Comparative Analysis of Supervised Learning Algorithms on Insurance Expenses

1. Introduction

This project aims to predict insurance expenses using a regression approach on an insurance dataset. The dataset was originally provided as part of the "Mathematics for Data Science" course (Assignment 5 AITU) and contains 1,338 records with the following features:

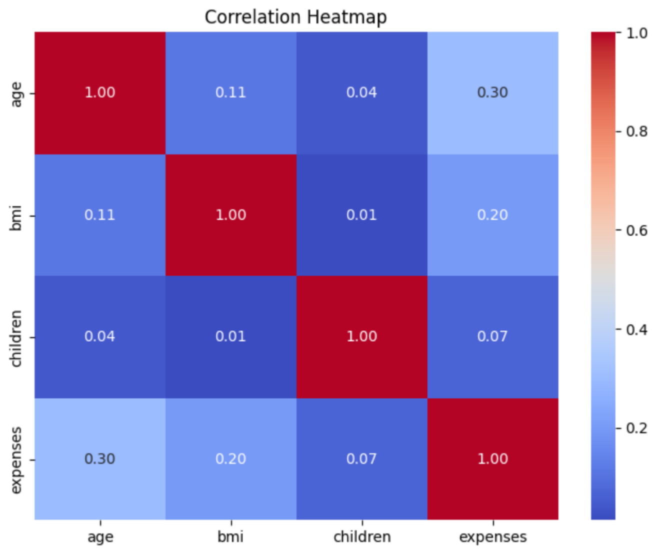
- **age**: Age of the individual.
- **sex**: Gender (represented as 0 or 1).
- **bmi**: Body Mass Index.
- **children**: Number of children.
- **smoker**: Smoking status (0 for non-smoker, 1 for smoker).
- **expenses**: Insurance expenses (target variable).

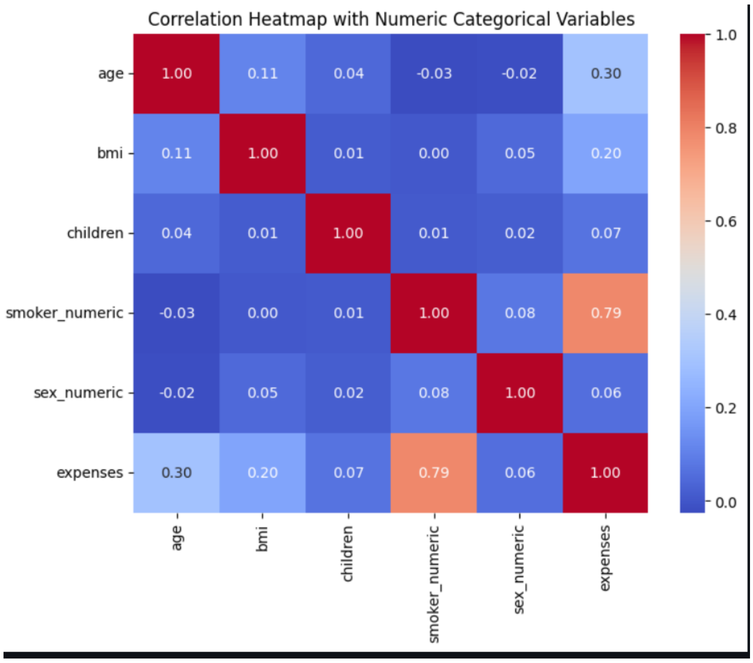
The goal is to develop several regression models to predict the insurance expenses and compare their performance using standard evaluation metrics.

2. Data Exploration and Preprocessing

Data Exploration

- **Dataset Overview**:
  The dataset contains 1,338 entries and 6 columns. An initial review using methods like .head(), .info(), and .describe() confirmed that there are no missing values in any column.
- **Descriptive Statistics**:
  The numerical features exhibit a wide range of values. For example, the age feature has a mean of 39.21 years with a standard deviation of about 14.05, while the expenses range from a minimum of around 1,725.55 to a maximum of 63,770.43.
- **Correlation Analysis**:
  A correlation heatmap was generated to visualize the relationships between features. This analysis helped in understanding the dependencies among numerical variables and informed the feature scaling decisions.





Data Preprocessing

- **Column Name Standardization**:
  All column names were converted to lowercase to ensure consistency (e.g., sex instead of Sex).

- **Data Type Conversion**:
  The categorical features (sex and smoker) were converted to the 'category' data type.

- **Scaling and Encoding**:
  - Numerical features (age, bmi, children) were standardized using StandardScaler.
  - Categorical features (sex and smoker) were transformed using one-hot encoding with drop='first' to avoid multicollinearity.

- **Train-Test Split**:
  The dataset was split into an 80% training set and a 20% test set to evaluate model performance.

3. Model Development

Five regression models were implemented using scikit-learn's Pipeline to seamlessly integrate preprocessing and model training:

1. **Linear Regression**
   A baseline linear model.

2. **Decision Tree Regressor**
   A model that recursively splits the dataset based on feature values.

3. **Random Forest Regressor**
   An ensemble method that builds multiple decision trees and averages their outputs for a robust prediction.

4. **Support Vector Regressor (SVR)**
   A kernel-based method suited for non-linear regression problems. Note that SVR was more sensitive to parameter settings in this dataset.

5. **Gradient Boosting Regressor**
   An ensemble approach that builds models sequentially to minimize prediction errors.

For the Random Forest model, hyperparameter tuning was performed using GridSearchCV with the following parameters:

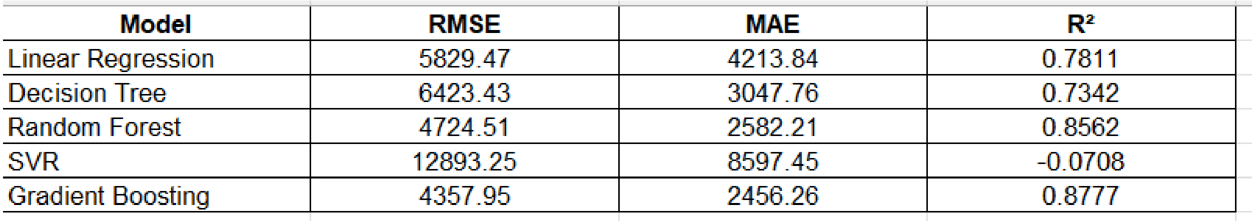
- n_estimators: [50, 100, 200]
- max_depth: [None, 10, 20]
- min_samples_split: [2, 5]

The best hyperparameters found were:

{'regressor__max_depth': 10, 'regressor__min_samples_split': 5, 'regressor__n_estimators': 50}

4. Results and Discussion

Each model was evaluated on the test set using RMSE, MAE, and R². An example summary of the results is shown in the table below:



Model	RMSE	MAE	R²
Linear Regression	5829.47	4213.84	0.7811
Decision Tree	6423.43	3047.76	0.7342
Random Forest	4724.51	2582.21	0.8562
SVR	12893.25	8597.45	-0.0708
Gradient Boosting	4357.95	2456.26	0.8777

Best hyperparameters for Random Forest:

{'regressor__max_depth': 10, 'regressor__min_samples_split': 5, 'regressor__n_estimators': 50}

Improved Random Forest model:

RMSE: 4658.92

MAE: 2539.15

R2: 0.86

Performance Analysis of Models and the Improved Random Forest Version

In this study, **Gradient Boosting demonstrated the best performance**, showing the lowest errors and the highest coefficient of determination. However, the **Improved Random Forest model** (after hyperparameter tuning) came significantly close to **Gradient Boosting** and exhibited a **good balance between accuracy and interpretability**.

Limitations and Potential Improvements

- **Feature Engineering**: Adding new features or transforming existing ones (e.g., interaction terms, polynomial features) could enhance model accuracy.
- **Cross-Validation**: Implementing more extensive cross-validation strategies would provide a **more reliable estimate** of model performance.
- **Hyperparameter Tuning for Other Models**: Future work could include optimizing hyperparameters for **SVR** and **Gradient Boosting** models to improve their effectiveness.

5. Conclusion

In this analysis, various regression models were applied to **predict insurance expenses**. **Gradient Boosting achieved the best results**, but the **Improved Random Forest model** also demonstrated **high accuracy** after hyperparameter optimization. These findings confirm that **ensemble methods** are **the most effective** for this type of regression task. Future research can focus on **advanced feature engineering** and **more extensive hyperparameter tuning** to achieve even greater accuracy.

6. References

- Scikit-learn documentation: https://scikit-learn.org/stable/documentation.html
- Pandas documentation: https://pandas.pydata.org/docs/
- Insurance dataset from the "Mathematics for Data Science" course (Assignment 5)