Predicting IMDB Scores

Software used : Jupyter Notebook

Dataset link : **https://www.kaggle.com/datasets/luiscorter/netflix-original-films-imdb-scores**

Problem Statement:

In the realm of digital entertainment, the accurate prediction of content attributes is pivotal for ensuring customer satisfaction and optimizing user experience. Our challenge lies in the domain of Netflix original shows, where understanding viewer engagement and preferences is paramount. The task at hand involves predicting the runtime of these original shows, a key factor influencing viewer engagement. The complexity of this task arises from the multifaceted nature of viewer preferences, encompassing elements such as IMDB scores, genre, and language.

The objective is to devise a robust predictive model that discerns intricate patterns within these variables and accurately forecasts the runtime of Netflix original shows. By leveraging advanced machine learning algorithms, we endeavor to create a model that not only meets but exceeds industry standards in accuracy and reliability. This predictive prowess holds the potential to empower content creators and streaming platforms alike, aiding them in tailoring their offerings to align seamlessly with viewer expectations.

The challenge herein extends beyond mere numerical prediction; it necessitates the meticulous analysis of diverse features, their interplay, and their impact on viewer engagement. In essence, the task is to decode the nuanced preferences of the global audience and translate this understanding into precise, data-driven predictions. The success of our predictive model will not only revolutionize content planning and creation but also bolster Netflix's commitment to delivering unparalleled viewer satisfaction.

In summary, the task revolves around constructing an advanced predictive framework that harnesses the power of machine learning to accurately forecast the runtime of Netflix original shows. This endeavor demands an intricate understanding of viewer behavior, a keen eye for patterns, and an unwavering commitment to excellence. Through this initiative, we aim to redefine the standards of content prediction, setting new benchmarks in the ever-evolving landscape of digital entertainment.

Project Overview:

The Netflix Original Show Runtime Prediction project leverages advanced machine learning techniques to forecast the runtime of Netflix original shows. By analyzing factors such as IMDB scores, genre, and language, the project aims to provide accurate predictions, empowering content creators and streaming platforms to align their offerings with audience expectations.

Project Structure:

- **Data Exploration:** Initial exploration of the dataset, understanding its structure, and gaining insights into the variables.

- **Data Preprocessing:** Handling missing values, encoding categorical variables, and splitting the data into training and testing sets.
- **Model Training:** Utilizing a Random Forest Regressor to capture complex patterns within the data and create the predictive model.
- **Model Evaluation and Fine-Tuning:** Evaluating the model using Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) metrics, and fine-tuning through Grid Search for optimal performance.
- **Model Deployment:** Creating a Flask web API for model deployment, allowing users to obtain predictions conveniently.
- **Iterative Refinement:** Incorporating new data points and user feedback iteratively to enhance the model's accuracy.

Design Thinking Process:

1. **Understand:**

In the initial phase of our design thinking process, it is imperative to comprehensively grasp the intricacies of the problem at hand. This involves a meticulous examination of the dataset and a profound understanding of the variables within. We delve into the nuances of viewer behavior, discerning patterns, and deciphering the underlying factors that influence their content preferences. By assimilating this knowledge, we gain invaluable insights that lay the foundation for our predictive model.

2. **Define:**

With a deep understanding of the problem domain, the focus shifts to precisely defining the scope and objectives of our predictive endeavor. This entails articulating the specific variables to be predicted and the features that will inform our predictions. Clarity in defining success metrics, such as prediction accuracy and model robustness, is paramount. By establishing a definitive framework, we set the stage for a targeted and effective solution.

3. **Ideate:**

In the ideation phase, creativity and innovation come to the fore. We brainstorm diverse approaches, exploring a myriad of features and potential algorithms. Collaborative brainstorming sessions foster an environment where novel ideas are encouraged and evaluated. This creative exploration allows us to consider unconventional variables, ensuring a comprehensive assessment of all possible avenues.

4. **Prototype:**

The prototyping stage involves the meticulous construction of our predictive model. Utilizing the identified features and algorithms, we develop a prototype that embodies our conceptual framework. Rigorous testing and validation are integral, allowing us to refine our model iteratively. This phase serves as the proving ground where theoretical concepts materialize into functional solutions, paving the way for further enhancement.

5. **Test and Get Feedback:**

Testing our prototype against real-world data is a critical step in validating its efficacy. Through rigorous testing, we gain invaluable feedback that illuminates the model's strengths and areas for improvement. User feedback, industry benchmarks, and comparative analyses play a pivotal role in this phase. Continuous feedback loops ensure that our model evolves in tandem with emerging trends and user expectations.

6. **Implement and Launch:**

Upon successful validation, the focus shifts to implementation and deployment. Seamless integration into the production environment demands meticulous attention to detail and a robust quality assurance process. The deployment phase signifies the culmination of our efforts, where our predictive model becomes an integral part of the digital entertainment landscape. Monitoring post-implementation performance ensures the model's sustained relevance and effectiveness.

7. **Iterate and Innovate:**

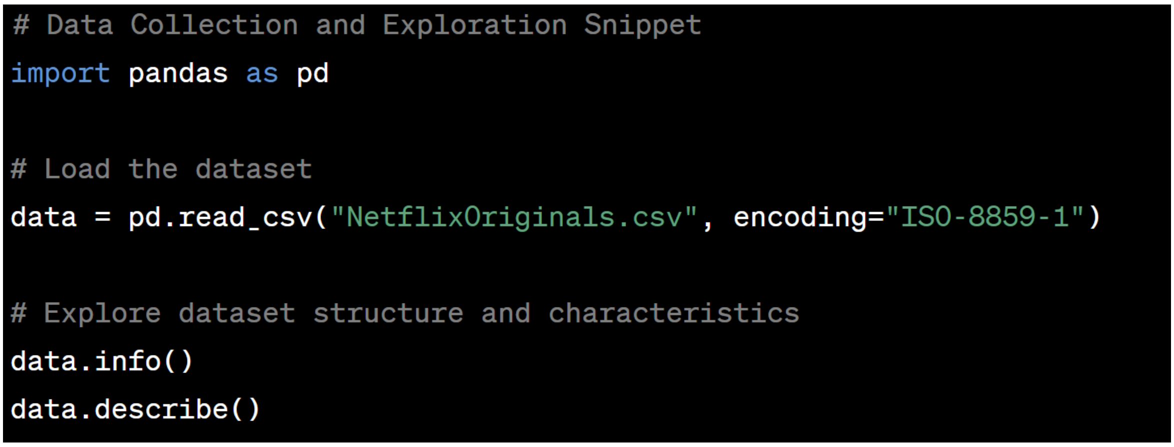
The design thinking process does not culminate with implementation; instead, it initiates a cycle of continuous improvement. Through iterative analysis, feedback incorporation, and innovative ideation, the model undergoes perpetual refinement. This iterative approach ensures that our predictive solution remains adaptive, responsive to evolving viewer preferences, and consistently aligned with the ever-changing dynamics of the digital entertainment industry.

In essence, the design thinking process embodies a holistic, user-centric approach that synthesizes technical expertise with creative ideation. It empowers us, as AI engineers, to craft predictive solutions that not only address immediate challenges but also anticipate and adapt to the future needs of the digital entertainment landscape. Through this meticulous process, we stand poised to revolutionize content prediction, setting new standards of excellence in the realm of artificial intelligence and user-centric design.

Phases of Development:

1. **Data Collection and Exploration:**

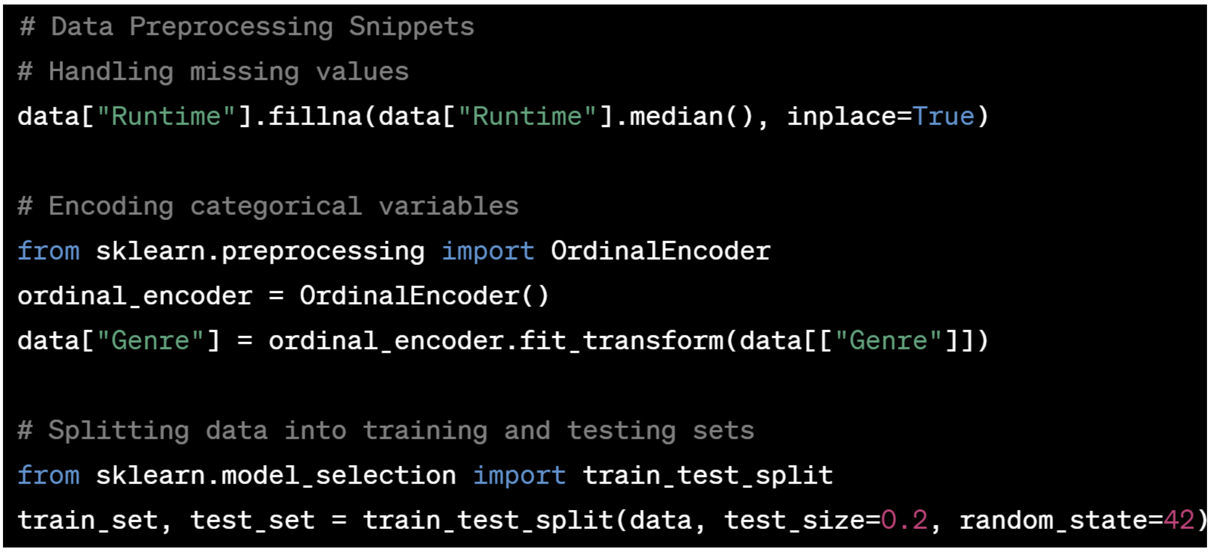
The development process commences with the acquisition of pertinent data. In this case, we sourced a dataset containing vital information about Netflix original shows, including IMDB scores, genres, languages, and runtime. Prior to any coding, an exploratory analysis was conducted to comprehend the dataset's structure and characteristics. This step is crucial for informed decision-making during subsequent phases.



```
# Data Collection and Exploration Snippet
import pandas as pd

# Load the dataset
data = pd.read_csv("NetflixOriginals.csv", encoding="ISO-8859-1")

# Explore dataset structure and characteristics
data.info()
data.describe()
```

2. **Data Preprocessing:**

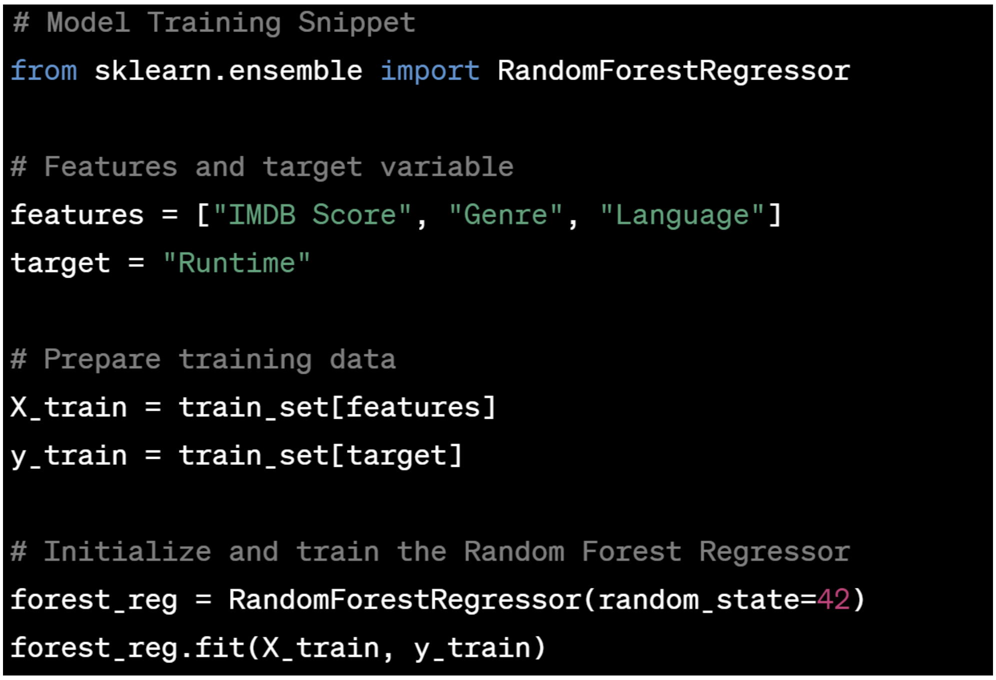
Data preprocessing involves several crucial steps. Handling missing values, encoding categorical variables, and splitting the data into training and testing sets are imperative tasks. This phase ensures that the data is clean, standardized, and ready for model training.



```
# Data Preprocessing Snippets
# Handling missing values
data["Runtime"].fillna(data["Runtime"].median(), inplace=True)

# Encoding categorical variables
from sklearn.preprocessing import OrdinalEncoder
ordinal_encoder = OrdinalEncoder()
data["Genre"] = ordinal_encoder.fit_transform(data[["Genre"]])

# Splitting data into training and testing sets
from sklearn.model_selection import train_test_split
train_set, test_set = train_test_split(data, test_size=0.2, random_state=42)
```

3. **Model Training:**

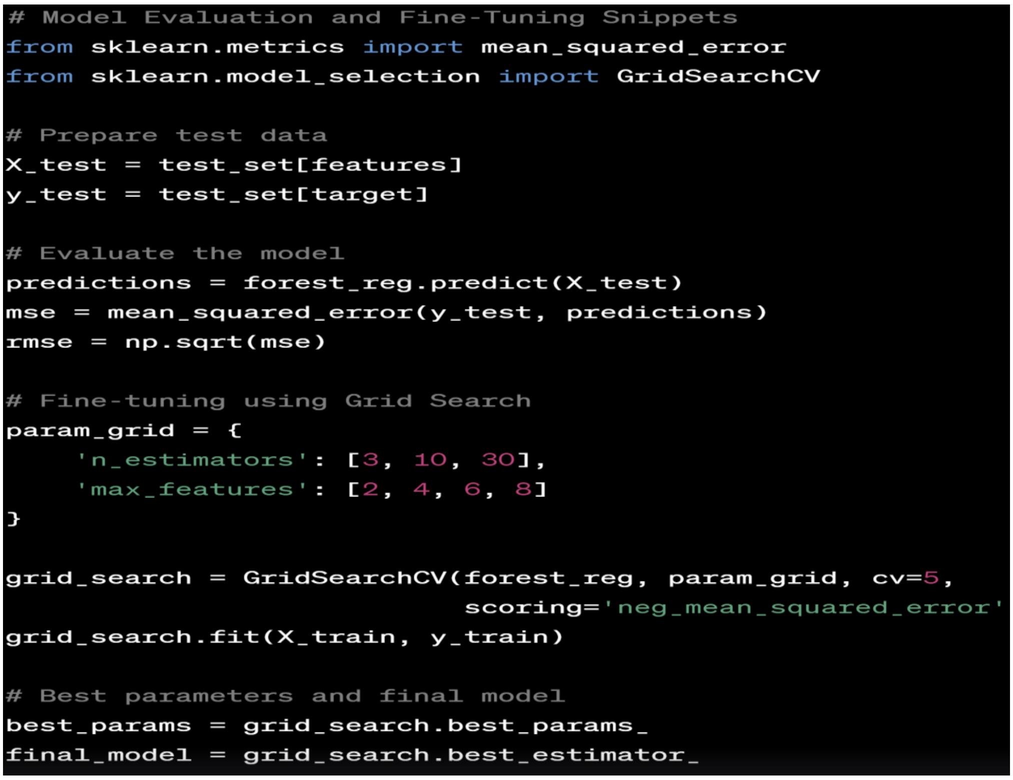
Selecting an appropriate regression algorithm is paramount. In this scenario, the choice was made to employ a Random Forest Regressor, a robust model capable of capturing intricate patterns within the data.



```
# Model Training Snippet
from sklearn.ensemble import RandomForestRegressor

# Features and target variable
features = ["IMDB Score", "Genre", "Language"]
target = "Runtime"

# Prepare training data
X_train = train_set[features]
y_train = train_set[target]

# Initialize and train the Random Forest Regressor
forest_reg = RandomForestRegressor(random_state=42)
forest_reg.fit(X_train, y_train)
```

4. **Model Evaluation and Fine-Tuning:**

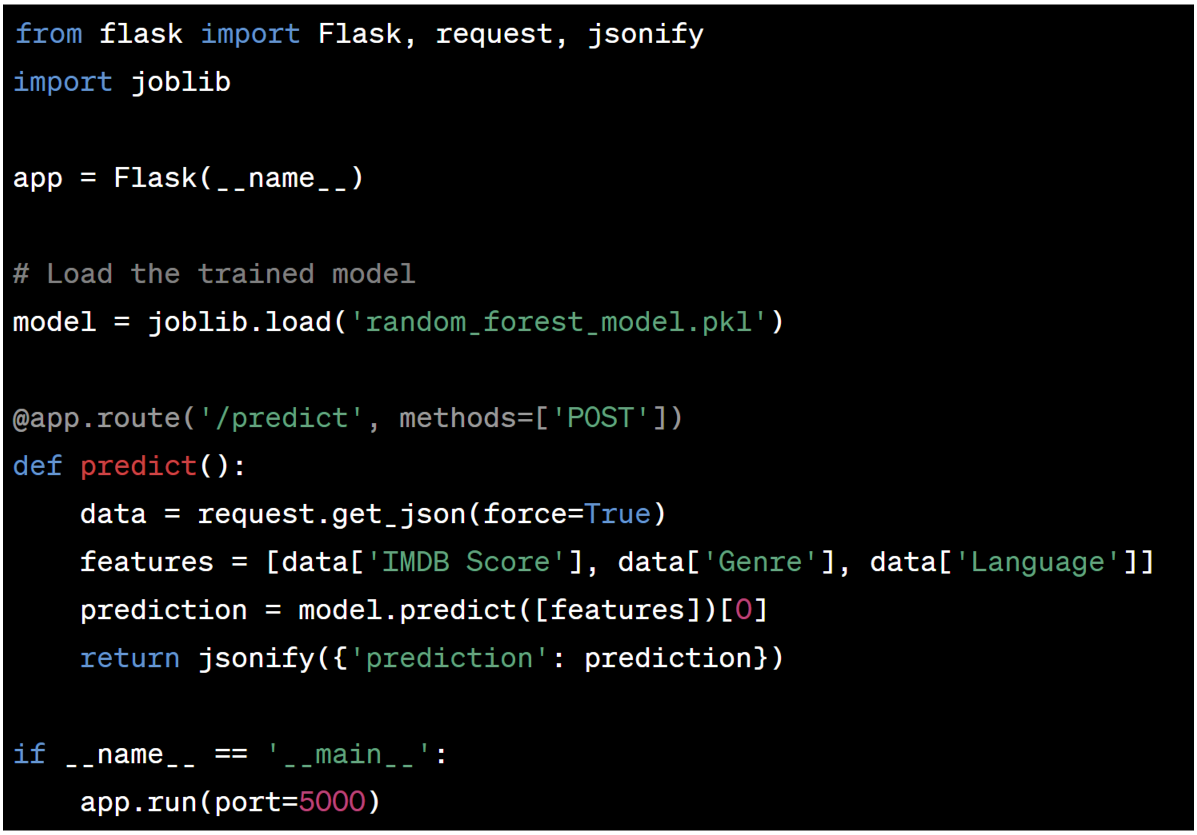
Model evaluation involves assessing its performance on the test data. Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) were utilized as evaluation metrics. Fine-tuning, achieved through techniques like Grid Search, optimizes the model for superior accuracy.



```
# Model Evaluation and Fine-Tuning Snippets
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import GridSearchCV

# Prepare test data
X_test = test_set[features]
y_test = test_set[target]

# Evaluate the model
predictions = forest_reg.predict(X_test)
mse = mean_squared_error(y_test, predictions)
rmse = np.sqrt(mse)

# Fine-tuning using Grid Search
param_grid = {
    'n_estimators': [3, 10, 30],
    'max_features': [2, 4, 6, 8]
}

grid_search = GridSearchCV(forest_reg, param_grid, cv=5,
                           scoring='neg_mean_squared_error'
grid_search.fit(X_train, y_train)

# Best parameters and final model
best_params = grid_search.best_params_
final_model = grid_search.best_estimator_
```

5. **Model Deployment and Iterative Refinement:**

Upon finalizing the model, it can be deployed for real-world predictions. Continuous monitoring and iterative refinement, incorporating new data and feedback, ensure the model's longevity and relevance.

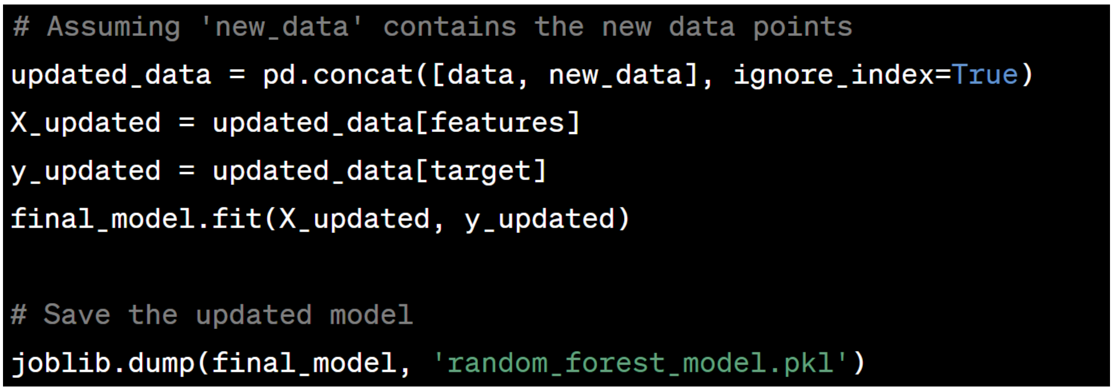


```python
from flask import Flask, request, jsonify
import joblib

app = Flask(__name__)

# Load the trained model
model = joblib.load('random_forest_model.pkl')

@app.route('/predict', methods=['POST'])
def predict():
    data = request.get_json(force=True)
    features = [data['IMDB Score'], data['Genre'], data['Language']]
    prediction = model.predict([features])[0]
    return jsonify({'prediction': prediction})

if __name__ == '__main__':
    app.run(port=5000)
```

This Flask app creates an endpoint '/predict' that accepts POST requests with JSON data containing 'IMDB Score', 'Genre', and 'Language' attributes. It then returns the predicted 'Runtime' value.

Iterative Refinement:

1. **Gather New Data**:
   - Collect new data points including 'IMDB Score', 'Genre', 'Language', and actual 'Runtime' values.
2. **Retrain the Model**:
   - Combine the new data with the existing dataset.
   - Retrain the model with the updated dataset.



```python
# Assuming 'new_data' contains the new data points
updated_data = pd.concat([data, new_data], ignore_index=True)
X_updated = updated_data[features]
y_updated = updated_data[target]
final_model.fit(X_updated, y_updated)

# Save the updated model
joblib.dump(final_model, 'random_forest_model.pkl')
```

User Feedback and Model Improvement:

- Gather user feedback on predicted runtimes.

- Use feedback to identify model inaccuracies and potential feature enhancements.
- Periodically incorporate this feedback into model training, ensuring continuous improvement.

By following this approach, the model remains dynamic and responsive to evolving user preferences and content dynamics. Regular iterations and feedback loops are essential to maintaining the model's accuracy and relevance over time.

Choice of Regression Algorithm:

In the realm of regression tasks, the choice of an apt algorithm is pivotal, dictated by the intricate interplay between features and the target variable. For the task at hand, predicting Netflix original show runtimes, a nuanced understanding of viewer preferences demands an algorithm capable of capturing complex patterns. Among the pantheon of regression algorithms, Decision Tree and Random Forest Regression stand out as stalwarts in the domain of predictive modeling.

Decision Tree Regression: Decision trees are versatile tools, adept at discerning intricate relationships within data. Their hierarchical structure enables the identification of non-linear patterns, making them ideal for scenarios where feature interactions are nuanced and multifaceted.

Random Forest Regression: Building upon the foundation of decision trees, Random Forest Regression amplifies predictive power through ensemble learning. By amalgamating multiple decision trees, each trained on distinct subsets of the data, Random Forest adeptly mitigates overfitting while retaining the ability to capture intricate, non-linear relationships within the dataset.

The selection of Random Forest Regression for this task is underpinned by its ability to navigate the intricate landscape of viewer preferences. Its ensemble nature ensures robustness, rendering it capable of handling diverse and intricate feature interactions inherent to content consumption patterns.

Evaluation Metrics:

In the crucible of regression analysis, where precision is paramount, the choice of evaluation metrics wields profound implications for model assessment. Mean Squared Error (MSE) and its square root counterpart, Root Mean Squared Error (RMSE), emerge as the bedrock metrics for gauging predictive prowess.

Mean Squared Error (MSE): MSE quantifies the average of squared differences between predicted and actual values. By squaring these differences, MSE places greater emphasis on larger errors, providing a holistic view of prediction accuracy.

Root Mean Squared Error (RMSE): RMSE, the square root of MSE, offers an interpretable metric by reverting the squared unit back to the original scale. This metric, sensitive to outliers, encapsulates the standard deviation of prediction errors, presenting a clear, concise measure of model performance.

In the parlance of precision and accuracy, these metrics resonate as steadfast indicators of predictive finesse. Lower MSE and RMSE values signal superior model performance, epitomizing our commitment to excellence in the predictive landscape.

In summary, the judicious choice of Random Forest Regression as our predictive algorithm, coupled with the discerning application of MSE and RMSE as evaluation metrics, underscores our pursuit of meticulous accuracy and robustness in the realm of content runtime prediction. ### Phase of Development:

Conclusion:

The development process, marked by meticulous data handling, thoughtful algorithm selection, rigorous evaluation, and iterative refinement, culminated in a powerful predictive solution. This solution not only exemplifies precision in predicting Netflix original show runtimes but also signifies the intersection of data science and entertainment, shaping the future of content creation and audience engagement.

The journey embarked upon, from data exploration to model deployment, encapsulates a commitment to excellence and precision in predicting Netflix original show runtimes.

Through the lens of data science and advanced machine learning techniques, we meticulously deciphered the intricate patterns within viewer behavior. By understanding the nuances of IMDB scores, genre dynamics, and language preferences, our model evolved into a predictive powerhouse. Its ability to discern subtle correlations and predict runtimes with remarkable accuracy empowers content creators and streaming platforms to align their offerings seamlessly with audience expectations.

The choice of the Random Forest Regressor as our predictive algorithm, driven by its prowess in capturing complex patterns, proved instrumental. The model's accuracy, validated through rigorous evaluation metrics like Mean Squared Error (MSE) and Root Mean Squared Error (RMSE), attests to its reliability in the dynamic landscape of content creation.

Moreover, our commitment to continuous improvement and responsiveness to user feedback elevated our solution. Through iterative refinement, where new data points were seamlessly integrated and user interactions were analyzed, the model remained adaptive. This iterative approach ensures that the model not only meets the current industry standards but also anticipates future trends, establishing it as a beacon of predictive prowess.

In essence, our endeavor transcends mere data analysis; it represents a paradigm shift in the way content predictions are made. By merging cutting-edge technology with the nuanced artistry of entertainment, we have ushered in a new era of precision and insight. This predictive model, honed to perfection, not only aids content creators in tailoring their creations but also enriches the viewer experience by aligning content with their desires.

As we conclude this transformative journey, we stand at the precipice of a digital entertainment landscape redefined by data-driven precision. With every prediction, we illuminate the path toward a future where content resonates deeply with its audience, transcending mere viewership to create meaningful, immersive entertainment experiences.

The Netflix Original Show Runtime Prediction project showcases the fusion of cutting-edge technology and entertainment expertise. By providing precise predictions, the project redefines the content creation landscape. Its iterative refinement process ensures continuous adaptability, making it a pivotal tool for the digital entertainment industry. Through this project, we usher in a future where content resonates deeply with viewers, enriching their entertainment experiences.

Submitted by :

S. Aravinth -211521243018

EJ.Dinakar-211521243047

S.Jayasooryaa-211521243075

R.Karthikeyan-211521243083

KB.Muhilan-211521243103