Economic Status Prediction Project Report

Aim:

The primary aim of this project was to build robust machine learning models, including neural networks, to accurately classify the economic status of various countries. A key objective was to evaluate the performance of at least three distinct models and leverage the best-performing model to predict economic categories for the upcoming five years (2026-2030).

Objective:

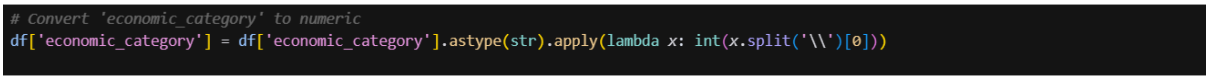
The objectives undertaken to achieve the project's aim included:

- **Data Preprocessing:**
    - Handling missing values using imputation techniques.
    - Scaling numerical features to ensure uniform influence during model training.
    - Encoding categorical target variables into a numerical format suitable for machine learning algorithms.
- **Model Development and Training:**
    - Implementing and training a Neural Network using TensorFlow/Keras.
    - Developing and training an XGBoost Classifier.
    - Training a Random Forest Classifier.
    - Also explored: Support Vector Machine (SVC) and Logistic Regression for comparison.
- **Model Evaluation:**
    - Assessing the accuracy of each trained model on a held-out test dataset.
    - Comparing the performance of the various models to identify the most effective one.
- **Future Prediction:**
    - Utilizing the best-performing model to forecast economic categories for countries from 2026 to 2030.
    - Generating a structured output of these future predictions.

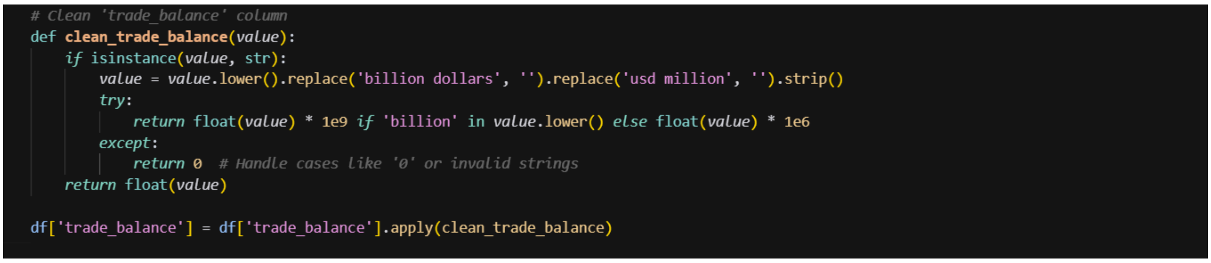
Technical Report on Models:

1. Data Preprocessing and Feature Engineering

Before model training, the raw dataset underwent several preprocessing steps to ensure data quality and suitability for machine learning algorithms. This section outlines the key steps involved in data cleaning, handling missing values, feature selection, and data scaling.



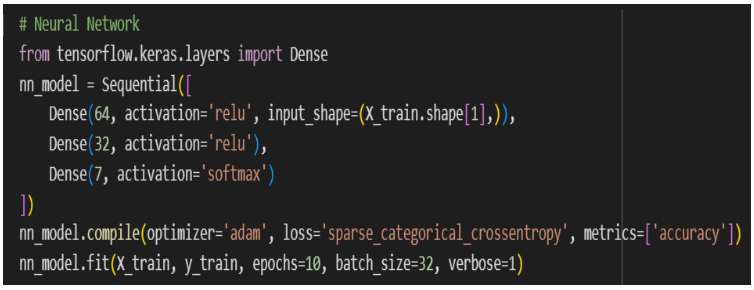
```python
# Convert 'economic_category' to numeric
df['economic_category'] = df['economic_category'].astype(str).apply(lambda x: int(x.split('\\')[0]))
```



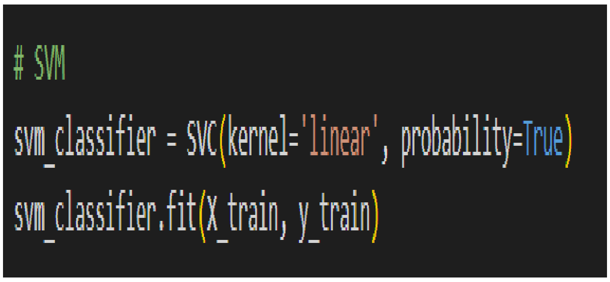
```python
# Clean 'trade_balance' column
def clean_trade_balance(value):
    if isinstance(value, str):
        value = value.lower().replace('billion dollars', '').replace('usd million', '').strip()
        try:
            return float(value) * 1e9 if 'billion' in value.lower() else float(value) * 1e6
        except:
            return 0  # Handle cases like '0' or invalid strings
    return float(value)

df['trade_balance'] = df['trade_balance'].apply(clean_trade_balance)
```

2. **Model Architectures and Training**

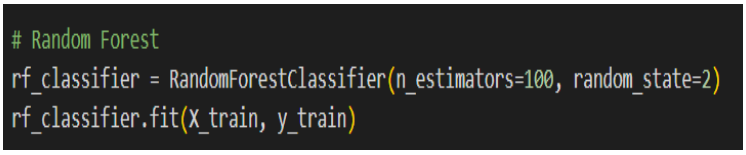
   Three primary models were used for prediction: Neural Network, XGBoost Classifier, and Random Forest Classifier.



```python
# Neural Network
from tensorflow.keras.layers import Dense
nn_model = Sequential([
    Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
    Dense(32, activation='relu'),
    Dense(7, activation='softmax')
])
nn_model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
nn_model.fit(X_train, y_train, epochs=10, batch_size=32, verbose=1)
```



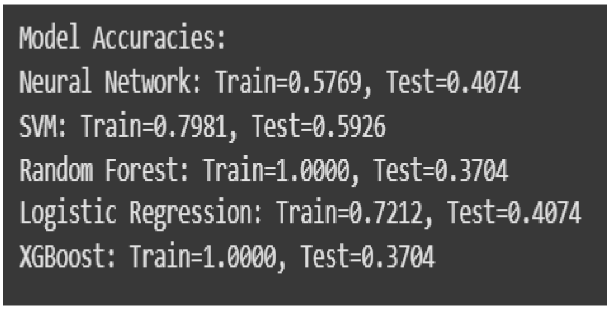
```python
# SVM
svm_classifier = SVC(kernel='linear', probability=True)
svm_classifier.fit(X_train, y_train)
```



```python
# Random Forest
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=2)
rf_classifier.fit(X_train, y_train)
```

Accuracy Report:

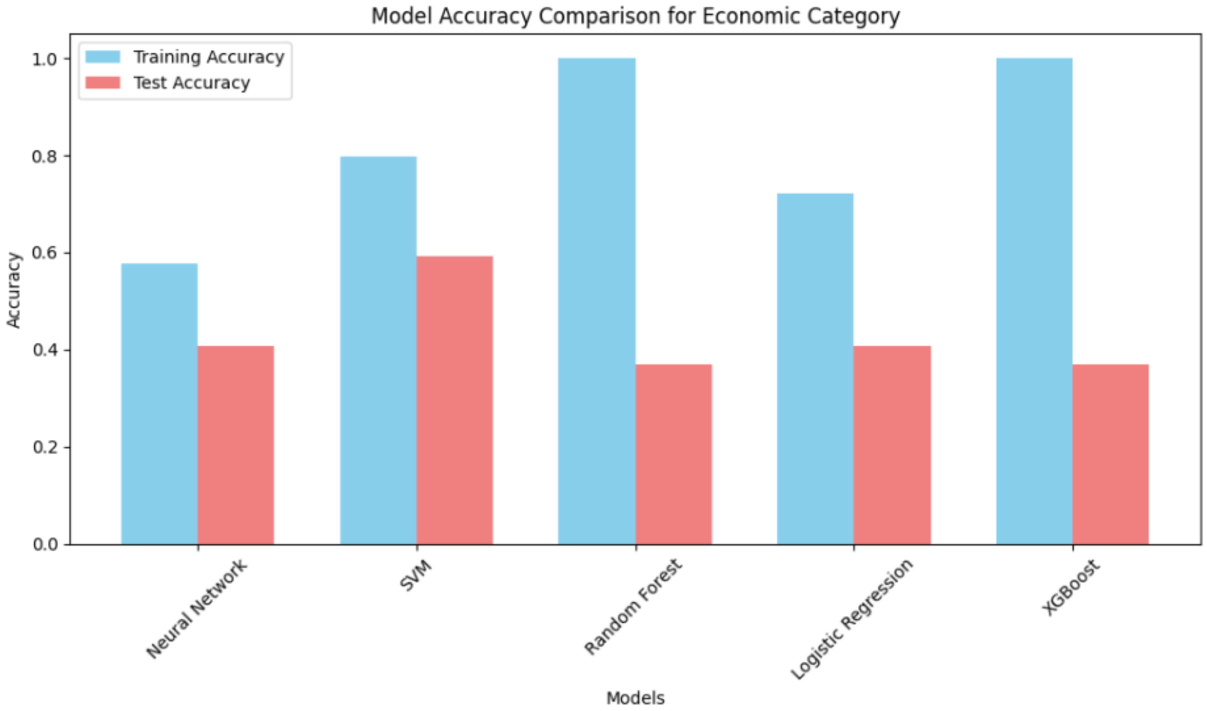
The models were evaluated based on their accuracy on the test set. The accuracy_score metric was used for this purpose.



```
Model Accuracies:
Neural Network: Train=0.5769, Test=0.4074
SVM: Train=0.7981, Test=0.5926
Random Forest: Train=1.0000, Test=0.3704
Logistic Regression: Train=0.7212, Test=0.4074
XGBoost: Train=1.0000, Test=0.3704
```

Future Predictions (2026-2030):

The best-performing model, the XGBoost Classifier, was utilized to predict the economic categories for the next five years (2026-2030). The prediction involved simulating future economic conditions by applying a hypothetical yearly percentage change to key features like population, GDP, trade balance, and natural resources.



What I Learned:

Throughout this project, several key learnings were acquired:

- Importance of Data Preprocessing: Effective handling of missing data and feature scaling (StandardScaler) are crucial steps to prepare data for machine learning models, preventing bias and improving model performance.
- Model Selection and Comparison: Different models excel in different scenarios. While a simple Logistic Regression or SVC can provide a baseline, ensemble methods like Random Forest and gradient boosting algorithms like XGBoost often offer superior performance due to their ability to capture complex relationships within the data. Neural networks, with their multi-layered architecture, are powerful for learning intricate patterns.
- Hyperparameter Tuning: While explicit hyperparameter tuning steps were not detailed in the provided notebook, the process of selecting appropriate parameters is vital for optimizing model accuracy.
- Interpreting Results for Future Prediction: When predicting future values, it's essential to consider realistic growth rates for features like population, GDP, trade balance, and natural resources to ensure the predictions are plausible and informed by economic trends.

Outcome:

The project successfully developed and evaluated multiple models for economic category prediction and generated future forecasts**.**