Song Release Year Prediction

SAN JOSE STATE UNIVERSITY

CMPE 255

GROUP 10 PROJECT REPORT

Group Members:

Charanveer Singh (011826308),

Moxank Prakashbhai Patel (016041064),

Xinyu He (012491544),

Hongjin Cheng (016032406)

GitHub link:
https://github.com/Moxi3231/song-year-preduction.git

I. Introduction

A song is a musical composition intended to be performed by the human voice. Each piece has its unique melody or tone, etc. These characteristics make up the song. Everyone has their favorite type of song, and this is because there are songs in each genre, and most of these songs in the same genre share some similar characteristics. In addition to personal preferences, some trendy songs from each era represent the characteristics of the most popular song genres. Most of the songs from different periods also have the features of the song style belonging to the period. By analyzing the characteristics of these songs and combining them with the time of their release, we can get a period in which we can judge the possible release of different songs.

OBJECTIVE

In this project, we will use the audio features provided in the dataset to predict the year of the song release. This will be done using the data mining techniques learned in the course and applying these techniques to the 90 attributes/audio features provided in the dataset. We will design a prediction model and train it with the dataset. By refining this model, it can predict the release time of the input songs.

II. System Design And Implementation

ALGORITHMS CONSIDERED/SELECTED

We consider the following Algorithm:

1. **Logistic Regression (*selected*)**

Due to its simplicity and the fact that the target variable is categorical, logistic Regression was chosen. Given that a song can fall under one of 90 categories (years), ordinal logistic Regression is used in this instance.

2. **Decision Tree (*selected*)**

Another well-liked supervised machine learning model for Classification and Regression issues is the Decision Tree. In our scenario, we used a decision tree classifier by method, which predicts the target class using a tree-like structure.

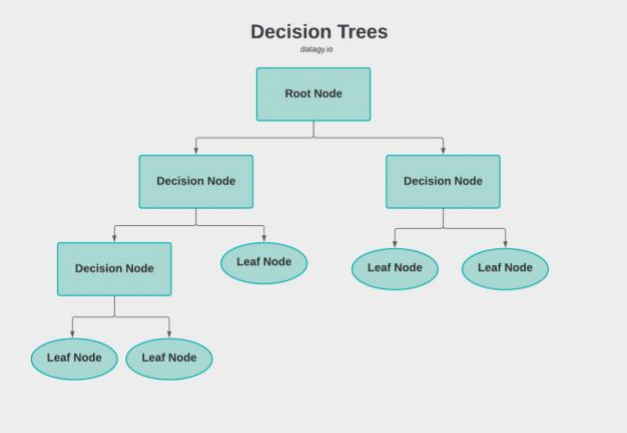

Figure 1: Decision Tree Structure

Decision Nodes consist of a decision statement, and leaf nodes are the binary answers. For example, if the decision Node statement is (age > 5), the Connected leaf nodes will say yes or no. Depending on the answer, we travel further down the Tree.

3. Random Forest (*selected*)

Random Forest Classifier for our project uses multiple Decision Trees to improve the accuracy score. A bagging (bootstrap + Aggregation) approach is used in this model. Using the original dataset is created using this technique. The final solution is chosen from these mini-decision trees using the majority answer.

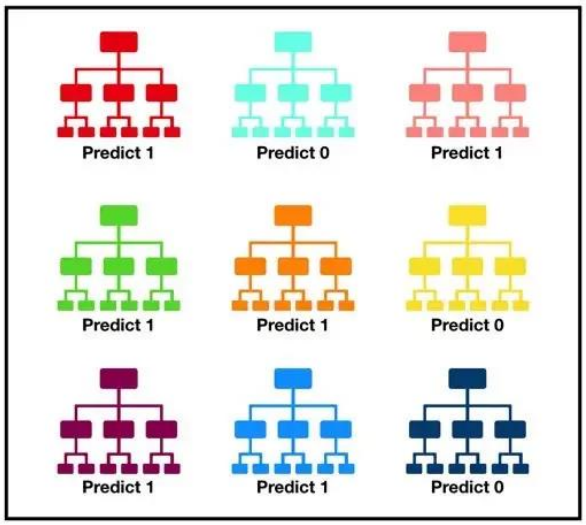

Figure 2: Random Forests Structure

4. Neural Network (*selected*)

Neural Networks are best for identifying the hidden pattern in the data. We initially tried a simple neural network which performed better than other models. So to improve on that, we tried hyper tuning by increasing the epochs and decreasing/ increasing the learning rate. Adjusting the clipping parameters of the model and so forth. And finally, we came up with a design which can be seen in Figure 3. This can be thought of as 15 different neural networks trying to predict the output, and lastly, the output from this is combined, and the final classification is done.

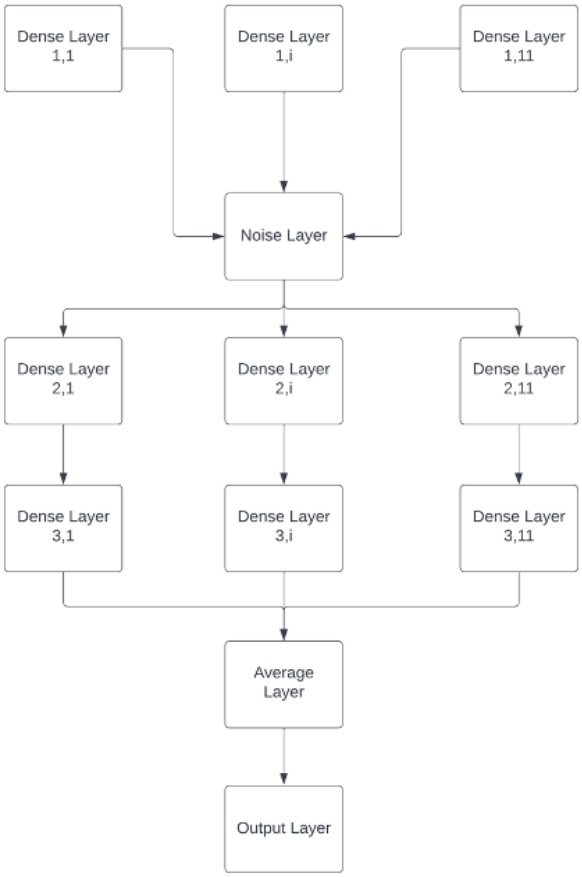

Figure 3. The final design of the neural network.

As the data was imbalanced, we used Categorical Cross Entropy as our loss function. We used Recall and Precision as metrics to evaluate the model's performance. Finally, we also used accuracy as our performance metric to assess the model on the test dataset.

For training the neural network, we employed K Fold Cross Validation. The value of k here was 20% of the training dataset. Further, to deal with the skew dataset and overcome overfitting, we added a noise layer in the neural network, preventing the model from being overfitted and providing better performance on unseen data.

The metrics of the loss function, recall, and Precision is illustrated below in Figure 4-6. They describe the model's performance on training data.

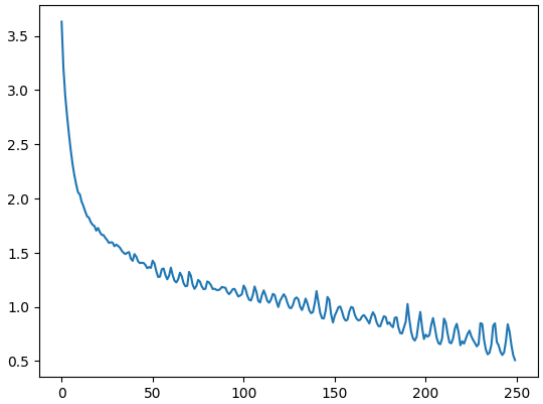

Figure 4: Categorical Cross Entropy Loss Function for 250 epochs

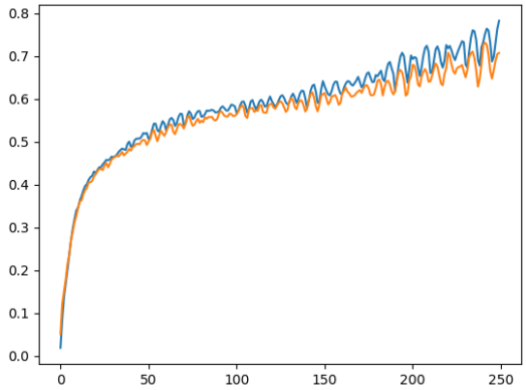

Figure 5: Precision vs. Validation Precision for 250 epochs

Orange: Validation data's Precision
Blue: Train data precision

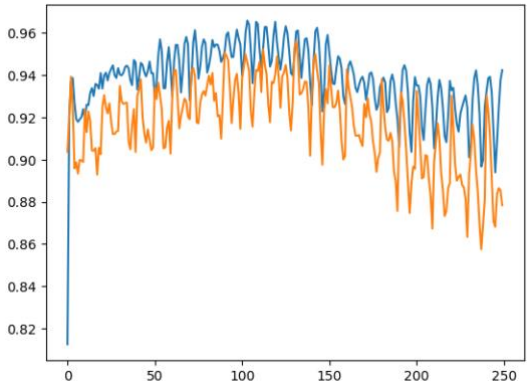

Figure 6: Train Recall vs. Validation Recall

Orange: Recall for validation dataset
Blue: Recall for Train dataset

5. LINEAR REGRESSION AND SUPPORT VECTOR MACHINE (*not selected*)

While using linear Regression and SVM, we didn't achieve much performance because the data didn't have a high correlation.

TECHNOLOGY AND TOOLS USED

- Jupyter Notebook (*python3*): We used Jupyter Notebook for easy code collaboration.
- For Regression, Classification we used sklearn.
- For Neural Networks, we used Tensorflow.
- For representing data, we used NumPy and pandas.
- For balancing the dataset, we used the imblearn library.
- For visualizing the data, we used matplotlib.
- Lastly, we also used GitHub to maintain the code.

ARCHITECTURE RELATED DECISIONS

- Specific to our data mining task: Despective method or predictive method?

  As the dataset consisted of labels, it was evident that the problem came under predictive learning.

- Two project parts: Data pro-processing / Model selection and performance.

  In the data preprocessing part, we did the following:

  1. Feature Extraction
  2. Feature Selection
  3. Data Visualization
  4. PCA (Dimensionality Reduction)
  5. Balancing the dataset.
  6. Splitting data into training and testing

In the Model selection part, we tried different models, then compared several performance metrics to find the best model.

EXPLORATORY DATA ANALYSIS

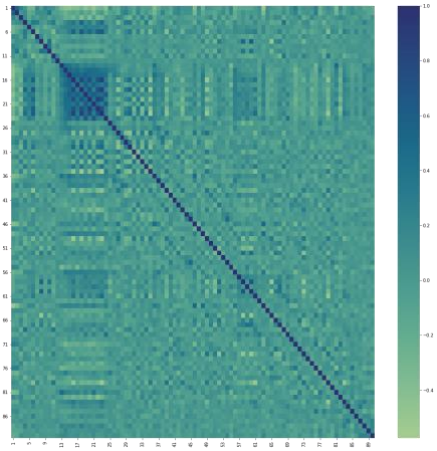

Figure 7: Correlation Heatmap among variables

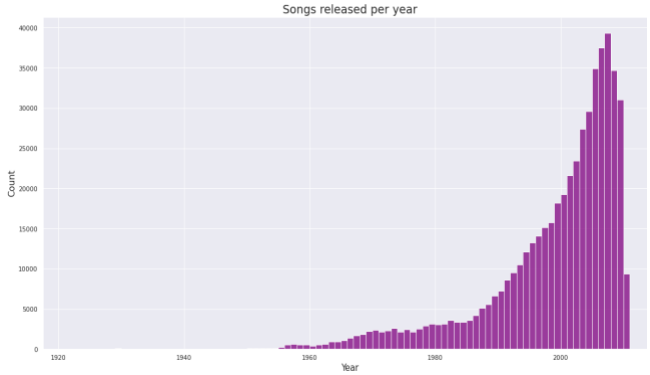

Figure 8: Songs per year plot

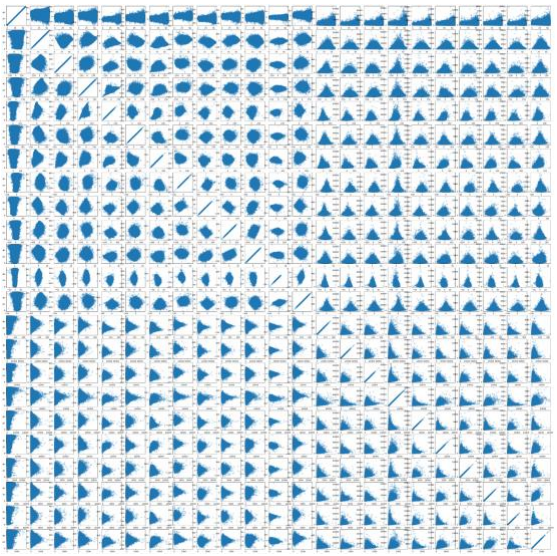

Figure 9: Scatter plot depicting the variance between features before dimensionality reduction

Figure 9 explains the data variance between the first 23 features. The variance is very high, and also the data is scattered. Due to this, models suffer, and performance degrades on unseen data.

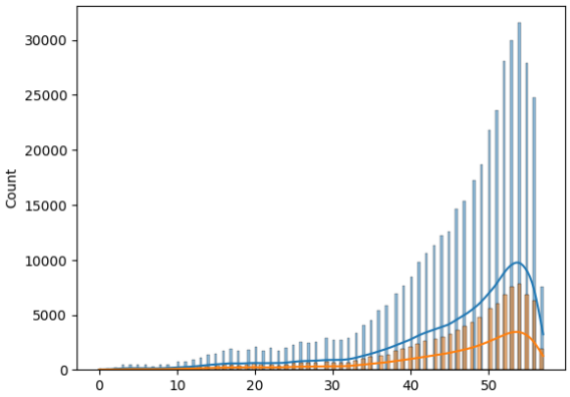

Figure 10: Removing outliers and transformed classes

Figure 8 shows the class distribution; some have no or significantly fewer records, and they act as noise for the model. Due to it, the performance wasn't increasing. So we removed those records and reduced the number of classes from 90 to 58. This is visualized in Figure 10.

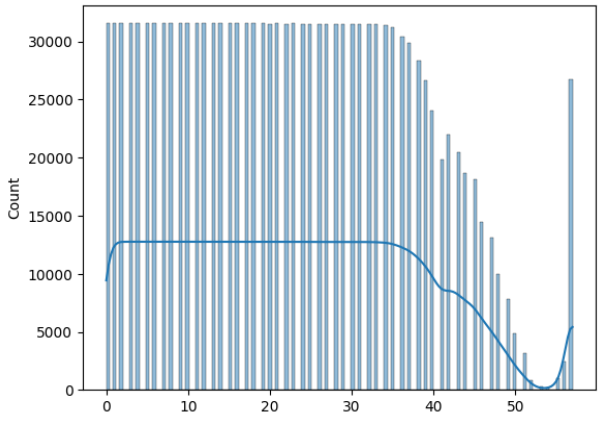

Figure 11: Balanced Dataset.

Still, after removing certain records, the dataset was highly imbalanced, so we used upsampling and SMOTE technique to balance the dataset. After using them, we got a balanced dataset which can be visualized in Figure 11. The number of records in the final training dataset was around 14 lakhs.

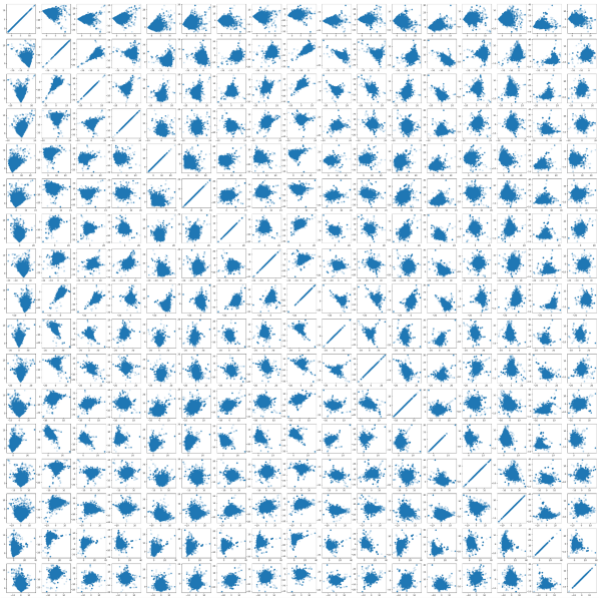

Figure 12: Scatter plot showing the variance between components after dimensionality reduction.

To understand the dataset better, visualization in Figure 1 above, for instance, illustrates the significant correlation between variables 13 and 24. Additionally, Figure 2 explains the vast majority of songs released in the 2000s.

As the dataset was still skewed and the dimensions were 90, we decided to reduce the variance, ultimately decreasing the variance. We used PCA to reduce the variance of the dataset. We only kept features that accounted for 95% of the variance. Figure 12 depicts the variance of the reduced-dimension dataset. Although the data still has some variance but comparatively, it is relatively lesser.

III. Experiments

DATASET USED

Name: YearPredictionMSD Data Set
Source:
https://archive.ics.uci.edu/ml/datasets/YearPredictionMSD
Type of data: Multivariate
Size of data: 427MB
Number of instances: 515345
Preprocessing Performed: Remove Outliers and missing values, Drop duplicate values, PCA for Dimensionality Reduction

The entire dataset of the experiments is 280GB, the subset of 10,000 songs(1%) is 1.8 GB. However, it is still huge, and we only use the data to predict the year of a song by some input attribute. Hence, we continue to select our dataset YearPredictionMSD, which is only 449MB.

DESIGN METHODOLOGY

After a quick dataset analysis, the problem could be easily solved using simple logistic Regression. So we applied logistic Regression on the original dataset after splitting it into training and test dataset. But we didn't get the performance as the data is heavily skewed. To overcome the skew dataset, we applied dimensionality reduction. After using it, we tried logistic Regression again. The performance did improve, but it wasn't much.

So we approached the problem from a new perspective. We attempted to classify the records into years instead of regressing them. For classification, first, we tried using a Decision Tree. The performance was worse than the Regression. So we tried using the Random Forest Classifier. The performance was better than logistic Regression. So we tried Random Forest with different parameters. But we didn't get much of the performance. So on further analysis of the dataset, we learned that even after dimensionality reduction, the dataset had too much imbalance and was still skewed. To overcome that, we employed SMOTE to balance the class distribution. After utilizing SMOTE, we reduced the dimensions. Then we developed a simple neural network, and the observation was that the neural network outperformed previous models.

But the accuracy of the model was only 50%. So, upon further analysis, we found that some features had little to no contribution. So, iteratively we added features and saw if it improved the performance or not. Then we found some features that didn't help with improving the model performance and removed them. This was feature elimination. After that, we tried various combinations of neural networks, and After hyper-tuning, we finally got an accuracy of 88%.

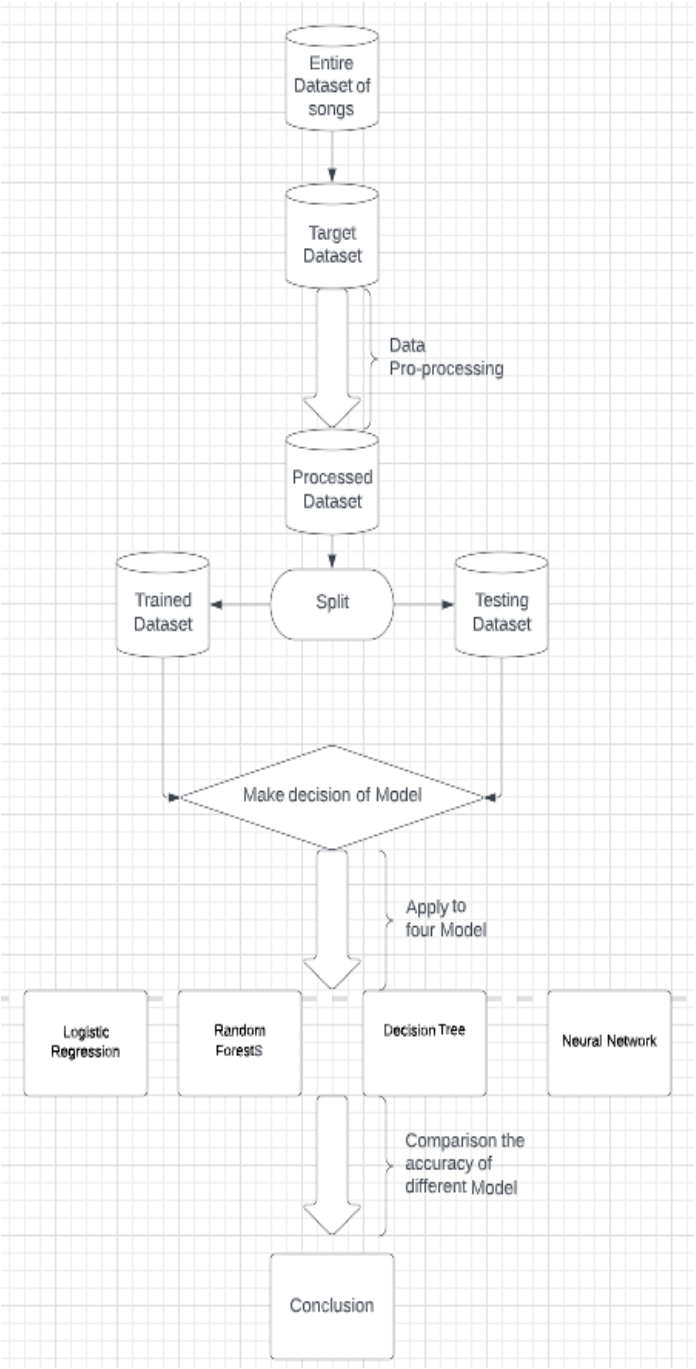

Figure 13: Project Design

RESULTS

Accuracy scores for considered models can be seen below:

Model	*Accuracy Score*
Logistic Regression	10%
Random Forest	30%
Decision Tree (*worst*)	7%
Neural Network (*best*)	88%

DIFFICULTIES FACED

Improving the accuracy score of the various models our team worked on remained the main challenge throughout the process. We combined various preprocessing techniques to enhance the outcomes, such as determining the best correlation among different variables. Overall, we were happy with the results, which show that not every model will work with a particular dataset.

THINGS THAT WORKED

Neural Network Model outperformed the remaining models.

THINGS THAT DIDN'T WORK

Logistic Regression, Random Forest, and Decision Tree Model with low accuracy scores.

IV. DISCUSSION AND CONCLUSION

DECISIONS MADE

During the first discussion, we had two proposals. One is about online shopping research on websites, and the other is about song release year prediction. After voting, we chose the second one.

During the second discussion, we decided which part each person would be responsible for. Each part was discussed in depth to determine our future project's general direction.

We progressed in each section during the third discussion and generally completed the project. But we have not yet integrated each part, nor have we discussed and improved them.

During the fourth discussion, we discussed the PowerPoint's content, divided the presentation content that each group member needed to speak about, and how we should improve the part we were responsible for.

CONCLUSION

We tried many different models and used dimensionality reduction to reduce the variance of the data. After that, we tried upsampling the dataset to reduce the skewness of the data. Then after tuning the hyperparameters of the neural network, we got an accuracy above 80%.
Further, for this dataset, we could classify decades instead of years to have a more robust performance. During the testing phase, some models for the subset of the original dataset noticed a significant improvement in accuracy scores. For instance, if only songs released after the year 2000 are taken into account, the accuracy score of Logistic Regression almost increased by 20%. Overall, it was a great experience demonstrating each step's significance, including preprocessing, dataset cleaning, model selection, exploratory data analysis, etc.

V. TASK DISTRIBUTION

Charanveer Singh: Logistic Regression Model, Decision Tree Model, Random Forest Model, Pre-processing, and EDA

Moxank Prakashbhai Patel: Balancing the dataset and designing, implementing the neural network, and then tuning the hyperparameters of the neural network.

Xinyu He: Exploratory Data Analysis and Reducing the dimensionality, Evaluate the Linear Regression performance.

Hongjin Cheng: Exploratory Data Analysis, finding the relevant subset of features from the original dataset, and evaluating the performance using SVD.

Everyone ended up doing their task.

VI. REFERENCES

1. *raghav1810. (n.d.). Music-analysis-and-year-prediction/release-year-predictorrn-on-msd.ipynb at master · RAGHAV1810/music-analysis-and-year-prediction. GitHub. Retrieved December 1, 2022, from https://github.com/raghav1810/Music-analysis-and-Year-prediction/blob/master/release-year-prediction-on-msd.ipynb*
2. *Importance of feature scaling. scikit. (n.d.). Retrieved December 1, 2022, from https://scikit-learn.org/stable/auto_examples/preprocessing/plot_scaling_importance.html#sphx-glr-auto-examples-preprocessing-plot-scaling-importance-py*
3. *Yiu, T. (2021, September 29). Understanding random forests. Medium. Retrieved December 1, 2022, from https://towardsdatascience.com/understanding-random-forest-58381e0602d2#:~:text=The%20random%20forest%20is%20a,that%20of%20any%20individual%20tree.*
4. *Wikimedia Foundation. (2022, November 30). Artificial Neural Network. Wikipedia. Retrieved December 1, 2022, from https://en.wikipedia.org/wiki/Artificial_neural_network*
5. *Wikimedia Foundation. (2022, October 3). Simple linear Regression. Wikipedia. Retrieved December 1task from https://en.wikipedia.org/wiki/Simple_linear_regression*
6. *Swaminathan, S. (2019, January 18). Logistic Regression - detailed overview. Medium. Retrieved December 1, 2022, from https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc*
7. *Nik. (2022, April 17). Decision tree classifier with Sklearn in python • datagy. datagy. Retrieved December 1, 2022, from https://datagy.io/sklearn-decision-tree-classifier/*
8. *Random Forest classifier using Scikit-learn. GeeksforGeeks. (2022, November 28). Retrieved December 1, 2022, from https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn/*