**Title: Predictive Modeling for Insurance Charges: A Comparative Analysis of Regression Models**

**Introduction**

Insurance companies often rely on predictive modeling techniques to estimate insurance charges for customers based on various factors such as age, gender, BMI, smoking habits, and region. In this study, we conducted a comparative analysis of three regression models: Decision Tree, Random Forest, and Support Vector Machine (SVM), to predict insurance charges using a real-world dataset.

**Dataset Description**

The dataset used in this study contains information about insurance charges and several demographic and lifestyle factors of individuals. It includes features such as age, sex, BMI, number of children, smoking habits, and region. The target variable is the insurance charges incurred by each individual.

age sex bmi children smoker region charges

0 19 female 27.900 0 yes southwest 16884.92400

1 18 male 33.770 1 no southeast 1725.55230

2 28 male 33.000 3 no southeast 4449.46200

3 33 male 22.705 0 no northwest 21984.47061

4 32 male 28.880 0 no northwest 3866.85520

**Data Preprocessing**

Before building the predictive models, we performed data preprocessing steps, including:

Label encoding categorical variables (sex, smoker, region).

Splitting the dataset into training and testing sets (80% training, 20% testing).

Standardizing the features using StandardScaler.

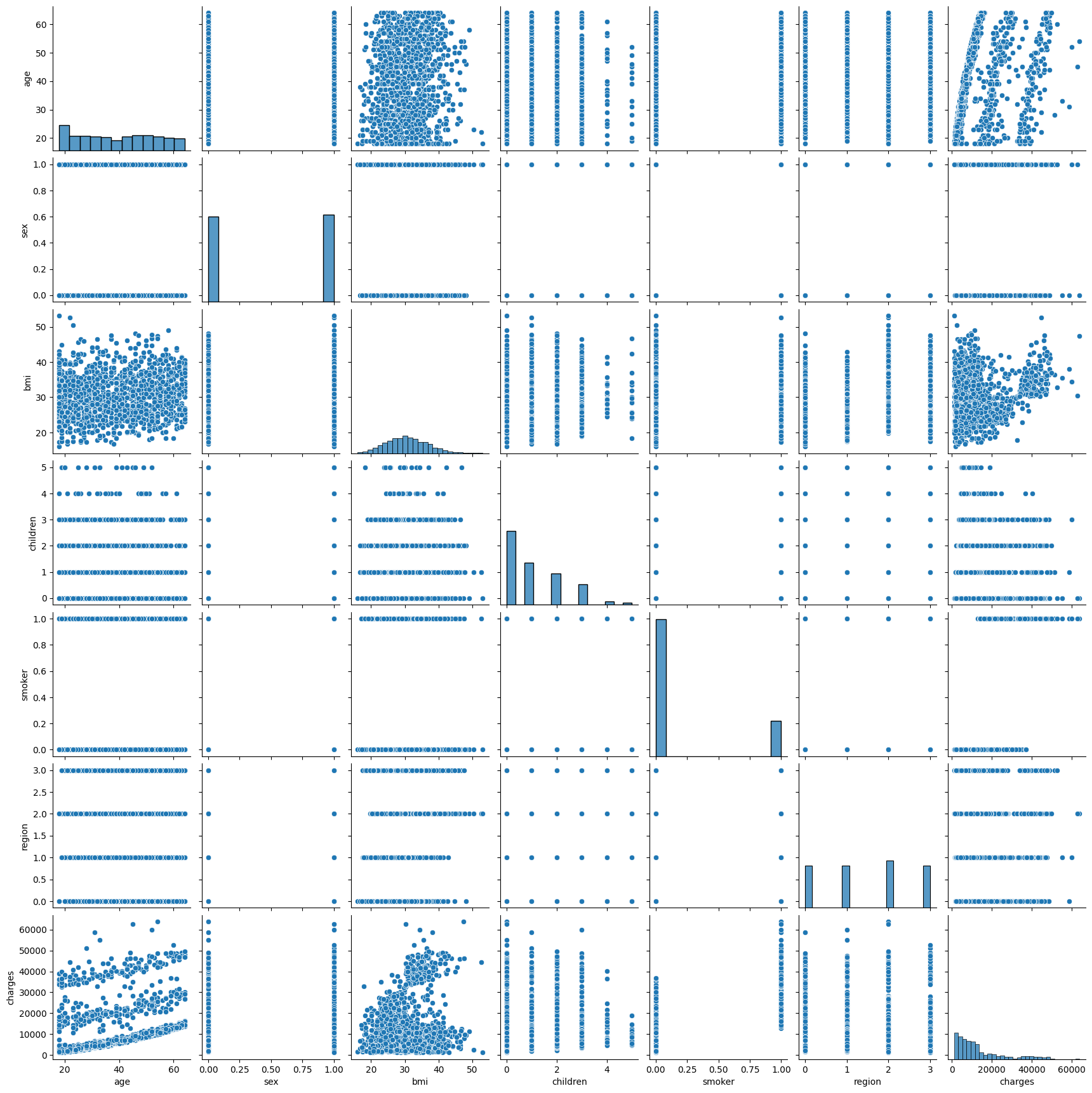
**Model Building and Evaluation**

We trained three regression models on the training data and evaluated their performance on the testing data using Mean Squared Error (MSE) and R-squared (R2) as evaluation metrics. The models evaluated are as follows:

1. Decision Tree Regression
2. Random Forest Regression
3. Support Vector Machine (SVM) Regression

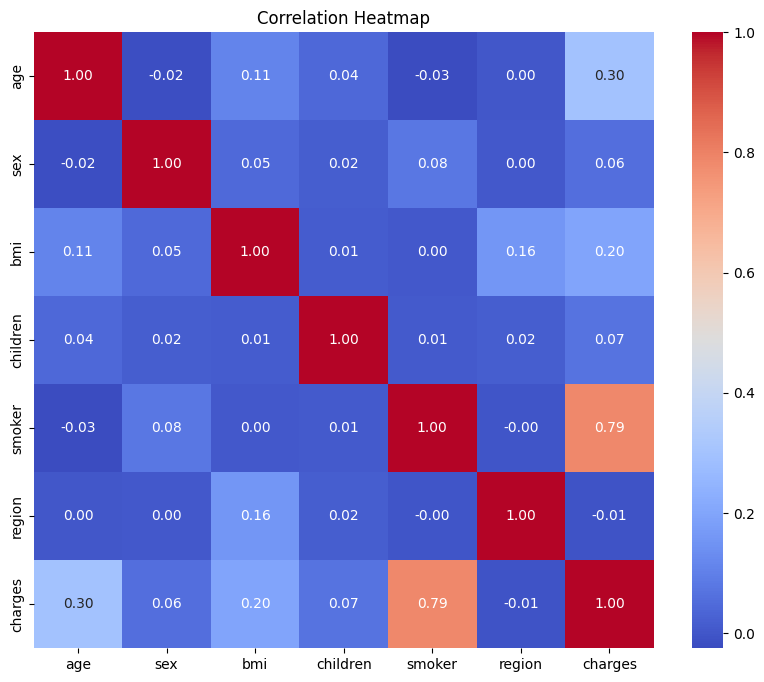
**Data Visualization**

Before building the predictive models, we explored the data using various visualizations to gain insights into the relationships between variables and the distribution of data. Here are some of the visualizations we created:

****

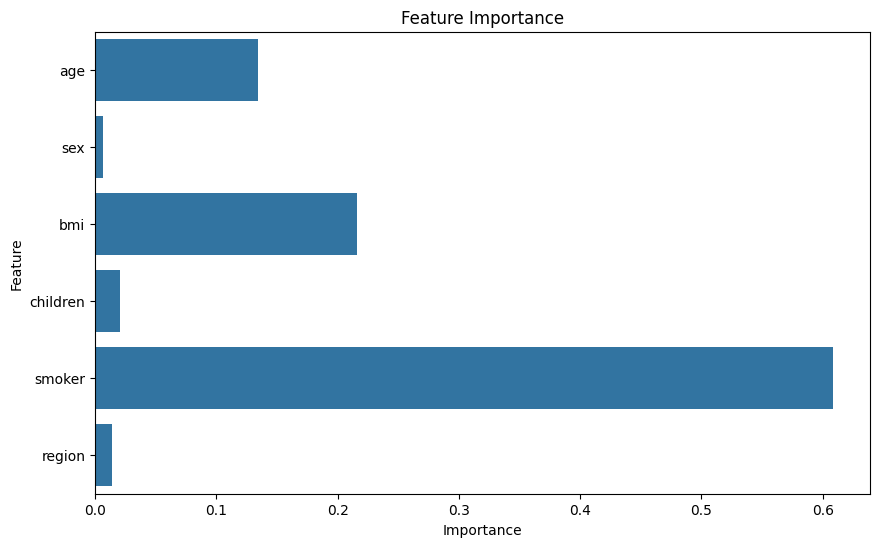
**Figure 1: Pair plot**

A pair plot allows us to visualize relationships between numerical variables and identify potential patterns or correlations.

****

**Figure 2: Correlation Heat Map**

We created a heat map to visualize the correlation between numerical variables, helping us understand the strength and direction of relationships.

****

**Figure 3: Feature Importance Plot**

We Plotted the feature importance scores generated by the Random Forest model to identify the most important features driving insurance charges.

**Table 1: Presentation of Results**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Decision Tree Regression | Random Forest Regression | Support Vector Machine (SVM) Regression |
| Mean Squared Error | 47,349,691.41 | 20,898,625.74 | 165,839,509.92 |
| R-squared | 0.695 | 0.865 | -0.068 |

**Discussion of Findings**

Random Forest Regression outperformed the other models with the lowest MSE and highest R-squared. Decision Tree Regression showed good performance but had higher MSE compared to Random Forest. SVM Regression had the highest MSE and a negative R-squared, indicating its limited effectiveness in this context.

**Conclusion**

In this study, we conducted a comparative analysis of regression models for predicting insurance charges. The results suggest that Random Forest Regression is the most suitable model for this task, followed by Decision Tree Regression. These findings can help insurance companies improve their pricing strategies and better estimate insurance charges for customers.

**Future Directions**

Experiment with more advanced regression techniques such as Gradient Boosting and Neural Networks. Incorporate additional features or external data sources to enhance model performance. Explore interpretability techniques to gain insights into the factors driving insurance charges.

***Codes***

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.tree import DecisionTreeRegressor

from sklearn.ensemble import RandomForestRegressor

from sklearn.svm import SVR

from sklearn.metrics import mean\_squared\_error, r2\_score

# Load the dataset into a Pandas DataFrame

file\_path = '/content/insurance.csv.xls'  # Update this path according to your file location

df = pd.read\_csv(file\_path)

print(df.head())

label\_encoder = LabelEncoder()

df['sex'] = label\_encoder.fit\_transform(df['sex'])

df['smoker'] = label\_encoder.fit\_transform(df['smoker'])

df['region'] = label\_encoder.fit\_transform(df['region'])

# Split the data into features (X) and target variable (y)

X = df.drop('charges', axis=1)  # Features

y = df['charges']  # Target variable

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Standardize the features using StandardScaler

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

# Train and evaluate Decision Tree model

decision\_tree = DecisionTreeRegressor(random\_state=42)

decision\_tree.fit(X\_train\_scaled, y\_train)

y\_pred\_dt = decision\_tree.predict(X\_test\_scaled)

mse\_dt = mean\_squared\_error(y\_test, y\_pred\_dt)

r2\_dt = r2\_score(y\_test, y\_pred\_dt)

print("Decision Tree - Mean Squared Error:", mse\_dt)

print("Decision Tree - R-squared:", r2\_dt)

# Train and evaluate Random Forest model

random\_forest = RandomForestRegressor(random\_state=42)

random\_forest.fit(X\_train\_scaled, y\_train)

y\_pred\_rf = random\_forest.predict(X\_test\_scaled)

mse\_rf = mean\_squared\_error(y\_test, y\_pred\_rf)

r2\_rf = r2\_score(y\_test, y\_pred\_rf)

print("Random Forest - Mean Squared Error:", mse\_rf)

print("Random Forest - R-squared:", r2\_rf)

# Train and evaluate Support Vector Machine (SVM) model

svm = SVR()

svm.fit(X\_train\_scaled, y\_train)

y\_pred\_svm = svm.predict(X\_test\_scaled)

mse\_svm = mean\_squared\_error(y\_test, y\_pred\_svm)

r2\_svm = r2\_score(y\_test, y\_pred\_svm)

print("Support Vector Machine (SVM) - Mean Squared Error:", mse\_svm)

print("Support Vector Machine (SVM) - R-squared:", r2\_svm)

import seaborn as sns

import matplotlib.pyplot as plt

# Pairplot to visualize relationships between variables

sns.pairplot(df)

plt.show()

# Correlation heatmap

plt.figure(figsize=(10, 8))

sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f")

plt.title('Correlation Heatmap')

plt.show()

# Feature Importance (for Random Forest)

feature\_importance = random\_forest.feature\_importances\_

plt.figure(figsize=(10, 6))

sns.barplot(x=feature\_importance, y=X.columns)

plt.title('Feature Importance')

plt.xlabel('Importance')

plt.ylabel('Feature')

plt.show()