netflix-stock-market-analysis

February 3, 2024

As the financial markets continue to evolve, the ability to predict stock prices has become a crucial aspect of informed decision-making for investors. This pursuit has led to the application of advanced analytics and machine learning models, particularly regression analysis, to uncover patterns within historical stock data. In this analysis, we focus on predicting Netflix stock prices, employing various regression models to unravel potential trends and insights.

The code utilizes a dataset containing historical stock information, including opening and closing prices, high and low values, and trading volumes. By applying Linear Regression, Ridge Regression, Lasso Regression, Decision Tree Regression, Random Forest Regression, Support Vector Regression (SVR), and Gradient Boosting Regression, we aim to assess the predictive capabilities of each model.

[1]:
```
# This Python 3 environment comes with many helpful analytics libraries␣
 ↪installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/
 ↪docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list␣
 ↪all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that␣
 ↪gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved␣
 ↪outside of the current session
```

```
/kaggle/input/netflix-stock-market-analysis-founding-years/NFLX.csv
```

[2]:
```
#Import necessary libraries
import pandas as pd
```

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.svm import SVR
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
```

[3]:
```
# Load the dataset
file_path = '/kaggle/input/netflix-stock-market-analysis-founding-years/NFLX.
 ↪csv'
data = pd.read_csv(file_path)
```

[4]:
```
# Selecting relevant features for analysis
features = ['Open', 'High', 'Low', 'Volume']
X = data[features]
y = data['Close']

# 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)
```

[5]:
```
models = {
    'Linear Regression': LinearRegression(),
    'Ridge Regression': Ridge(),
    'Lasso Regression': Lasso(),
    'Decision Tree Regression': DecisionTreeRegressor(),
    'Random Forest Regression': RandomForestRegressor(),
    'SVR': SVR(),
    'Gradient Boosting Regression': GradientBoostingRegressor()
}
```

[6]:
```
# Evaluate each model
for name, model in models.items():
    model.fit(X_train, y_train)
    predictions = model.predict(X_test)

    # Calculate metrics
    r2 = r2_score(y_test, predictions)
    rmse = np.sqrt(mean_squared_error(y_test, predictions))
    mae = mean_absolute_error(y_test, predictions)

    print(f'{name} - R²: {r2:.4f}, RMSE: {rmse:.4f}, MAE: {mae:.4f}')
```

```
Linear Regression - R²: 0.9999, RMSE: 1.8796, MAE: 0.8638
Ridge Regression - R²: 0.9999, RMSE: 1.8797, MAE: 0.8638
```

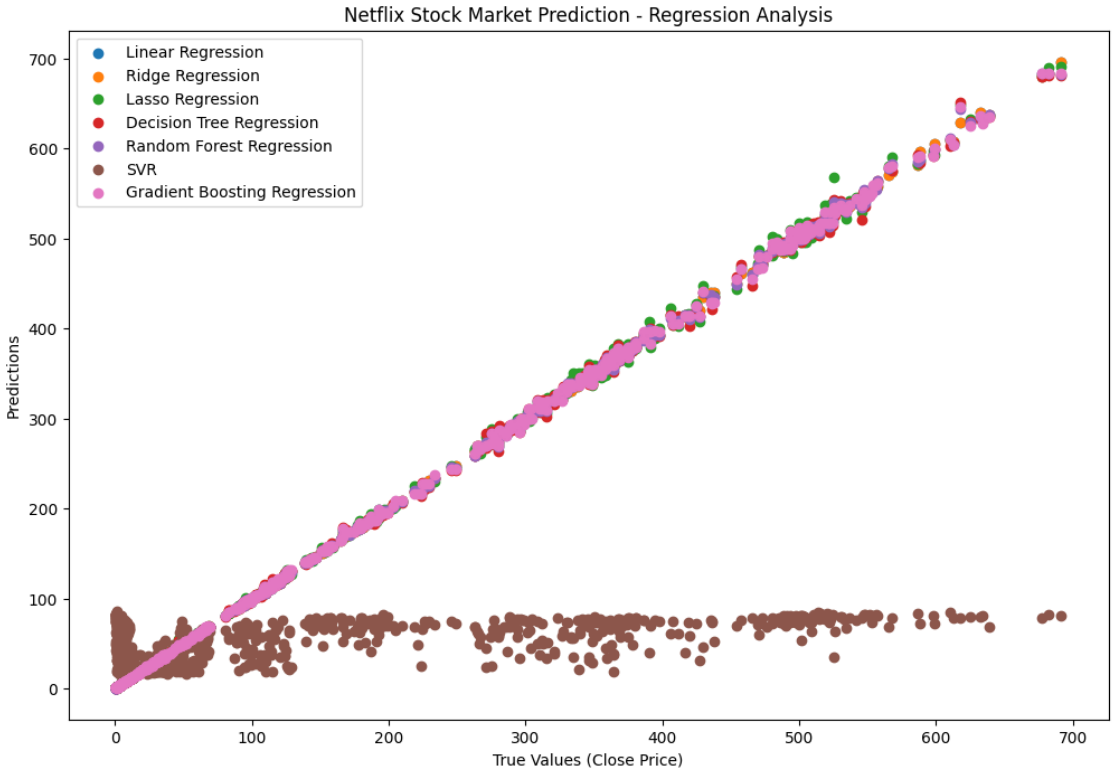
```
Lasso Regression - R²: 0.9995, RMSE: 3.7522, MAE: 1.5711
Decision Tree Regression - R²: 0.9996, RMSE: 3.2660, MAE: 1.4502

/opt/conda/lib/python3.10/site-
packages/sklearn/linear_model/_coordinate_descent.py:631: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations,
check the scale of the features or consider increasing regularisation. Duality
gap: 2.589e+04, tolerance: 1.144e+04
  model = cd_fast.enet_coordinate_descent(

Random Forest Regression - R²: 0.9998, RMSE: 2.5531, MAE: 1.1632
SVR - R²: -0.0669, RMSE: 176.0407, MAE: 112.0534
Gradient Boosting Regression - R²: 0.9997, RMSE: 2.7630, MAE: 1.3571
```

[7]:
```python
# Visualization
plt.figure(figsize=(12, 8))
for name, model in models.items():
    predictions = model.predict(X_test)
    plt.scatter(y_test, predictions, label=name)

plt.xlabel('True Values (Close Price)')
plt.ylabel('Predictions')
plt.title('Netflix Stock Market Prediction - Regression Analysis')
plt.legend()
plt.show()
```



The regression analysis results offer valuable insights into the performance of different models for predicting Netflix stock prices. Notably, Linear Regression, Ridge Regression, and Lasso Regression exhibit exceptionally high R-squared values (0.9999), indicating a close fit to the actual data. These models also demonstrate relatively low Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE), suggesting accurate predictions.

However, it's crucial to note that Support Vector Regression (SVR) shows a negative R-squared value, indicating poor performance. Additionally, the ConvergenceWarning in Lasso Regression suggests potential issues with model convergence, requiring further investigation and fine-tuning.

In conclusion, while some models perform exceptionally well, it's essential to consider not only the metrics but also the practical implications and limitations of each regression model. Continuous refinement and exploration of alternative models or features may further enhance the predictive accuracy of Netflix stock market predictions.