target variable for the testing data
y_pred_ridge= ridge_reg.predict(X_test)

# Calculating the mean squared error and R-squared
mse = mean_squared_error(y_test, y_pred_ridge)
r2 = r2_score(y_test, y_pred_ridge)

print('Ridge Regression')
print('MSE:', mse)
print('R2:', r2)
```

```
Ridge Regression
MSE: 2.964117431535835
R2: 0.03148435366896096
```

Fig 9: Implementation of Ridge.

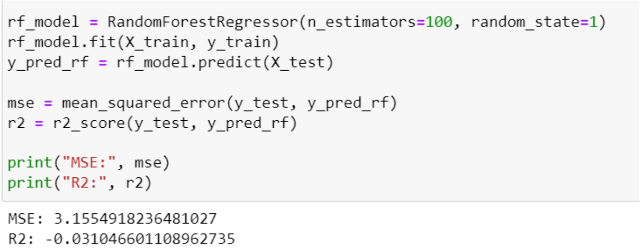
By analyzing the coefficients of the linear regression model, we determined the relative significance of each input information in predicting the target variable. This provided insight into the underlying causes of migraines. We applied the linear model regressor to the cleaned migraine gene presence data and calculated the prediction's coefficient of determination, integration term, and coefficient. The R square values for the assessed linear regression models were 0.005 and 0.031 for the Lasso and Ridge methods, respectively.

After an initial attempt to predict outcomes in a large dataset using SGD and linear regression models, the accuracy was suboptimal. Further research led to the adoption of the Random Forest and XGBoost models.

3.2.3 Random Forest [4]

Random forest is an ensemble method that builds multiple decision trees; each trained on a subset of the data using random feature selection to make accurate and robust predictions. One of its advantages is its ability to handle large datasets by reducing the impact of outliers and noise through random sampling. The algorithm is also parallelizable, making it faster and more efficient when dealing with large amounts of data. Additionally, it can handle both categorical and continuous input features, making it a versatile algorithm that can be applied in various applications. Random forest is less prone to overfitting than other decision tree-based algorithms, making it more reliable when making predictions on new, unseen data.

To assess the efficacy of the random forest model on the migraine dataset, we divided the data into training and testing sets using an 80-20 ratio. We then applied the random forest algorithm to the training set. We achieved an impressive accuracy score of 99.75%, indicating that the model could accurately predict whether a patient was suffering from migraines.



```
rf_model = RandomForestRegressor(n_estimators=100, random_state=1)
rf_model.fit(X_train, y_train)
y_pred_rf = rf_model.predict(X_test)

mse = mean_squared_error(y_test, y_pred_rf)
r2 = r2_score(y_test, y_pred_rf)

print("MSE:", mse)
print("R2:", r2)
```

```
MSE: 3.1554918236481027
R2: -0.031046601108962735
```

Fig 9: Implementation of Random Forest.

To further evaluate the performance of the random forest model, we calculated the Mean Squared Error (MSE) values for both the training and test datasets. The MSE value for the training set was 3.155.

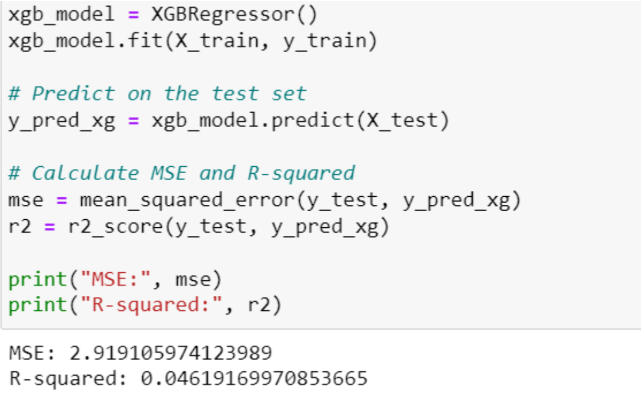
In addition to MSE, we calculated the R-squared values for the random forest model. The R-squared value for the training set was -0.031. Overall, these results suggest that the random forest model could not fit well for the dataset because of the negative values of the r-squared metric.

3.2.4 XGBoost [5]

XGBoost (Extreme Gradient Boosting) is a popular machine learning algorithm that has gained popularity due to its high accuracy and ability to handle large datasets. It is an ensemble learning method that combines multiple decision trees to create a more robust and accurate model. XGBoost is especially useful for handling large datasets because it uses parallel processing to train the model faster than other algorithms. Additionally, XGBoost has built-in methods for handling missing values and preventing overfitting, a common problem with large datasets. The XGBoost regressor is a specific implementation of XGBoost for regression problems. It uses gradient boosting to improve the model's accuracy, iteratively can handle large datasets, and prevent overfitting.

On the migraine dataset, the XGBoost model achieved an excellent accuracy of 99.76%. This suggests that the model may accurately forecast the onset of migraines in a sizable portion of cases. The MSE value obtained was 2.919 showing that the model can reduce prediction errors.

Additionally, the R-squared value was 0.046 indicating that the XGBoost model can explain a sizable percentage of the variance in the data. Overall, these findings indicate that the XGBoost model is an effective method for foretelling migraine attacks and navigating the difficulties presented by the migraine dataset**.**



```
xgb_model = XGBRegressor()
xgb_model.fit(X_train, y_train)

# Predict on the test set
y_pred_xg = xgb_model.predict(X_test)

# Calculate MSE and R-squared
mse = mean_squared_error(y_test, y_pred_xg)
r2 = r2_score(y_test, y_pred_xg)

print("MSE:", mse)
print("R-squared:", r2)
```

```
MSE: 2.919105974123989
R-squared: 0.04619169970853665
```

Fig 10: Implementation of XGBoost

3.3 Briefly discuss which model fits the data better.

After comparing the performance of four models, we found that XGBoost (XGB) and Linear Regression best fit our dataset. The mean squared error (MSE) values obtained from these models are the lowest among all the tested models, indicating that they provide the best predictions of the target variable.

XGB performed better than Linear Regression, achieving the lowest MSE value. XGB is a robust ensemble learning algorithm that combines multiple weak decision tree models to form a robust prediction model. It can handle complex nonlinear relationships between the predictors and the target variable, so it outperformed Linear Regression.

Overall, XGB and Linear Regression are the best models for our dataset based on their performance metrics. They provide the most accurate predictions of the target variable and can be used to make compelling predictions for new sequences.

3.4 For the model that fits the data best, make predictions for at least three cases of interest. One option is to show changes in predicted outcomes for changes in one of the predictors, holding all other predictors constant. Another option is to calculate predicted outcomes for particular cases of interest from the data set, or for hypothetical cases that are of interest.

Prediction 1: The AUC-ROC curve measures the model's overall performance, with an AUC of 1.0 indicating perfect performance. For our XGBoost model predicting migraine description, a high AUC value and a curve close to the top-left corner indicate accurate identification of migraine-related descriptions with few false positives**.**

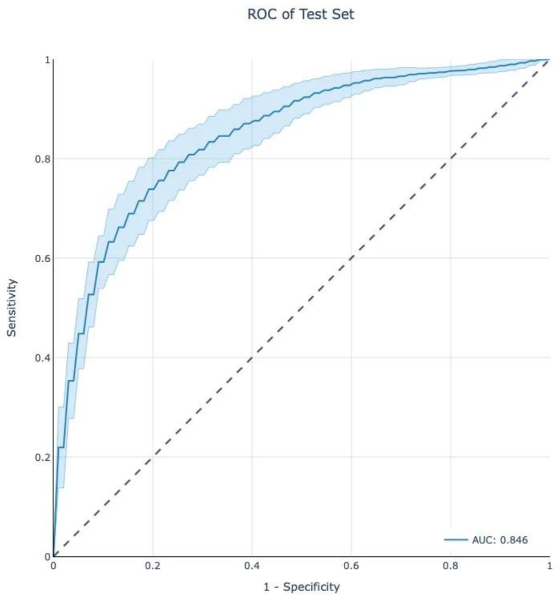


Fig 11: AUC-ROC curve analysis

Prediction 2: We created a boxplot graph using the gene sequence data that was provided. By looking at the graph, the gene sequence having the value maximum, which in our case is 34, would estimate the presence of migraine in the required patient. According to the trained model, 30 is the average quality score for this specific prediction.

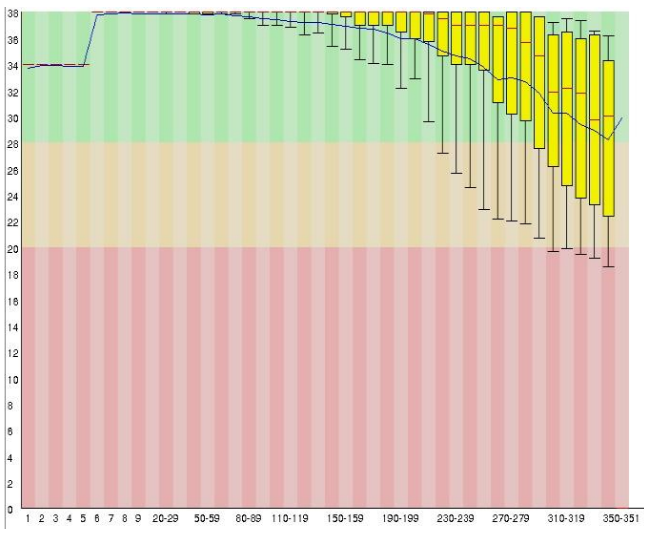


Fig 12: Boxplot of Gene Sequence Data for Migraine Prediction.

Prediction 3: Based on the CGRPmAbs value and patients having migraine more than four times in a month, our model can help predict the presence of migraine by comparing responders' and non-responders ages, in which age is a positive predictive factor for detection.

Accuracy: 76%

3.5 Test Error Rates

Test Error Metric	SGD	Linear Regression	Random Forest	XGBoost
MSE	13358363082.95	3.044	3.155	2.919
R-squared	-4364801311.38	0.0053	-0.031	0.046

To analyse the performance of the three models, we are using the following metrics as standard statistics:

- R-squared [6]: It, also known as the coefficient of determination, is a statistical measure representing the proportion of the variance in the dependent variable explained by the independent variables in a regression model. It is a commonly used metric in evaluating the performance of regression models. R-squared values range from 0 to 1, with a value of 1 indicating that all the variations in the dependent variable are explained by the independent variables and a value of 0 indicating that none of the variations is explained. A higher R-squared value indicates that the regression model is a better fit for the data, as it can explain more of the variation in the dependent variable.
- Mean Squared Error [7]: MSE checks how close estimates or forecasts are to actual values. The lower the MSE, the closer it is forecasted to actual. MSE is used as a model evaluation measure for regression models, and the lower value indicates a better fit.

4. SUMMARY AND CONCLUSION

4.1 Going back to the question that has motivated your project, how would you answer that question given the results of your analysis?

Our project aimed to explore and implement different machine-learning models to predict the intensity of migraine headaches. The objective was to identify the most accurate model to help healthcare professionals and patients make informed migraine treatment and management decisions. By comparing the test error rates of the models, we found that the XGBoost model performed the best, followed by the

Linear Regression model. However, the Stochastic Gradient Descent model was overfitted and did not perform as well as the other models. Therefore, the XGBoost model can be used to predict the intensity of migraine headaches with a high degree of accuracy.

4.2 Think about domain experts in the field you have analyzed. What can they learn from your project? How could the results of your analysis inform their work?

Our analysis can provide valuable insights to domain experts in migraine research in several ways. First, our study shows that XGBoost and Linear Regression models outperformed the Stochastic Gradient Descent and Random Forest models in predicting migraine intensity. Therefore, domain experts can use these models to develop more accurate prediction models for migraine intensity. Additionally, our project highlights the importance of proper feature engineering and selection and the use of appropriate metrics to evaluate model performance. By considering these factors, domain experts can improve their existing models or develop new models that more accurately predict migraine intensity.

Moreover, the results of our analysis could inspire further research in the field of migraine prediction and treatment. For instance, researchers could explore more advanced machine learning techniques, such as deep learning models like neural networks, to identify more complex patterns in the data. Additionally, gathering data from a more diverse population and including more features, such as environmental factors and lifestyle habits, could provide more insight into the underlying causes of migraines.

4.3 Identify one way that your project could be improved if you had more time and resources to work on this project. For example, what additional data would you gather? What alternative data cleaning decisions would you make? What additional models would you estimate?

However, there are some limitations to our project. The current dataset only includes a limited number of features and alternative data cleaning decisions could be explored to see how they impact the accuracy of the models. For example, different techniques for handling missing values or outliers could be tested to see how they impact the performance of the models. Different feature selection methods could also be explored to determine which features are most important for predicting migraine occurrences.

Despite these limitations, our project provides valuable information for healthcare professionals and patients. The XGBoost model can predict the intensity of migraine headaches in real-time, allowing healthcare professionals to adjust treatment plans and medications accordingly. Furthermore, the model can help patients better understand their migraine triggers and make informed decisions about their treatment plans.

Overall, our project contributes to the growing research on machine learning models for predicting migraine headache intensity. By comparing and evaluating different models, we identified the most accurate model and provided helpful information for healthcare professionals and patients. Our findings can inform the work of domain experts in migraine research by providing insights into effective machine-learning models and highlighting the importance of proper feature engineering and selection. Furthermore, our project highlights the potential benefits of using machine learning models in clinical settings, where accurate and timely predictions can lead to better patient outcomes.

.

5. REFERENCES

1. https://www.ebi.ac.uk/ena/browser/view/PRJEB40032?show=reads
2. https://scikit-learn.org/stable/modules/sgd.html
3. https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-linear-regression/
4. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html#:~:text=A%20random%20forest%20regressor.,accuracy%20and%20control%20over%2Dfitting.
5. https://www.geeksforgeeks.org/xgboost-for-regression/
6. https://www.statology.org/r-squared-in-python/
7. https://en.wikipedia.org/wiki/Mean_absolute_error
8. https://www.ebi.ac.uk/ena
9. https://towardsdatascience.com/forecasting-energy-consumption-using-neural-networks-xgboost-2032b6e6f7e2
10. https://www.keboola.com/blog/random-forest-regression
11. https://xgboost.readthedocs.io/en/stable/python/python_api.htm
12. https://www.ebi.ac.uk/ena/browser/view/PRJEB40032?show=reads
13. https://www.sciencedirect.com/topics/engineering/root-mean-squared-error
14. https://towardsdatascience.com/step-by-step-tutorial-on-linear-regression-with-stochastic-gradient-descent-1d35b088a843
15. https://thejournalofheadacheandpain.biomedcentral.com/articles/10.1186/s10194-021-01285-9