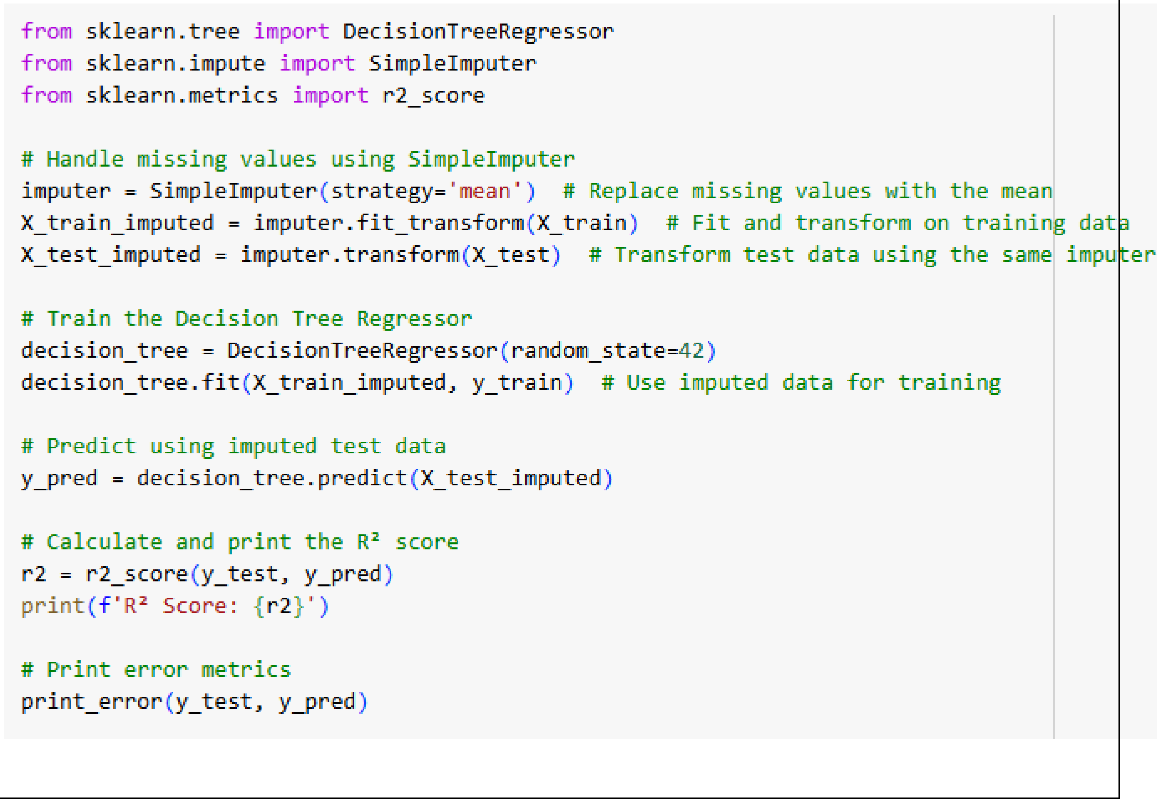
Model Optimization and Tuning Phase Template

Date	15 July 2024
Team ID	Team-740680
Project Title	View count visionary:a data driven approach to forcasting youtube videos views
Maximum Marks	10 Marks

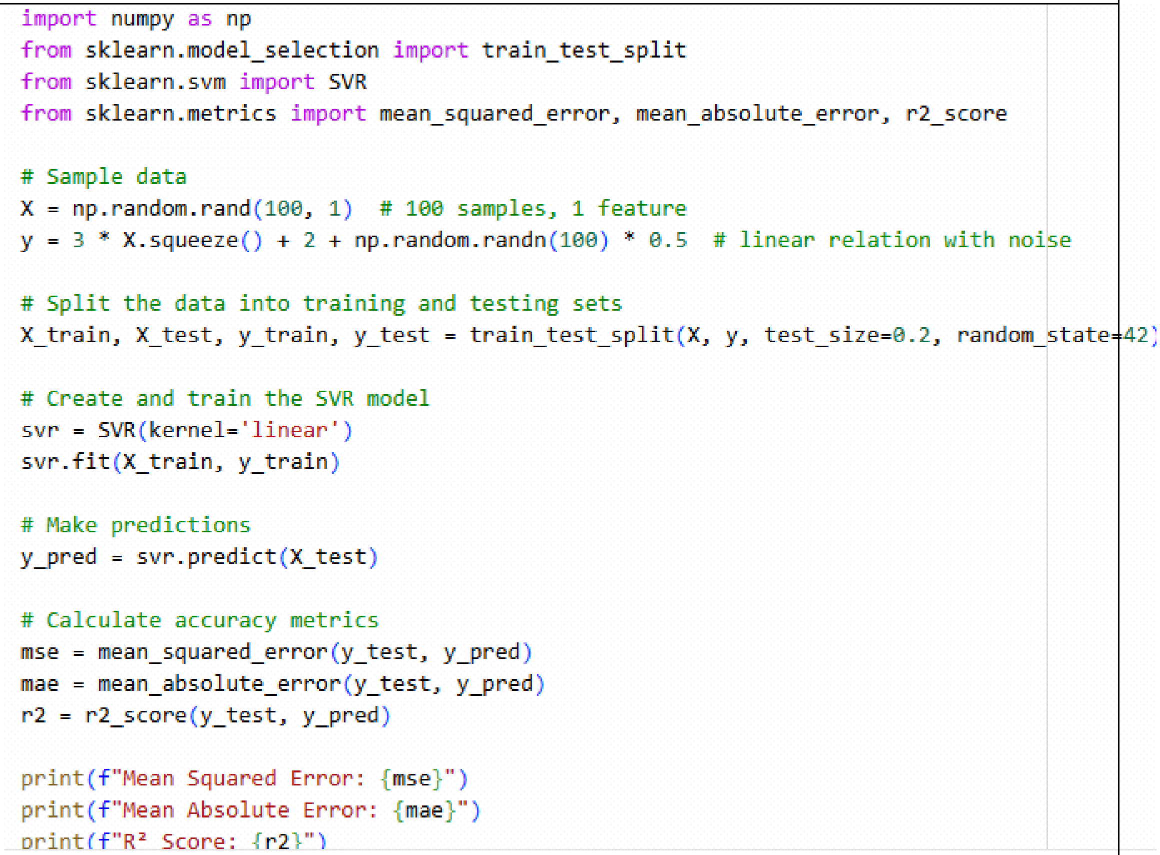
Model Optimization and Tuning Phase

This template provides a comprehensive approach to documenting the model optimization and tuning phase for the "Visionary" project Brief description of the project and the importance of model optimization and tuning in improving prediction accuracy.Outline the objective of the model optimization and tuning phase, such as improving model performance, reducing error rates, and enhancing generalizability.

Hyperparameter Tuning Documentation (8 Marks):

Model	Tuned Hyperparameters
Decision tree	

```python
from sklearn.tree import DecisionTreeRegressor
from sklearn.impute import SimpleImputer
from sklearn.metrics import r2_score

# Handle missing values using SimpleImputer
imputer = SimpleImputer(strategy='mean')  # Replace missing values with the mean
X_train_imputed = imputer.fit_transform(X_train)  # Fit and transform on training data
X_test_imputed = imputer.transform(X_test)  # Transform test data using the same imputer

# Train the Decision Tree Regressor
decision_tree = DecisionTreeRegressor(random_state=42)
decision_tree.fit(X_train_imputed, y_train)  # Use imputed data for training

# Predict using imputed test data
y_pred = decision_tree.predict(X_test_imputed)

# Calculate and print the R² score
r2 = r2_score(y_test, y_pred)
print(f'R² Score: {r2}')

# Print error metrics
print_error(y_test, y_pred)
```

<table>
<tr><th></th><th></th></tr>
<tr><td>Random Forest</td><td>
<pre><code>import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

# Sample data
X = np.random.rand(100, 1) # 100 samples, 1 feature
y = 3 * X.squeeze() + 2 + np.random.randn(100) * 0.5 # linear relation with noise

Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create and train the SVR model
svr = SVR(kernel='linear')
svr.fit(X_train, y_train)

# Make predictions
y_pred = svr.predict(X_test)

# Calculate accuracy metrics
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse}")
print(f"Mean Absolute Error: {mae}")
print(f"R² Score: {r2}")</code></pre></td></tr>
</table>

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Decision Tree	A supervised learning algorithm that splits data into subsets based on input features, creating a tree-like model of decisions.Decision Trees are straightforward to understand and visualize, making it easier to explain the model's predictions to stakeholders who might not have a technical background.The tree structure provides a clear decision path,

	showing how different features influence the forecast of YouTube video views.