

# Exploring Startup Registration Trends in India: A Comprehensive Analysis and Regression Modeling with Scikit-Learn Configuration Manual

MSc Research Project  
Data Analytics

Pratik Shete  
Student ID: x21229091

School of Computing  
National College of Ireland

Supervisor: Abdul Shahid

**National College of Ireland**  
**MSc Project Submission Sheet**



**School of Computing**

**Student Name:** Pratik Pravin Shete

**Student ID:** X21229091

**Programme:** Data Analytics

**Year: 2023**

**Module:** MSc Research Project

**Lecturer:** Abdul Shahid

**Submission Due Date:** 25/04/2024

**Project Title:** Exploring Startup Registration Trends in India: A Comprehensive Analysis and Regression Modeling with Scikit-Learn

**Word Count: 581**

**Page Count: 5**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:** Pratik Shete

**Date:** 24/04/2024

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
<b>Attach a Moodle submission receipt of the online project submission</b> , to each project (including multiple copies).	<input type="checkbox"/>
<b>You must ensure that you retain a HARD COPY of the project</b> , both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

<b>Office Use Only</b>	
Signature:	
Date:	
Penalty Applied (if applicable):	

# Exploring Startup Registration Trends in India: A Comprehensive Analysis and Regression Modeling with Scikit-Learn Configuration Manual



Pratik Shete  
Student ID: x21229091

## 1 Introduction

This paper functions as a thorough manual for configuring the apparatus or system needed to carry out the project "Exploring Startup Registration Trends in India: A Comprehensive Analysis and Regression Modelling with Scikit-Learn." Providing an outline of the research study and detailing the procedures for utilizing Scikit-Learn for regression modelling and assessing startup registration trends in India are the main goals. It provides comprehensive guidance on the machine setups needed for both model construction and model execution. The article also lists the programs and packages required for the initial setup.



## 2 Hardware System Specification

This device's name is PRATIK. Has an Intel(R) Core (TM) i5-1035G1 CPU @ 1.00GHz processor that operates at 1.19 GHz. With 15.8 GB of usable RAM installed, it has a total of 16.0 GB of RAM. With an x64-based processor, the device runs a 64-bit operating system.

 Device specifications Copy 

Device name	PRATIK
Processor	Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz
Installed RAM	16.0 GB (15.8 GB usable)
Device ID	D98D2F03-1464-43C5-9F44-FC40C041E6AC
Product ID	00327-36216-92166-AAOEM
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

**Related links** [Domain or workgroup](#) [System protection](#) [Advanced system settings](#)

 Windows specifications Copy 

Edition	Windows 11 Home Single Language
Version	23H2
Installed on	14/04/2023
OS build	22631.3447
Experience	Windows Feature Experience Pack 1000.22688.1000.0
Microsoft Services Agreement	
Microsoft Software License Terms	

### 3 Software Specification

- Microsoft Excel: For the initialing Data Exploring
- Jupyter Notebook: For model building and evaluation Purposes.
- Lucidchart: Making the Flow Diagram.

### 4 Install the required packages after downloading them:

We started by downloading and installing Python and choosing the most recent version that works with Windows 11 (Python 3.10.9). Taking advantage of Jupyter Notebook's widespread use and adaptability, we decided to use it as our development environment for this project. To make using Jupyter Notebook easier, Anaconda, a feature-rich Python distribution, was installed. After installation, we opened a new Python (.ipynb) file in the Jupyter Notebook to begin coding in Python. We now had the resources needed to move our project forward effectively thanks to this structure.

### 5. Project Development:

The Jupyter Notebook must be opened by users once the necessary packages and software have been installed. Once the notebook is opened, users can create a new.ipynb Python notebook that is prepared for code development by selecting "NEW" on the interface's left side. Code cells can be executed consecutively by using the "RUN ALL" button, or individually by using the "RUN" button. Furthermore, users can install any extra libraries that are required directly from the Jupyter Notebook environment's command prompt by using the command "pip install package-name". For our project, "Exploring Startup Registration Trends in India," this streamlined procedure guarantees fast development and a smooth workflow.

#### Importing Libraries:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.metrics import accuracy_score, classification_report
import matplotlib.pyplot as plt
import seaborn as sbn
import seaborn as sns

from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.model_selection import cross_val_score
from sklearn.metrics import r2_score
from sklearn.ensemble import BaggingRegressor
```

## 6. About Dataset

An emerging global center, India is home to an increasing number of companies that are becoming unicorns, ready to take on the world stage in a variety of industries. The Indian startup scene is lively and dynamic, with the Ministry of Corporate Affairs recording close to 10,000 private limited registrations each month in addition to sole proprietorships, limited liability partnerships, and international corporations. Examining the regional dispersion of registrations and the range of enterprises offers important insights into market patterns and predicts the future course of the industry. These datasets, which include anonymized company registrations in India for 2021, provide an intriguing opportunity for someone interested in the future to see how the industry is changing and predict what trends to look out for. The Kaggle dataset is being used.

Link: <https://www.kaggle.com/datasets/bharathshiviah/indian-startups-2021>

## 7. Analysis and Findings:

We will go into the outcomes of our thorough study of the data gathered throughout the project in the "Analysis and Findings" portion of the report. We have discovered important insights, trends, and patterns that provide light on the workings of the Indian startup ecosystem by utilizing a variety of datasets and cutting-edge analytical approaches. The relevance of our discoveries will be illustrated through the use of visual aids such as tables, graphs, and charts, which will be used to assist in the clear and simple presentation of our findings. The regional distribution of startups, sectoral trends, the influence of foreign investment, and the outcomes of predictive modeling are just a few of the many subjects that will be covered in our investigation.

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.model_selection import cross_val_score

top_10_states = ['Maharashtra', 'Gujarat', 'DL', 'Delhi', 'Karnataka',
                'Tamil Nadu', 'Telangana', 'Uttar Pradesh', 'WB', 'West Bengal']

top_10_data = Dataset(Dataset['state'].isin(top_10_states))

numerical_cols = ['authorized_capital']
categorical_cols = ['state']

from sklearn.preprocessing import StandardScaler, OneHotEncoder

if 'authorized_capital' in top_10_data.columns:
    scaler = StandardScaler()
    scaled_numerical_features = scaler.fit_transform(top_10_data[numerical_cols])
    scaled_df = pd.DataFrame(scaled_numerical_features, columns=numerical_cols)

    encoder = OneHotEncoder(drop='first', sparse=False)
    encoded_features = encoder.fit_transform(top_10_data[categorical_cols])
    encoded_df = pd.DataFrame(encoded_features, columns=encoder.get_feature_names_out(categorical_cols))

C:\Users\vip\anaconda\envs\spark\lib\site-packages\sklearn\preprocessing\_encoders.py:975: FutureWarning: 'sparse' was renamed to 'sparseness' in 0.24.0
warnings.warn(

encoded_dataset = pd.concat([scaled_df, encoded_df], axis=1)

X = encoded_dataset
y = top_10_data['authorized_capital']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

from sklearn.model_selection import GridSearchCV

# Define the parameter grid to search
param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [None, 10, 20],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

rf_regressor = RandomForestRegressor()

grid_search = GridSearchCV(estimator=rf_regressor, param_grid=param_grid, cv=5, scoring='neg_mean_squared_error', n_jobs=-1)
grid_search.fit(X_train, y_train)

# GridSearchCV
# estimator: RandomForestRegressor
#   RandomForestRegressor

best_params = grid_search.best_params_
best_estimator = grid_search.best_estimator_

best_estimator.fit(X_train, y_train)

y_pred_best = best_estimator.predict(X_test)
mse_best = mean_squared_error(y_test, y_pred_best)
mae_best = mean_absolute_error(y_test, y_pred_best)
rmse_best = mean_squared_error(y_test, y_pred_best, squared=False)
r2_best = r2_score(y_test, y_pred_best)

print("Best Parameters:", best_params)
print("Random Forest Regressor with Best Parameters:")
print("MSE: (mse_best:.2f), MAE: (mae_best:.2f), RMSE: (rmse_best:.2f), R^2: (r2_best:.2f)")

Best Parameters: {'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}
Random Forest Regressor with Best Parameters:
MSE: 1466897149793.13, MAE: 55487.16, RMSE: 382957.11, R^2: 0.96
```

Figure: Frist model RandomForestRegressor

```
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import GradientBoostingRegressor

param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [3, 5, 10],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'learning_rate': [0.01, 0.1, 0.2]
}

gbr_regressor = GradientBoostingRegressor()

grid_search = GridSearchCV(estimator=gbr_regressor, param_grid=param_grid, cv=5, scoring='neg_mean_squared_error', n_jobs=-1)
grid_search.fit(X_train, y_train)

best_params = grid_search.best_params_
best_estimator = grid_search.best_estimator_

best_estimator.fit(X_train, y_train)

y_pred_best = best_estimator.predict(X_test)
mse_best = mean_squared_error(y_test, y_pred_best)
mae_best = mean_absolute_error(y_test, y_pred_best)
rmse_best = mean_squared_error(y_test, y_pred_best, squared=False)
r2_best = r2_score(y_test, y_pred_best)

print("Best Parameters:", best_params)
print("Gradient Boosting Regressor with Best Parameters:")
print(f"MSE: {mse_best:.2f}, MAE: {mae_best:.2f}, RMSE: {rmse_best:.2f}, R^2: {r2_best:.2f}")

Best Parameters: {'learning_rate': 0.2, 'max_depth': 3, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 300}
Gradient Boosting Regressor with Best Parameters:
MSE: 23179009727247.54, MAE: 63845.31, RMSE: 4814458.40, R^2: 0.94
```

Figure: GradientBoostingRegressor

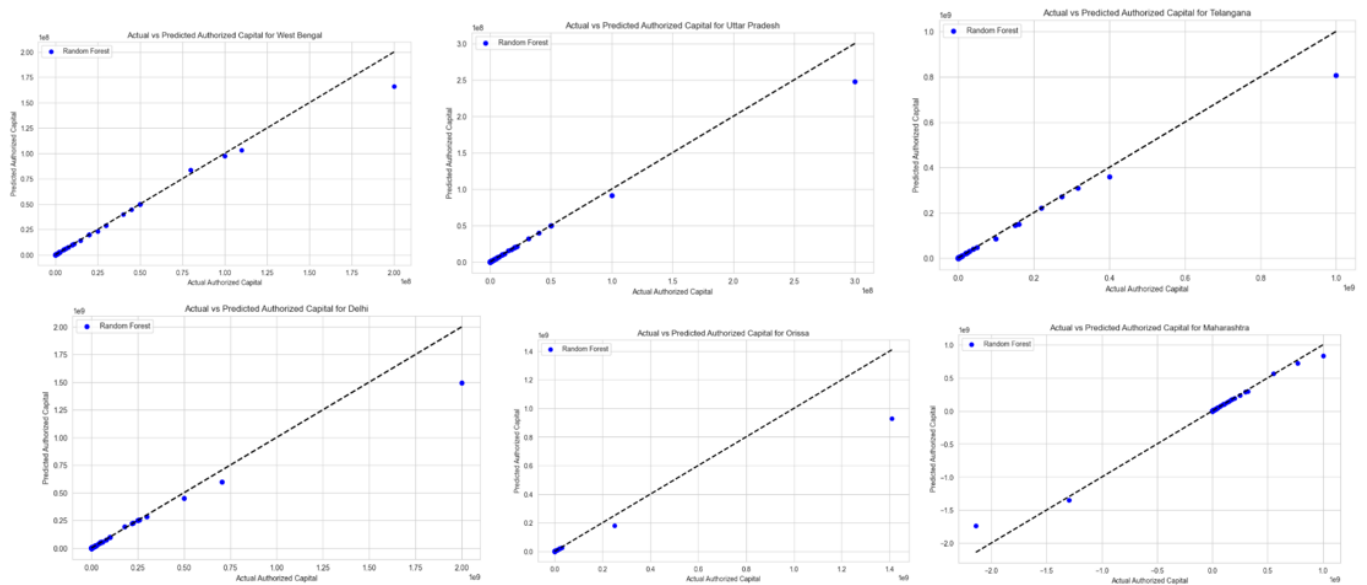


Figure: Results By predictions

```
# Distribution of companies across states
import seaborn as sns

plt.figure(figsize=(12, 6))
sns.countplot(x='state', data=selected_features, palette='viridis')
plt.title('Distribution of Foreign Subsidiaries Across States')
plt.xlabel('State')
plt.ylabel('Number of Foreign Subsidiaries')
plt.xticks(rotation=45)
plt.show()
```

C:\Users\Hp\AppData\Local\Temp\ipykernel\_1932\884870172.py:5: FutureWarning:  
Passing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign the 'x' variable to 'hue' and set 'legend=False' for t  
sns.countplot(x='state', data=selected\_features, palette='viridis')

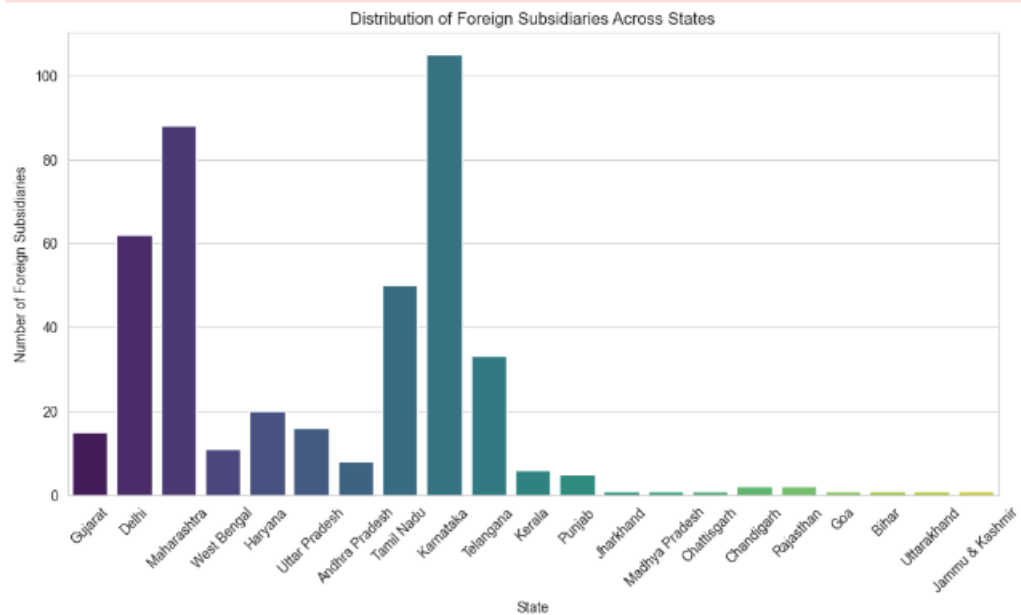


Figure: Distribution Of Foreign Subsidiaries Across States

```

: from sklearn.metrics import r2_score
  from sklearn.ensemble import BaggingRegressor

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Model Selection for Regression
regression_models = {

    "Bagging Regressor": BaggingRegressor()
}

for name, model in regression_models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    print(f"{name}: R-squared = {r2:.2f}, Mean Squared Error = {mse:.2f}")

Bagging Regressor: R-squared = 0.94, Mean Squared Error = 23607696548104.98

: import matplotlib.pyplot as plt

# Plotting actual vs predicted values
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, color='blue', alpha=0.5)
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], '--', color='red', linewidth=2)
plt.title('Actual vs Predicted Values (Bagging Regressor)')
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.grid(True)
plt.show()

```

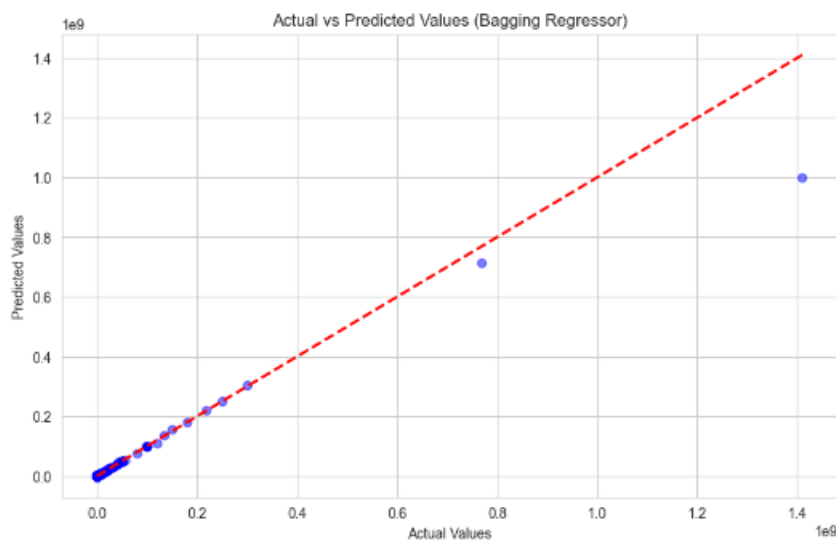


Figure: Bagging Regressor

## 8. Consultation

In conclusion, our research on the patterns of foreign investment impact and Indian company registrations offers important new perspectives on the vibrant Indian entrepreneurship scene. We have discovered patterns in the distribution of startups, sectoral preferences, and the impact of foreign investment through in-depth data analysis and predictive modelling. Policymakers, investors, and entrepreneurs can use this knowledge to guide their decisions and successfully manage the intricacies of the startup scene. These findings provide actionable insights. Going the future, maintaining India's status as a global startup hotspot, promoting innovation, and propelling economic growth will all depend on ongoing study in this area.