Predictive Analytics for Netflix: LSTM, CNN, and Random Forest Models

Exploring LSTM, CNN, and Random Forest Models for Stock Price & Subscriber Growth Predictions

M

Maryam Khattak

Presenter

Project Overview

Focus:

Building a real-time AI-based system for Netflix stock price prediction and regional subscriber growth forecasting.

Models:

LSTM and CNN for stock price forecasting.
Random Forest Regressor (RFR) for predicting subscriber growth.

Visualization:

Power BI for interactive and real-time stock and subscriber growth insights.

Project Objectives & Research Questions

Stock Price Prediction

Create LSTM and CNN models to predict real-time stock price of Netflix Inc. (NFLX) using historical data obtained from Yahoo Finance and compare their performance for accuracy and reliability.

Subscriber Growth Rate Prediction

Implement a machine learning model as Random Forest Regressor, to predict subscriber growth in each region, and examine regional factors that influence subscriber trends to support strategic decision-making.

Visualization with Power BI

Integrate Power BI to visualize historical stock price movements, predicted subscriber growth rates, and comparative analysis, providing stakeholders with interactive dashboards to explore data insights and making informed decisions.

Research Questions

Can AI models accurately predict Netflix stock price movements?
How can regional subscriber growth be forecasted using machine learning techniques?

METHODOLOGY OVERVIEW

Methodology and Data Sources

Overview of the Approach and Data Utilization

CRISP-DM Methodology

Utilizes a structured framework comprising six phases: business understanding, data understanding, data preparation, modeling, evaluation, and deployment.

Stock Price Data

Incorporates Netflix stock price data sourced from Yahoo Finance through the yFinance API for accurate financial analysis.

Subscriber Growth Insights

Analyzes subscriber growth using the Netflix Userbase dataset in CSV format to forecast user trends effectively.

Integration of Data Sources

Applies distinct modeling techniques for each dataset to enhance predictive accuracy and provide a comprehensive analysis of Netflix's performance.

MODELING TECHNIQUES

Model Development and Training

Key Techniques for AI-Powered Forecasting Models

Development Tools

Utilizes Google Colab, Python, TensorFlow, Keras, and scikit-learn for streamlined model development.

Random Forest Regressor (RFR)

Handles high-dimensional data efficiently for predicting subscriber growth in a complex environment.

Long Short-Term Memory (LSTM)

Ideal for capturing temporal dependencies in stock prices, enhancing forecasting accuracy.

Convolutional Neural Network (CNN)

Effective in detecting patterns within time-series data, improving model responsiveness.

LSTM OVERVIEW

LSTM Model Architecture

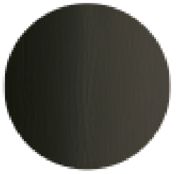
Understanding the Structure for AI-Powered Forecasting

Input Layer

Utilizes normalized sequences to ensure data consistency.

Dense Layer

Final Dense layer outputs predictions for stock prices.



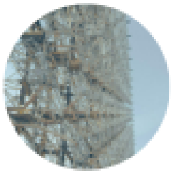
Training Process

Configured with a batch size of 1 and trained for 1 epoch.

```python
# Import the necessary libraries
from keras.models import Sequential
from keras.layers import LSTM, Dense, Input

# 3.1 Build & Train LSTM model
lstm_model = Sequential()
# Add Input layer
lstm_model.add(Input(shape=(time_step, 1)))
# Add LSTM layers
lstm_model.add(LSTM(50, return_sequences=True))
lstm_model.add(LSTM(50, return_sequences=False))
# Add Dense layers
lstm_model.add(Dense(25))
lstm_model.add(Dense(1))
# Compile the model
lstm_model.compile(optimizer='adam', loss='mean_squared_error')

# Train the model
lstm_model.fit(X, y, batch_size=1, epochs=1)
```



LSTM Layers

Two LSTM layers with 50 units each for capturing temporal dependencies.

Compilation

Employs Adam optimizer with MSE loss for efficient training.

CNN ARCHITECTURE

CNN Model Architecture

Key Components of a Convolutional Neural Network for Prediction

Input Layer

Normalized input sequence prepares data for processing.

Pooling Layer

MaxPooling1D reduces dimensionality for efficient computation.

Dense Layers

Final dense layer outputs predictions for stock price forecasting.

```python
# Import the necessary libraries
from keras.models import Sequential
from keras.layers import Conv1D, MaxPooling1D, Flatten, Dense, Input

# 3.2 Build & Train CNN model
cnn_model = Sequential()
# Add Input layer
cnn_model.add(Input(shape=(time_step, 1)))
# Add Convolutional layers
cnn_model.add(Conv1D(filters=64, kernel_size=2, activation='relu'))
cnn_model.add(MaxPooling1D(pool_size=2))
# Add Flatten and Dense layers
cnn_model.add(Flatten())
cnn_model.add(Dense(50, activation='relu'))
cnn_model.add(Dense(1))
# Compile the model
cnn_model.compile(optimizer='adam', loss='mean_squared_error')

# Train the model
cnn_model.fit(X, y, batch_size=1, epochs=1)
```

Convolutional Layers

Utilizes Conv1D with 64 filters and kernel size 2 for feature extraction.

Flatten Layer

Transforms 2D features into a 1D vector for the dense layers.

Compilation and Training

Uses Adam optimizer and MSE loss; trained with batch size of 1 for 1 epoch.

MODEL OVERVIEW

Random Forest Regressor Model

Leveraging Ensemble Methods for Predictive Analytics

```
#3.1 Train Random Forest Regressor
from sklearn.ensemble import RandomForestRegressor

# Initialize and train the Random Forest Regressor
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)

rf_model.fit(X_train, y_train)
```

01 Ensemble Method

Utilizes 100 decision trees to improve prediction accuracy and reduce overfitting.

02 Training with Data

Employs subscriber growth data, ensuring robust model training via split datasets.

03 Performance Evaluation

Assesses model effectiveness using R-squared (R^2) and RMSE metrics for accuracy.

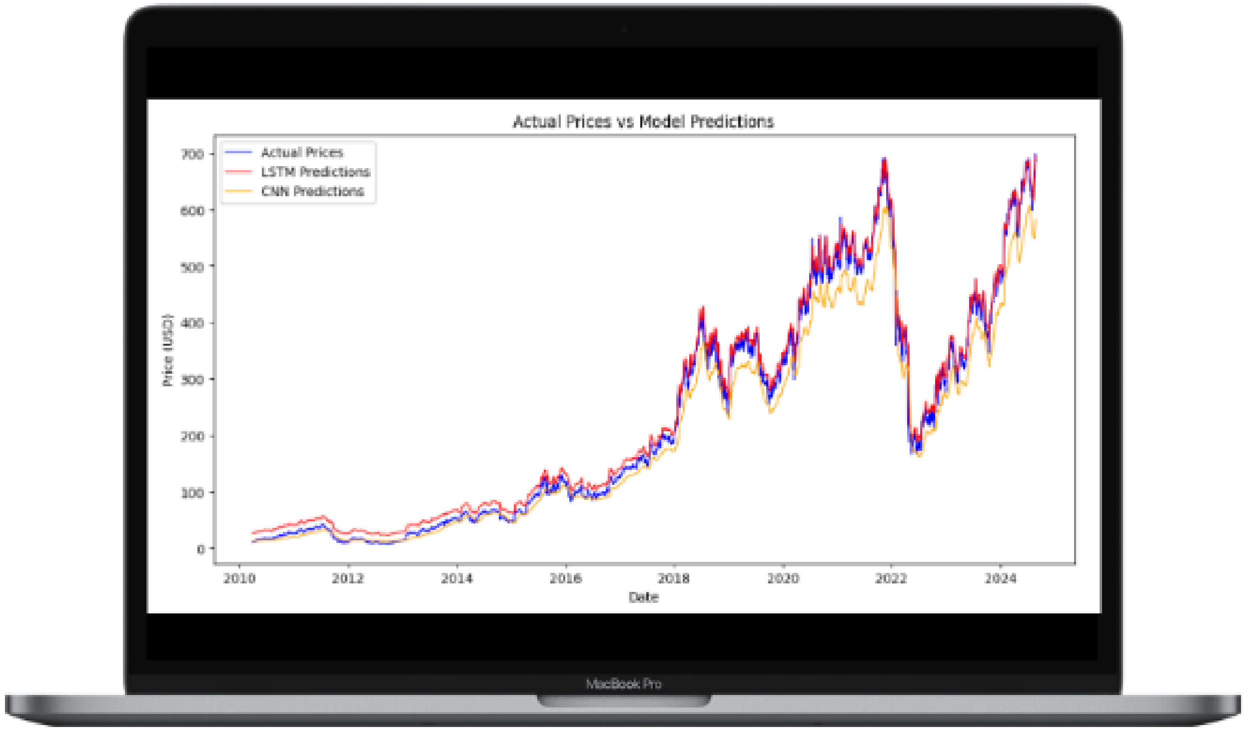
04 Predictive Insights

Delivers insights into subscriber trends, enabling strategic decision-making.

MODEL COMPARISON

Evaluation and Results: Netflix Stock Price Prediction

Comparative Analysis of LSTM and CNN Model Performance



LSTM Model Performance

01

Mean Squared Error (MSE): 111.74

Root Mean Squared Error (RMSE): 10.57

Outperforms CNN in accuracy

Better suited for time series data

CNN Model Performance

02

Mean Squared Error (MSE): 430.90

Root Mean Squared Error (RMSE): 20.76

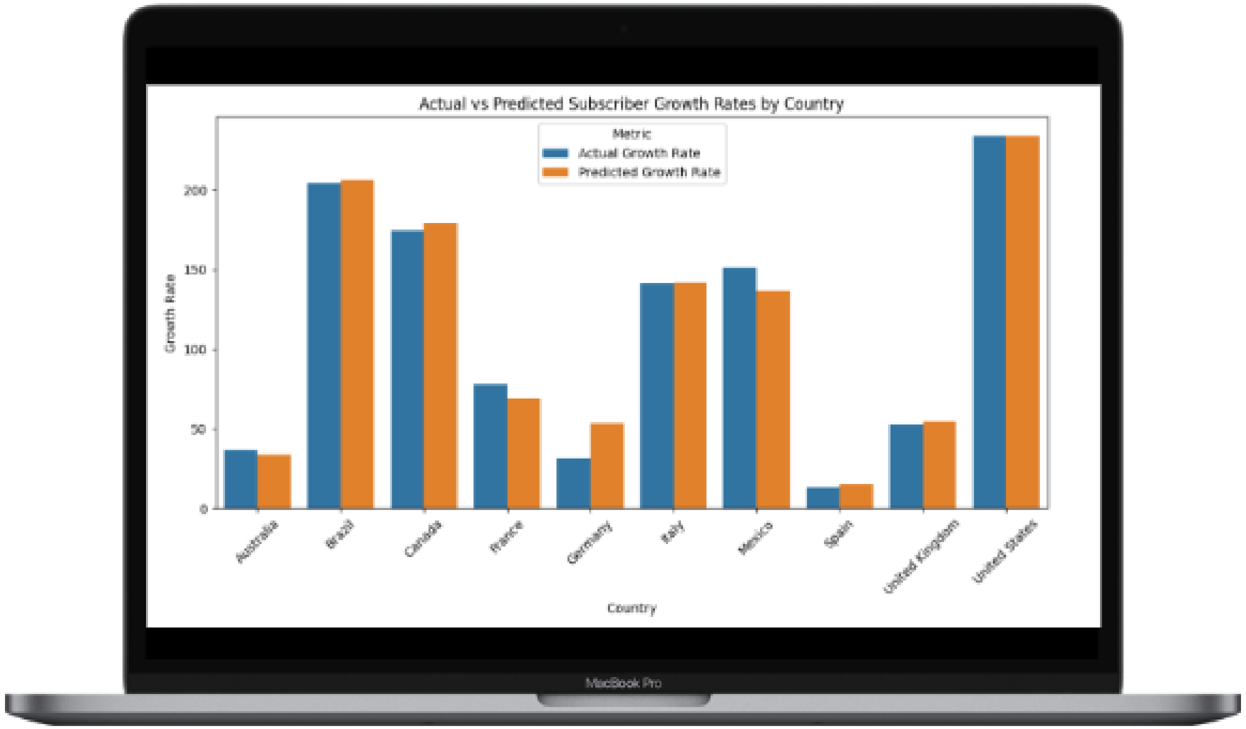
Less accurate than LSTM

Typically used for image processing

MODEL PERFORMANCE COMPARISON

Evaluation and Results: Subscriber Growth Prediction

Comparative Analysis of Random Forest Model Metrics



Random Forest Regressor

01

Subscriber growth prediction accuracy.

Important factors: Region, subscription type, revenue.

Random Forest Model Performance

02

R-squared (R^2): 0.9861 indicates excellent predictive accuracy.

RMSE: 43.73, showing low error in predictions.

Highly effective for forecasting subscriber growth.

Key Insights

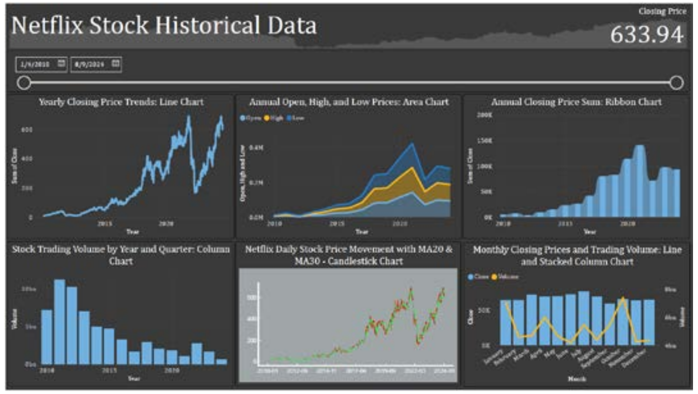
03

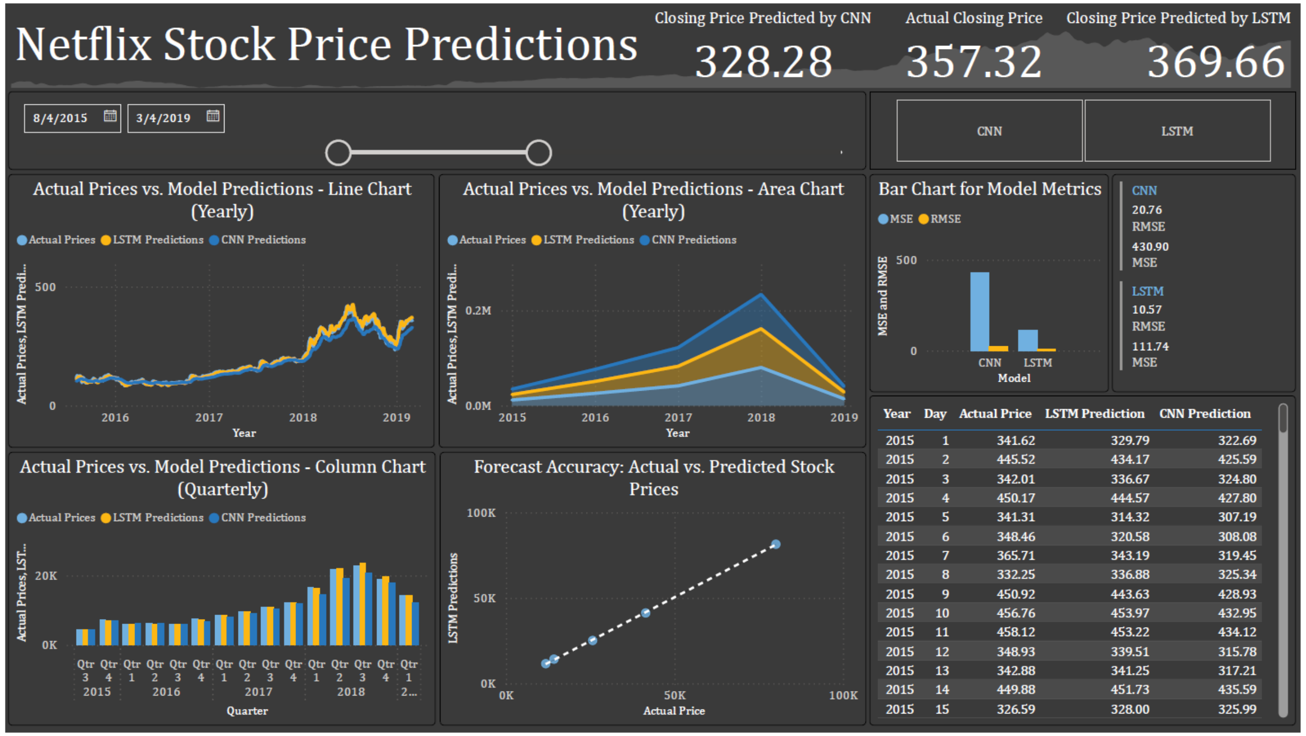
High accuracy in predicting regional subscriber growth.

STOCK PRICE PREDICTIONS

Power BI Integration: Stock Price Predictions

Comparative Analysis of Actual Prices vs. Model Predictions

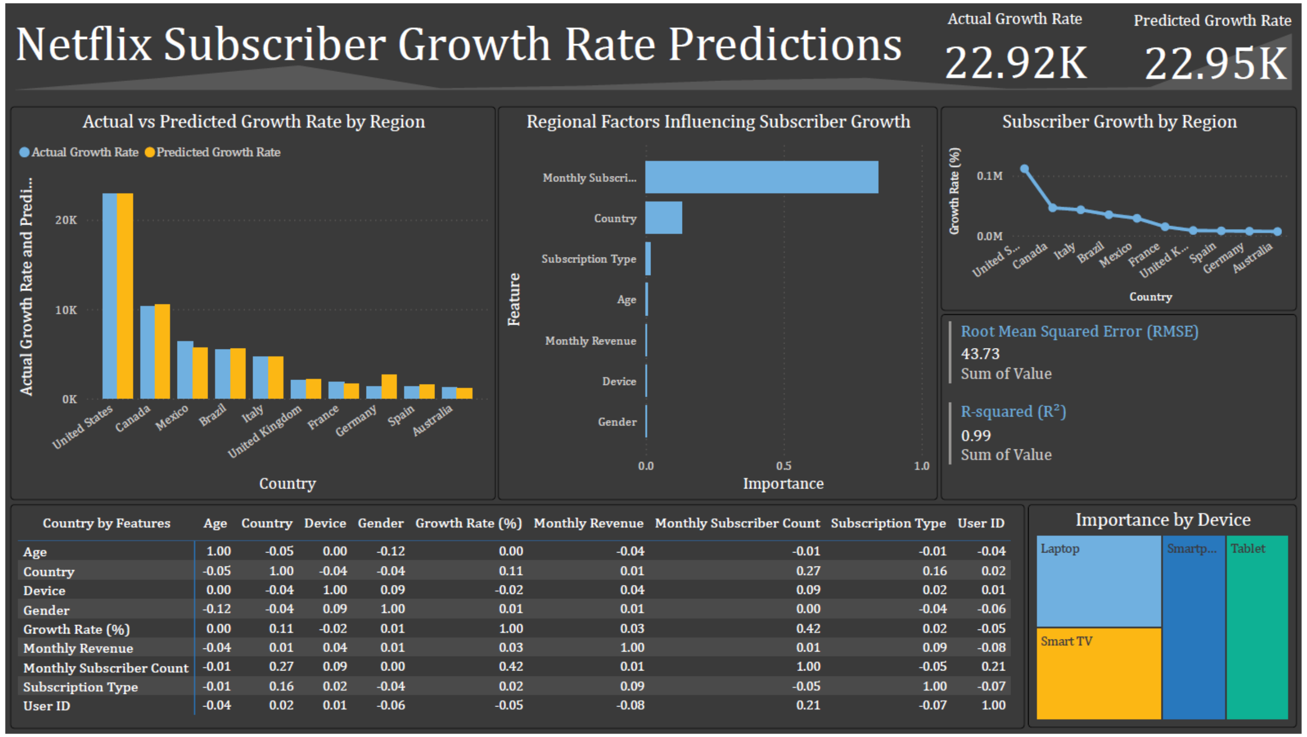




SUBSCRIBER GROWTH ANALYSIS

Power BI Integration: Subscriber Growth Predictions

Analyzing Actual vs. Predicted Subscriber Growth Across Regions



Country by Features	Age	Country	Device	Gender	Growth Rate (%)	Monthly Revenue	Monthly Subscriber Count	Subscription Type	User ID
Age	1.00	-0.05	0.00	-0.12	0.00	-0.04	-0.01	-0.01	-0.04
Country	-0.05	1.00	-0.04	-0.04	0.11	0.01	0.27	0.16	0.02
Device	0.00	-0.04	1.00	0.09	-0.02	0.04	0.09	0.02	0.01
Gender	-0.12	-0.04	0.09	1.00	0.01	0.01	0.00	-0.04	-0.06
Growth Rate (%)	0.00	0.11	-0.02	0.01	1.00	0.03	0.42	0.02	-0.05
Monthly Revenue	-0.04	0.01	0.04	0.01	0.03	1.00	0.01	0.09	-0.08
Monthly Subscriber Count	-0.01	0.27	0.09	0.00	0.42	0.01	1.00	-0.05	0.21
Subscription Type	-0.01	0.16	0.02	-0.04	0.02	0.09	-0.05	1.00	-0.07
User ID	-0.04	0.02	0.01	-0.06	-0.05	-0.08	0.21	-0.07	1.00

Conclusion and Recommendations

01

Summary

- LSTM outperformed CNN in stock price predictions, demonstrating strong temporal forecasting.
- Random Forest Regressor provided accurate subscriber growth predictions across regions.
- Power BI enhanced data interpretation through interactive visualizations.

02

Recommendations

- Extend models to other streaming platforms.
- Explore additional AI techniques (e.g., transformers, hybrid models).
- Enhance data integration for more robust real-time forecasts.