Problem Statement Document

Title: Exploration and Comparison of Regression Techniques and Models

Table of Contents 1.

TEAM MEMBERS

1. Shravani R S

2. Himaja S

3. Karanbir Singh

4. Tarunaa A C

5. Anushka Dutta

Date: 5th December 2024

Prepared By: Shravani R S

Problem Context

In many fields like finance, healthcare, and engineering, regression analysis plays a crucial role in understanding relationships between variables and making predictions. While linear regression is the simplest and most widely used model, alternative approaches like tree-based models, support vector machines (SVM), and K-nearest neighbors (KNN) often offer improved performance in specific scenarios. Additionally, time series prediction introduces its own challenges, requiring specialized techniques to account for sequential data dependencies.

Objective

The objective of this project is to:

1. **Explore** a given dataset to understand its structure and potential predictive capabilities.
2. Perform **feature engineering** and preprocessing to prepare the data for modeling.
3. Implement **linear regression** as a baseline model and evaluate its performance.
4. Explore **alternatives to linear regression**, including tree-based models, SVM, KNN, and others.
5. Address time-dependent patterns using **time series prediction** techniques if applicable.
6. Compare and analyze the performance of these models based on metrics such as accuracy, mean squared error (MSE), and R-squared values.

Scope

- **Dataset**: A real-world dataset (to be identified or provided) that demonstrates a regression problem (e.g., predicting house prices, stock prices, or sales revenue).
- **Techniques**: Includes data exploration, feature engineering, and model training/testing.
- **Models**: Linear regression, decision trees, random forests, SVM, KNN, and time series models.
- **Evaluation Metrics**: MSE, RMSE, R^2 score, etc., to compare model performance.

Approach

1. **Load and Explore Dataset**
    - Understand the dataset: variable types, missing values, distributions, and correlations.
    - Visualize data relationships and identify potential predictive variables.

2. **Feature Engineering**
    - Create new features (e.g., interaction terms, polynomial features).
    - Perform dimensionality reduction if necessary (e.g., PCA).
3. **Preprocessing and Data Splitting**
    - Handle missing data, outliers, and categorical variables.
    - Normalize/scale numerical data.
    - Split the dataset into training, validation, and testing sets.
4. **Implement Linear Regression**
    - Train a linear regression model as a baseline.
    - Analyze coefficients and residuals.
5. **Evaluate the Model** ○ Use appropriate metrics like MSE, RMSE, and R^2. ○ Perform cross-validation to assess generalizability.
6. **Explore Linear Regression Alternatives** ○ **Tree-Based Models**: Decision trees, random forests, gradient boosting.
    - **SVM**: Test kernel-based regression capabilities.
    - **KNN**: Assess performance with different values of k.
7. **Time Series Prediction**
    - For time-dependent data, preprocess and explore trends, seasonality, and autocorrelation.
    - Test models like ARIMA, LSTM, or Prophet for predictive performance.
8. **Compare and Report Results**
    - Summarize findings with visuals like error plots, feature importance, and model comparisons.
    - Recommend the best model(s) based on performance and interpretability.

Challenges and Risks

- Data quality issues (e.g., missing or unbalanced data).
- Overfitting in complex models.
- Difficulty interpreting black-box models like SVM or ensemble methods.
- Scalability for large datasets.

Expected Outcomes

- A comprehensive comparison of regression models on the chosen dataset.
- Identification of key factors influencing the predictions.
- Recommendations for suitable regression approaches depending on dataset characteristics and requirements.
- Insights into handling time series data for prediction tasks.

Tools and Libraries

- Python: pandas, NumPy, scikit-learn, XGBoost, TensorFlow/Keras (for LSTM).
- Visualization: Matplotlib, Seaborn, Plotly.
- Time Series: statsmodels, Prophet.

Conclusion

This project will provide a systematic understanding of regression techniques, including their strengths, limitations, and applications. The findings can guide the choice of models in future projects with similar requirements.