NAAN MUDHALVAN DATASCIENCE FUDAMANETAL PROJECT

PROJECT TITLE: EMAIL SPAM CLASSIFICATION

PRESENTED BY
YOGESHWARAN.G
3RD YR MECH DEPT
SACS MAVMM ENGINEERING COLLEGE

PROJECT STATEMENT

The problem statement for classifying email scams involves developing a machine learning model capable of accurately identifying and categorizing different types of email scams. This includes phishing scams, malware distribution, fraudulent schemes, and other forms of malicious or deceptive emails. The goal is to create a system that can automatically detect and flag suspicious emails to protect users from falling victim to scams.

PROBLEM SOLUTION

To address the email scam classification problem, you can follow these steps:Data Collection: Gather a diverse dataset of labeled emails, including examples of different types of scams such as phishing, malware distribution, and fraudulent schemes.Preprocessing: Clean and preprocess the email text data, including tasks like removing stop words, stemming or lemmatization, and handling special characters or encoding issues.Feature Extraction: Extract relevant features from the email text data, such as bag-of-words representation, TF-IDF (Term Frequency-Inverse Document Frequency), word embeddings, or other text representation techniques.Model Selection: Choose appropriate machine learning models for classification, such as Naive Bayes, Support Vector Machines, Random Forests, or deep learning models like Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs).

SYSTEM APPROACH

- Building the proposed solution would involve a combination of data processing, pyth0n programming and visualization skills.
- **System requirements:**

1. **Hardware** :
    - A computer with sufficient processing power, preferably with multiple cores or a GPU for faster training of machine learning models.
    - Adequate RAM to handle the size of the dataset and computational requirements.
2. **Software:**
    - - An operating system compatible with the required python libraries (e.g., windows,linux,macOS).

SYSTEM APPROACH – CONT.

- Library Requirements:
- Data processing and analysis:
  - **Pandas :** For data manipulation and analysis.
  - **Numpy :** For numerical operations on data.
- Data visualization:
  - **Matplotlib and seaborn**: For creating visualizations to understand data patterns.
  - **Plotly or Bokeh**: Interactive visualization libraries for more complex visualizations.

ALGORITHM & DEPLOYMENT

Algorithm selection

Data exploration :

- Explore the movie rating structure, features and variables.
- Identify potential patterns, correlations and outliners.

Problem formulation:

- Define the problem: Predict optimal booking times, likelihood of special requests based on historical data.

Algorithm selection:

- Regression tasks(e.g., predicting daily rates)
  - Consider linear regression, decision tree, or ensemble methods
- classification tasks(e.g., predicting special requests);
  - Consider logistic regressive, decision trees or random forests.

ALGORITHM & DEPLOYMENT

Data input:

Data collection:

- Gather historical data including booking dates ,special requests,and relevant details.

Data cleaning:

- Handle missing values, outliners, and any inconsistencies in the dataset.
- Convert categorical variables into numerical representation through encoding techniques.

Feature Engineering:

- Create new features or modify existing ones based on domain knowledge.
- Extract meaningful information from date variables, such as day-of-week or month.

ALGORITHM & DEPLOYMENT

Training process:

Data splitting:

- Divide the dataset into training and testing sets to evaluate the model's performance.

Feature scaling:

- Standardize or normalize numerical features to ensure they have consistent scale.

Modeling training:

- Use the selected algorithm to train the model on the all sites scores dataset.
- Adjust hyperparameters to optimize model performance.

Model evaluation:

- Evaluate the model on the dataset using appropriate metrics(e.g., Mean Squared Error for regression,accuracy,precision, recall for classification).
- Fine-tune the model if necessary.

CONCLUSION

- It proved that and we have determined the Email spam classification was done and analyze fully by visualstudio

REFERENCES

- http://www.kaggle.com/datasets
- http://seaborn.pydata.org/
- http://matplotlib.org/stable/contents.html
- http://pandas.pydata.org/pandas-docs/stable/user guide/ index.html

Thank you