STARTUP DATA ANALYSIS USING MACHINE LEARNING

Under The Guidance Of:
Sk. Riyaz Bagban

Designed By:
Sk. Aiman Sabaha
N. Lakshmi Sudheshna

Startup Data Analysis using Machine Learning.

1. Project Overview

- **Objective**: The primary objective of this project is to analyze key factors that influence startup success and develop a predictive model to determine whether a startup will rank among the top 500. This involves exploring datasets, identifying trends, and utilizing machine learning models for predictions.

- **Scope**: The project covers data cleaning, visualization, modelling, and interface development using various ML algorithms and python methods.

- **Dataset Description**:
    - Dataset is taken from Kaggle.
    - Key attributes and their descriptions.

      state_code: The state code where the startup is located.
      latitude: The latitude of the startup's geographical location.
      longitude: The longitude of the startup's geographical location.
      zip_code: The zip code of the startup's location.
      id: A unique identifier for the startup.
      city: The city where the startup is located.
      name: The name of the startup.
      labels: Additional tags or labels associated with the startup.
      founded_at: The date when the startup was founded.
      closed_at: The date when the startup was closed (if applicable).
      first_funding_at: The date when the startup received its first funding.
      last_funding_at: The date when the startup received its most recent funding.
      age_first_funding_year: The age of the startup (in years) at the time of its first funding.
      age_last_funding_year: The age of the startup (in years) at the time of its last funding.
      age_first_milestone_year: The age of the startup (in years)

when it achieved its first milestone.
age_last_milestone_year: The age of the startup (in years) when it achieved its most recent milestone.
relationships: The number of relationships the startup has (e.g., partnerships or collaborations).
funding_rounds: The total number of funding rounds the startup has participated in.
funding_total_usd: The total funding received by the startup, in USD.
milestones: The total number of milestones achieved by the startup.
is_CA: Indicates whether the startup is based in California (1 for yes, 0 for no).
is_NY: Indicates whether the startup is based in New York (1 for yes, 0 for no).
is_MA: Indicates whether the startup is based in Massachusetts (1 for yes, 0 for no).
is_TX: Indicates whether the startup is based in Texas (1 for yes, 0 for no).
is_otherstate: Indicates whether the startup is located in states other than CA, NY, MA, or TX (1 for yes, 0 for no).
category_code: The category or industry of the startup (e.g., software, biotech, etc.).
is_software: Indicates whether the startup belongs to the software category (1 for yes, 0 for no).
is_web: Indicates whether the startup belongs to the web category (1 for yes, 0 for no).
is_mobile: Indicates whether the startup belongs to the mobile category (1 for yes, 0 for no).
is_enterprise: Indicates whether the startup belongs to the enterprise category (1 for yes, 0 for no).
is_advertising: Indicates whether the startup belongs to the advertising category (1 for yes, 0 for no).
is_gamesvideo: Indicates whether the startup belongs to the games or video category (1 for yes, 0 for no).
is_ecommerce: Indicates whether the startup belongs to the e-commerce category (1 for yes, 0 for no).
is_biotech: Indicates whether the startup belongs to the biotech category (1 for yes, 0 for no).
is_consulting: Indicates whether the startup belongs to the

consulting category (1 for yes, 0 for no).
is_othercategory: Indicates whether the startup belongs to categories other than the ones listed above (1 for yes, 0 for no).
object_id: Another unique identifier for the startup.
has_VC: Indicates whether the startup has venture capital funding (1 for yes, 0 for no).
has_angel: Indicates whether the startup has angel investment (1 for yes, 0 for no).
has_roundA: Indicates whether the startup has raised Series A funding (1 for yes, 0 for no).
has_roundB: Indicates whether the startup has raised Series B funding (1 for yes, 0 for no).
has_roundC: Indicates whether the startup has raised Series C funding (1 for yes, 0 for no).
has_roundD: Indicates whether the startup has raised Series D funding (1 for yes, 0 for no).
avg_participants: The average number of participants in the startup's funding rounds.
is_top500: Indicates whether the startup is ranked among the top 500 startups (1 for yes, 0 for no).
status: The status of the startup (0 for acquired, 1 for closed).

  - Tabular dataset with 923x32 size.

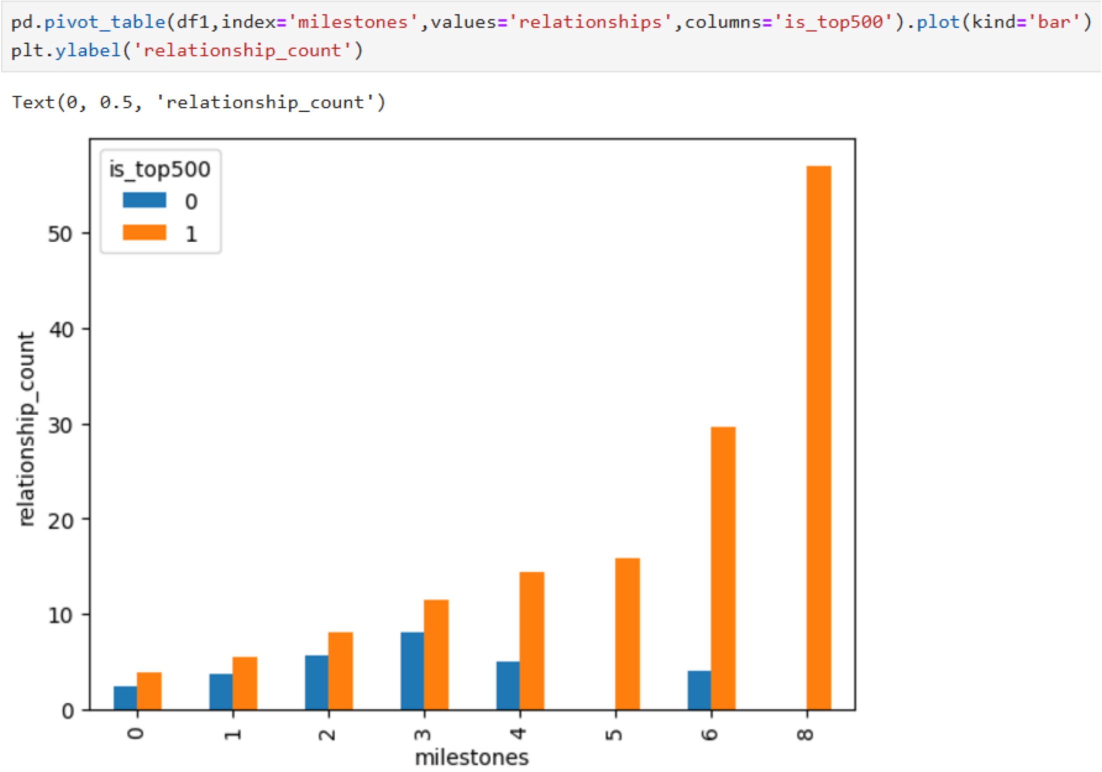
2. Data Cleaning and Preprocessing

- **Initial Analysis**:
  - Handling missing values.
  - Identifying and managing outliers.
  - Addressing duplicates.
- **Transformations**:
  - Deriving new column from existing columns
    - startup_age_in_years = closed_at – founded_at

      - funding_age_years = age_last_funding_year-age_first_funding_year
      - years_since_last_funding = startup_age_in_years-age_last_funding_year
   - Data normalization or standardization.
   - Encoding categorical variables.
   - Apply SMOTE for treating imbalanced data.
   - Feature selection and engineering using.

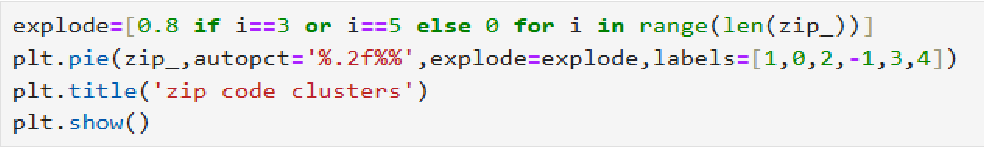
3. Exploratory Data Analysis (EDA)

- **Data Visualization**:
   - *Distribution plots for features.*

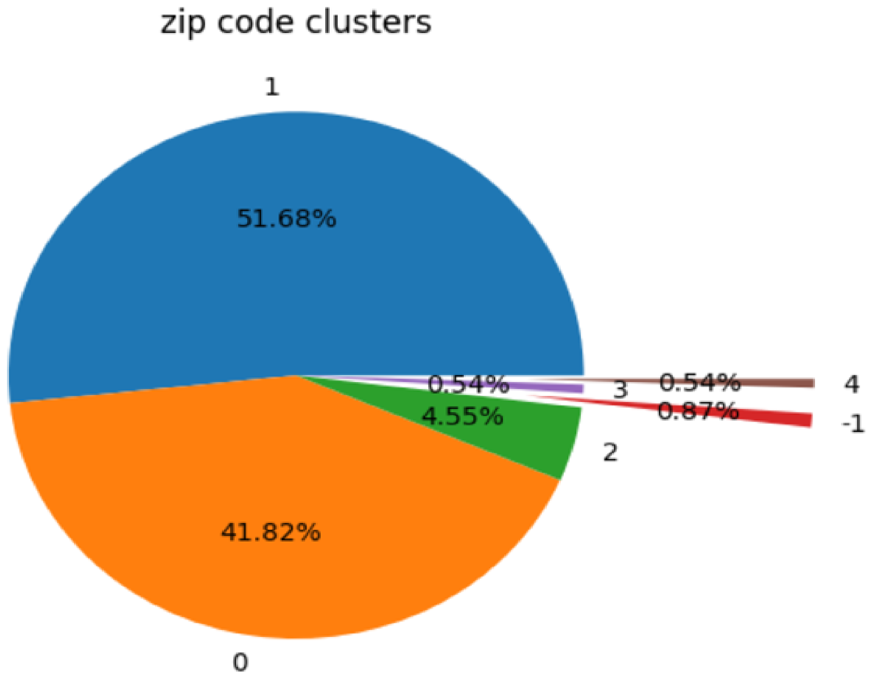


```
pd.pivot_table(df1,index='milestones',values='relationships',columns='is_top500').plot(kind='bar')
plt.ylabel('relationship_count')
```

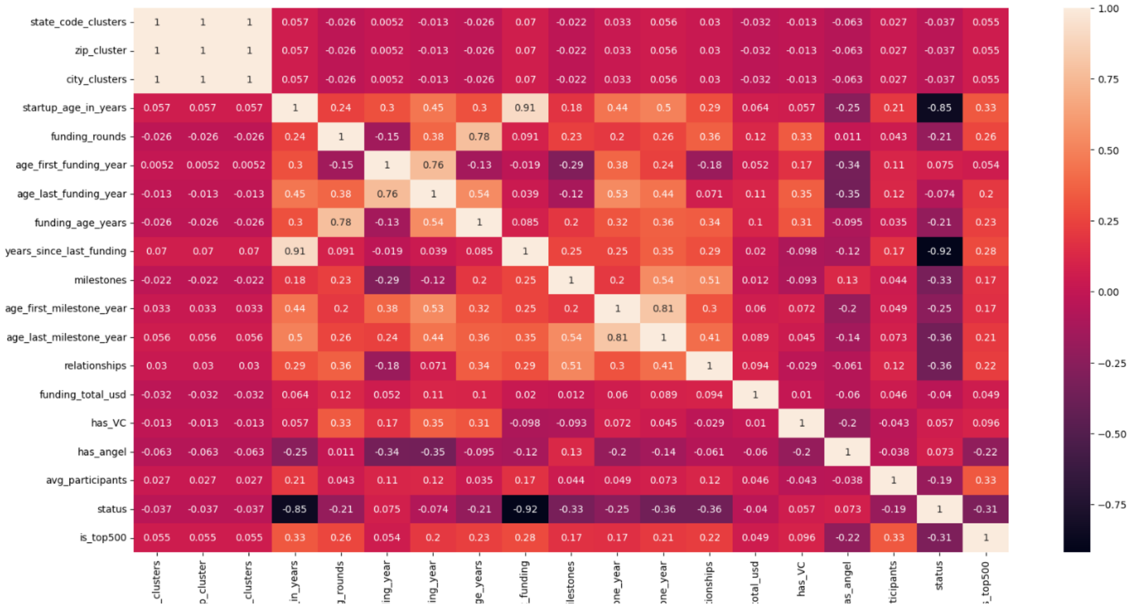
```
Text(0, 0.5, 'relationship_count')
```



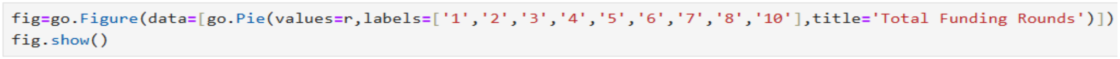
```
explode=[0.8 if i==3 or i==5 else 0 for i in range(len(zip_))]
plt.pie(zip_,autopct='%.2f%%',explode=explode,labels=[1,0,2,-1,3,4])
plt.title('zip code clusters')
plt.show()
```



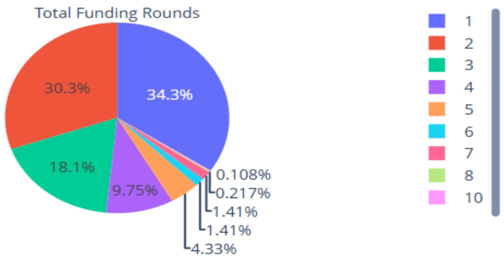
- *Correlation heatmaps.*

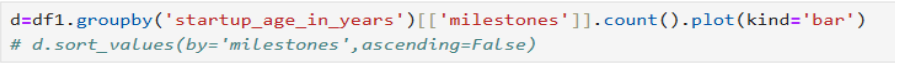


- *Relationship analysis using pie charts and bar plots.*

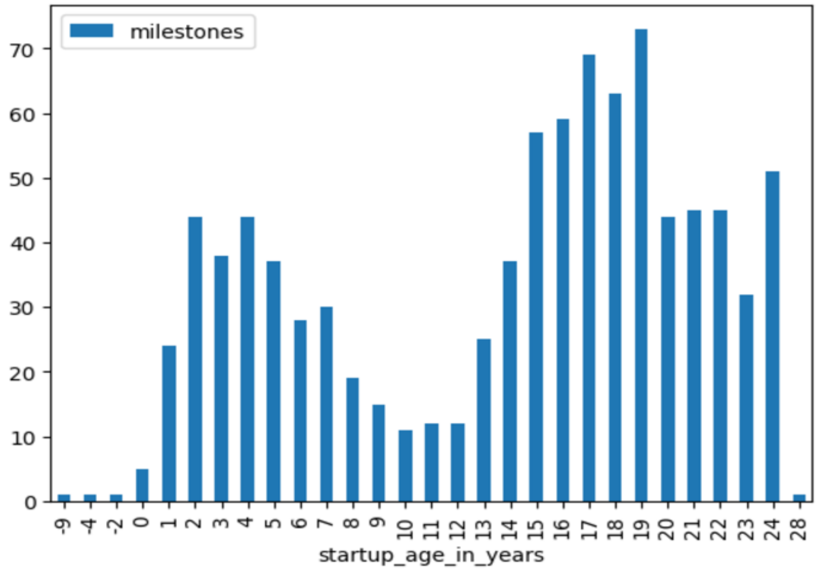


```
fig=go.Figure(data=[go.Pie(values=r,labels=['1','2','3','4','5','6','7','8','10'],title='Total Funding Rounds')])
fig.show()
```





```
d=df1.groupby('startup_age_in_years')[['milestones']].count().plot(kind='bar')
# d.sort_values(by='milestones',ascending=False)
```



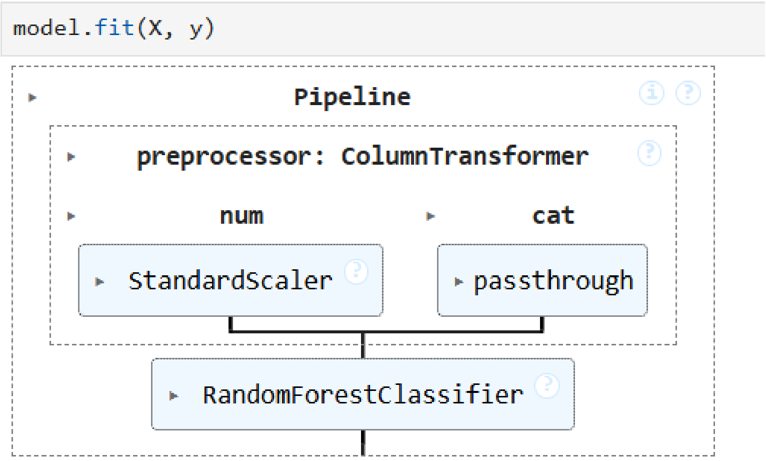
4. Algorithm Implementation

- **Algorithms Applied**:
  - Classification Algorithms
    - Logistic Regression.
    - Decision Tree Classifier.
    - Random Forest Classifier.
    - K Nearest Neighbors (KNN).

        - Gaussian NB.
        - XGBoost.
        - Bagging Classifier.
        - Support Vector Machine
        - AdaBoost Classifier.
    - These algorithms are selected to check and find the algorithm that provides more accuracy with the given values.
- **Model Training and Testing**:
    - Train-test split and cross-validation methods.
    - Accuracy and Prediction, Standard Deviation.
- **Accuracy Prediction**:
    - Results from various models are collected and compared so that the best model is selected for further process.
    - Comparative analysis of model performance.

5. Pipeline Development

- **Pipeline Design**: Outline the sequence of processes, including data preprocessing, model training, and prediction.



```
model.fit(X, y)
```

- Automated the workflow, from preprocessing to prediction ensuring reproducibility and efficiency.

6. User Interface (UI)

- **UI Design**:
  - The purpose of the interface is inputting new data and displaying predictions.
  - Spyder with Streamlit Tool used in this Interface.
- **Functionality**:
  - The user opens the webpage and enter the startup details like founding date, funding duration, period for milestones, relationship with others and the current status of the startup, etc. in the section.
  - After entering the input, the user hits the Predict button just below the input section. This results in the output that displays the startup is on top 500 or not based on the algorithm selected.
  - If the predicted output is on top 500 then the prediction results will display along with balloons and quotes to increase user's joy otherwise only result will be displayed with motivational quotes.
  - After displaying the prediction results, the user data will get stored in a CSV file for developer for future purpose.

7. Results and Evaluation

- **Final Model Performance**: Random Forest algorithm is giving high accuracy of around 91% with less standard deviation compared to other algorithms.
- **Visualization of Results**: Graphical representations for clarity.
- **Discussion**: Analysis of results, including limitations and potential improvements.

8. Conclusion and Future Work

The project successfully showcased the complete data analysis workflow, including cleaning, visualization and prediction using machine learning algorithms. It achieved accurate results by selecting the best model, automating the processes with pipelines and creating a user-friendly interface seamless input and prediction.

Challenges:

Missing Values and Inconsistent Format

- The missing values were handled using imputation techniques like predictive methods and null values are replaced with 0 or current dates for date columns.
- Data normalization and standardization ensures the consistency across features.

Algorithm Prediction

- Multiple algorithms were evaluated to find the best-performing one based on the accuracy and efficiency.
- Balancing the model accuracy with computational efficiency was achieved by optimizing algorithms and fine-tuning hyperparameters.

Pipeline Integration

- Built pipelines using frameworks like scikit-learn to automate the end-to-end process by standard scalar and Label Encoding.
- Modularize the pipeline for scalability and easy updates.

UI Design

- Designed the UI with intuitive workflows using Streamlit. Ensured the clear and responsiveness of results.

Future Scope:

- The interface can be further enhanced with more features, such as advanced visualizations or additional prediction models.
- Incorporating real-time data updates and scaling the pipeline for larger datasets could improve the system’s utility.
- Additional datasets and more complex models could be explored to further enhance accuracy and robustness.

9. Appendices

- Code snippets.
- Detailed tables or raw results.
- References for external resources or tools.
- For dataset: https://www.kaggle.com
- For Streamlit Documentation: https://docs.streamlit.io/
- For Pipeline Development: https://scikit-learn.org/stable/modules/pipeline.html
- For Visualization: https://plotly.com/python/