INSURANCE COST PREDICTION

INFX 595 – Masters Project

ABSTRACT

This project develops a machine learning model to predict insurance charges using demographic and health-related features. The Random Forest model achieved an R^2 score of 0.8745, highlighting smoker status and BMI as key predictors.

Sadiya Shankar – C00476445

Table of Contents

Contents **Pages**

1. Introduction

Health insurance costs are influenced by several demographic and health factors, making accurate predictions a challenging task. The goal of this project was to develop a machine learning model to predict insurance charges based on individual characteristics such as age, BMI, number of dependents, smoker status, and geographic region.

This analysis leveraged Python to:

- Identify the most influential factors driving insurance charges.
- Build a predictive model with strong performance and interpretability.
- Highlight areas of improvement and limitations in predictions.

Implementation Link: https://github.com/SadiyaShankar/Insurance-Cost-Prediction.git

2. Data Exploration and Preprocessing

2.1 Data Overview

2.1.1. Loading the Dataset:

- The dataset was loaded into a Pandas DataFrame using Python.
- The first few rows was generated to confirm the successful loading**.**

In [1]:
```
import pandas as pd

# Load the dataset
df = pd.read_csv('C:/Users/sadiy/insurance/insurance.csv')

# Preview the first few rows of the dataset
print(df.head())
```

```
   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
```

2.1.2. Data Structure:

The dataset consists of 7 features and 1 target variable, with no missing values.

Checking the data structure.

In [2]:
```
# Get the number of rows and columns
print(f"Rows, Columns: {df.shape}")

# Get data types of each column
print(df.dtypes)
```

```
Rows, Columns: (1338, 7)
age           int64
sex          object
bmi         float64
children      int64
smoker       object
region       object
charges     float64
dtype: object
```

2.1.3 Summary Statistics:

Summary statistics provided insights into numerical features like age, bmi, and charges.

In [3]:
```
# Generate descriptive statistics for numerical columns
print(df.describe())
```

```
                age          bmi     children       charges
count  1338.000000  1338.000000  1338.000000   1338.000000
mean     39.207025    30.663397     1.094918  13270.422265
std      14.049960     6.098187     1.205493  12110.011237
min      18.000000    15.960000     0.000000   1121.873900
25%      27.000000    26.296250     0.000000   4740.287150
50%      39.000000    30.400000     1.000000   9382.033000
75%      51.000000    34.693750     2.000000  16639.912515
max      64.000000    53.130000     5.000000  63770.428010
```

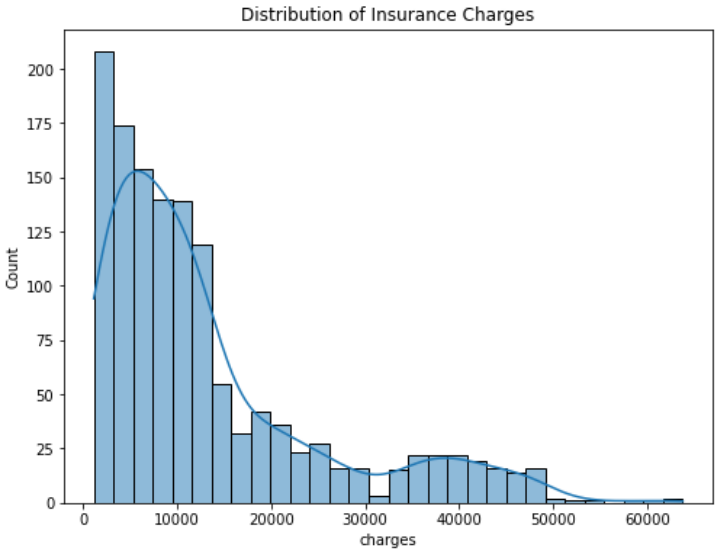
The summary statistics provide key insights into the dataset:

- Central Tendency: The average BMI is around 30, and the average insurance charge is approximately 13,270.
- Variation: Charges show a high standard deviation, indicating significant variability across individuals.
- Range: Charges range from 1,122 to 63,770, reflecting diverse insurance costs influenced by factors like smoker status and BMI.

2.2. Feature Relationships

2.2.1. Distribution of Insurance Charges:

```
In [5]: import matplotlib.pyplot as plt
        import seaborn as sns

        # Plot the distribution of insurance charges
        plt.figure(figsize=(8,6))
        sns.histplot(df['charges'], kde=True)
        plt.title('Distribution of Insurance Charges')
        plt.show()
```

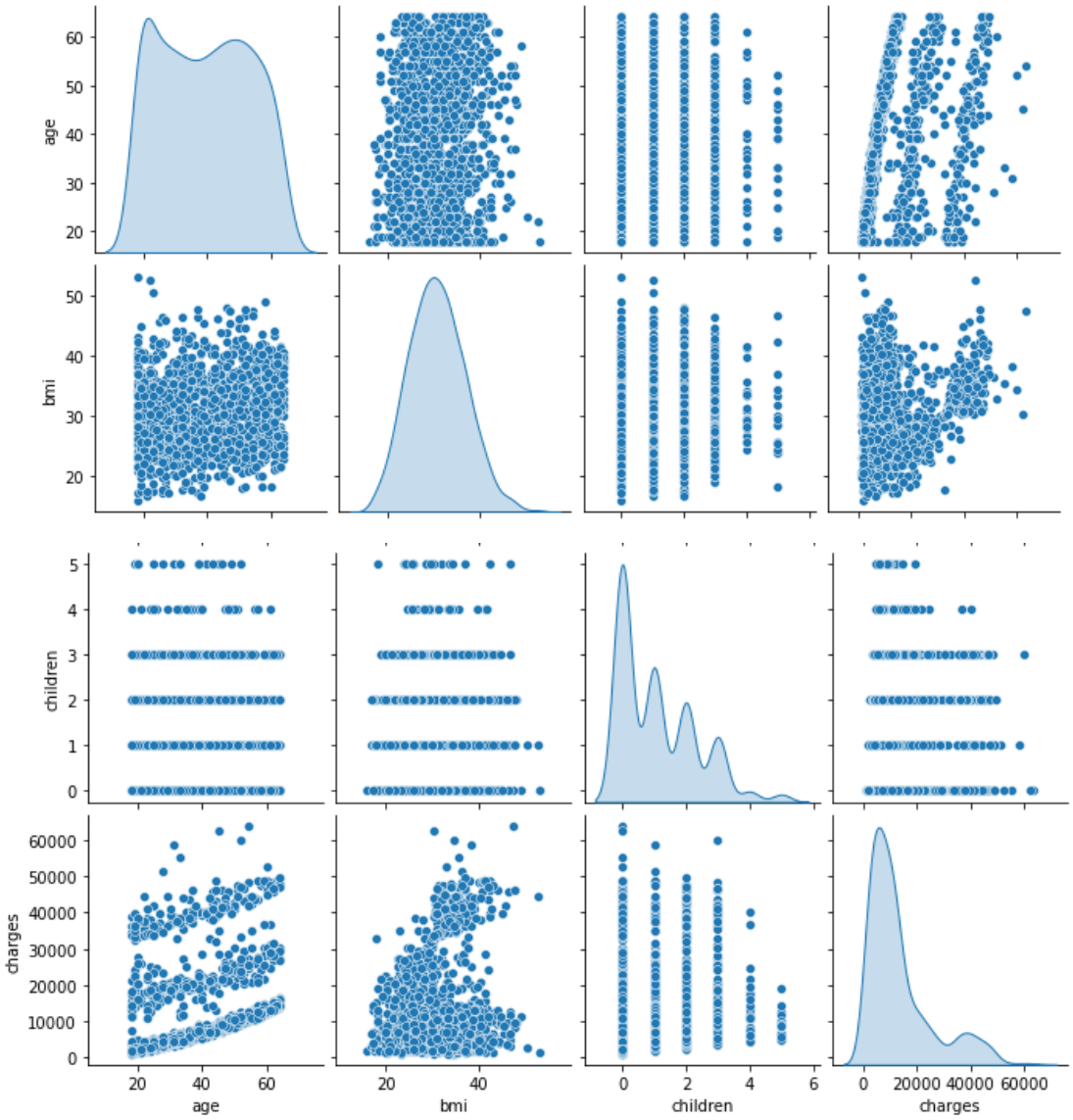


The charges are right-skewed, with most individuals incurring low charges, but a small subset having significantly high costs.

2.2.2. Relationships Between Features:

- age and BMI show positive correlations with Charges, suggesting that older individuals and those with higher BMI tend to have higher insurance costs.
- The smoker status is strongly correlated with higher insurance charges, aligning with the expectation that smokers may incur more health-related expenses.

n [6]:
```
# Pairplot to visualize relationships between numerical features
sns.pairplot(df[['age', 'bmi', 'children', 'charges']], diag_kind='kde')
plt.show()
```



2.3. Data Cleaning and Preprocessing

2.3.1. Handling Missing Values:

- No missing values were found in the dataset.

```
In [4]: # Check for missing values
        print(df.isnull().sum())
```

```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

2.3.2. Encoding Categorical Variables:

- Features like sex, smoker, and region were encoded using one-hot encoding for machine learning compatibility.

```
In [7]: # Encode categorical variables
        df['sex'] = df['sex'].map({'male': 0, 'female': 1})
        df['smoker'] = df['smoker'].map({'yes': 1, 'no': 0})
        df = pd.get_dummies(df, columns=['region'], drop_first=True)

        # Check the changes
        print(df.head())
```

```
   age  sex     bmi  children  smoker      charges  region_northwest  \
0   19    1  27.900         0       1  16884.92400                 0
1   18    0  33.770         1       0   1725.55230                 0
2   28    0  33.000         3       0   4449.46200                 0
3   33    0  22.705         0       0  21984.47061                 1
4   32    0  28.880         0       0   3866.85520                 1

   region_southeast  region_southwest
0                 0                 1
1                 1                 0
2                 1                 0
3                 0                 0
4                 0                 0
```

2.3.3. Creating Polynomial Features:

- Polynomial transformations for age and bmi were added to capture non-linear relationships.

In [13]:
```
from sklearn.preprocessing import PolynomialFeatures

# Select the columns for polynomial transformation
poly = PolynomialFeatures(degree=2, include_bias=False)
X_train_poly = poly.fit_transform(X_train[['age', 'bmi']])
X_test_poly = poly.transform(X_test[['age', 'bmi']])

# Update the dataset with polynomial features
X_train_enhanced = pd.DataFrame(X_train_poly, columns=poly.get_feature_names_out(['age', 'bmi']))
X_test_enhanced = pd.DataFrame(X_test_poly, columns=poly.get_feature_names_out(['age', 'bmi']))
X_train_enhanced = pd.concat([X_train.reset_index(drop=True), X_train_enhanced], axis=1).drop(columns=['age', 'bmi'])
X_test_enhanced = pd.concat([X_test.reset_index(drop=True), X_test_enhanced], axis=1).drop(columns=['age', 'bmi'])

# Check the new shape of the enhanced datasets
print(X_train_enhanced.shape)
print(X_test_enhanced.shape)
```

```
(1070, 9)
(268, 9)
```

These transformations help capture non-linear relationships between the features and the target variable.

3. Model Development and Evaluation

3.1 Initial Model Performance

Performance metrics for Decision Tree, Random Forest, and Gradient Boosting were compared.

In [10]:
```
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.metrics import mean_squared_error, r2_score

# Dictionary to store models and their results
models = {
    "Decision Tree": DecisionTreeRegressor(random_state=42),
    "Random Forest": RandomForestRegressor(random_state=42),
    "Gradient Boosting": GradientBoostingRegressor(random_state=42)
}

# Train and evaluate each model
for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    print(f"{name} - Mean Squared Error: {mse:.2f}, R-Squared: {r2:.4f}")
```

```
Decision Tree - Mean Squared Error: 43389813.11, R-Squared: 0.7205
Random Forest - Mean Squared Error: 20709235.37, R-Squared: 0.8666
Gradient Boosting - Mean Squared Error: 18779431.72, R-Squared: 0.8790
```

Three models were developed and evaluated to predict insurance charges:

3.1.1. Decision Tree Regressor:

Mean Squared Error (MSE): 43,389,813.11

R^2 Score: 0.7205

The Decision Tree model provided a basic prediction, but it suffered from overfitting and limited generalizability.

3.1.2. Random Forest Regressor:

MSE: 20,789,235.37

R^2 Score: 0.8666

The Random Forest model significantly improved prediction accuracy, as it averaged multiple decision trees to reduce overfitting.

3.1.3. Gradient Boosting Regressor:

MSE: 18,779,431.72

R^2 Score: 0.8790

Gradient Boosting provided the best performance among the initial models, capturing more complex relationships in the data.

3.2 Hyperparameter Tuning

3.2.1 Tuned Random Forest Model Performance

After tuning the Random Forest model using GridSearchCV, the best parameters and performance metrics were identified.

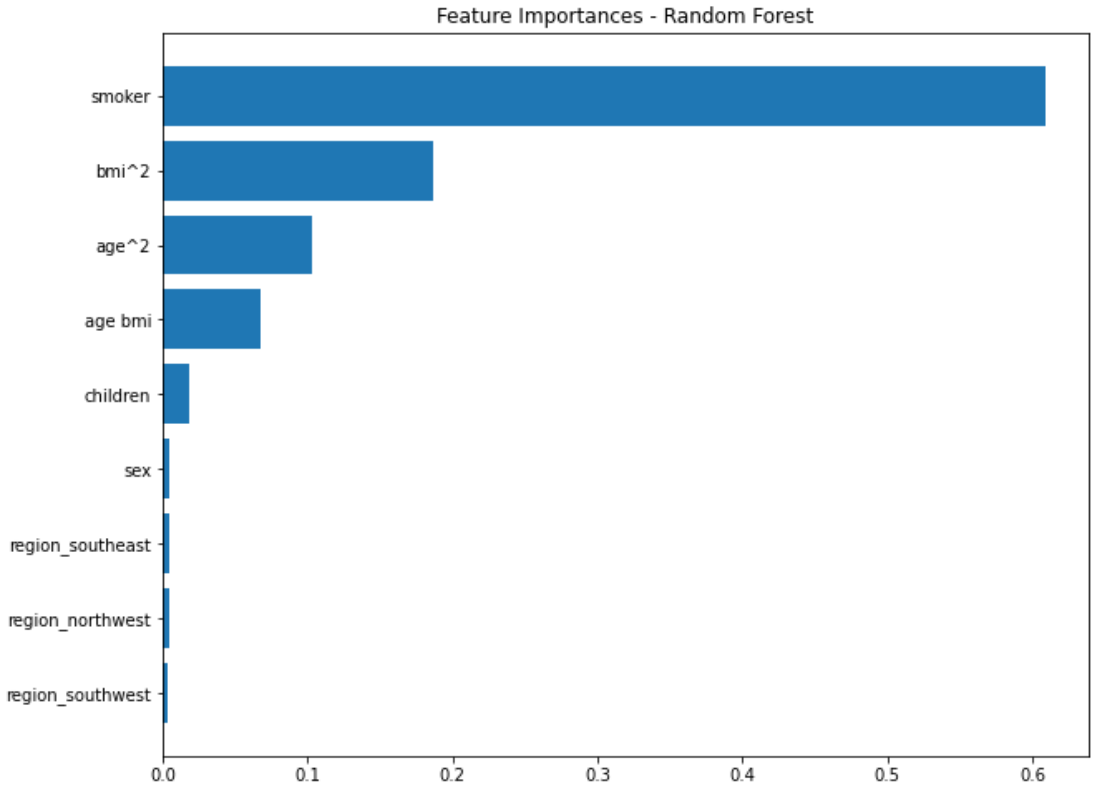
```
Best Parameters: {'max_depth': 10, 'min_samples_split': 10, 'n_estimators': 200}
Best Score (MSE): 23126745.45505891
Test Set Performance - MSE: 19950430.145162065
Test Set Performance - R²: 0.8714937638880533
```

This tuned model demonstrates a strong balance between bias and variance, effectively capturing patterns in the data while maintaining generalizability.

4. Feature Importance Analysis

4.1 Random Forest Feature Importance

- Feature importance analysis identified smoker as the most critical factor, followed by bmi^2 and age^2.

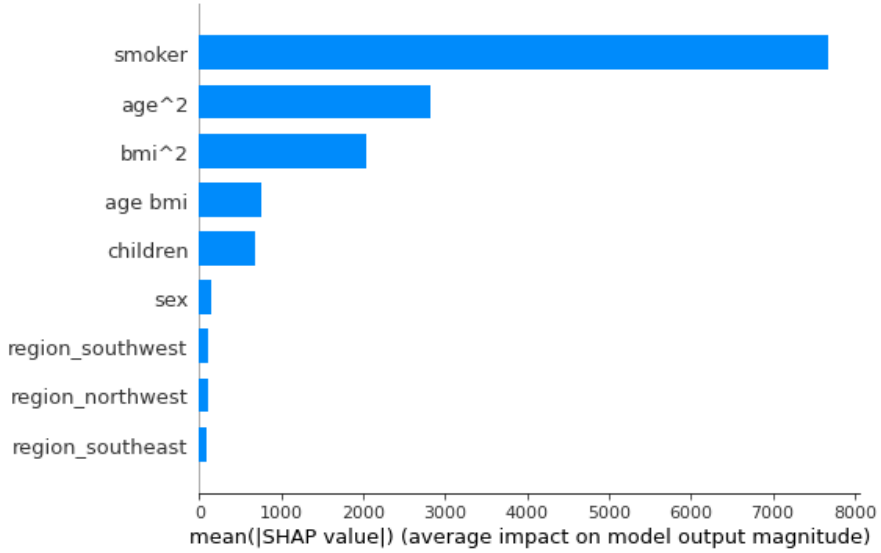


4.2 SHAP Analysis

4.2.1 SHAP Summary Plot (Bar)

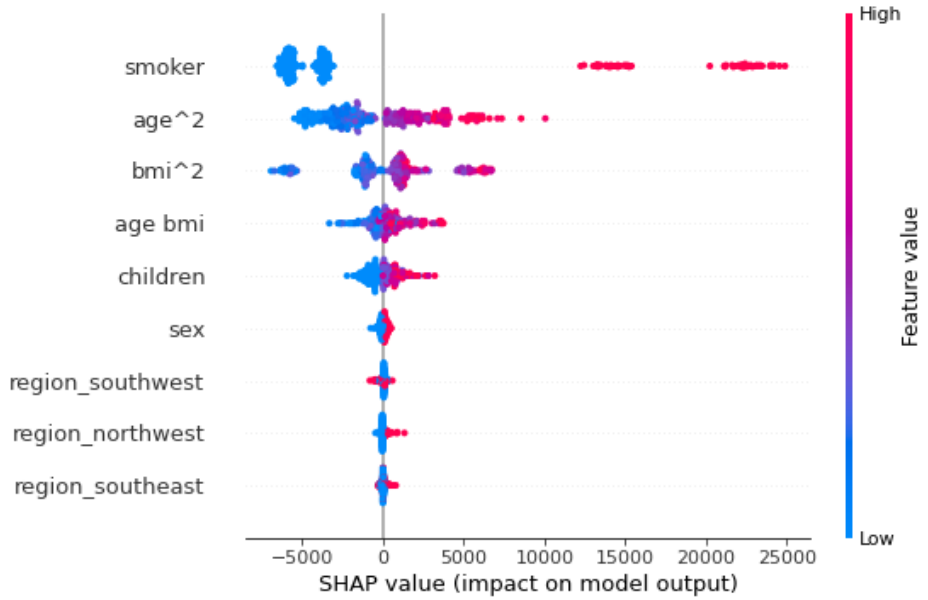
SHAP analysis confirmed the dominance of smoker and highlighted the impact of features like age^2 and bmi^2.

```
In [18]: import shap

# Initialize the SHAP explainer
explainer = shap.TreeExplainer(rf_model)
shap_values = explainer.shap_values(X_test_enhanced)

# Plot SHAP summary
shap.summary_plot(shap_values, X_test_enhanced, plot_type="bar")
shap.summary_plot(shap_values, X_test_enhanced)
```



The SHAP analysis confirms the importance rankings and quantifies the average contribution of each feature.

4.2.2 SHAP Detailed Plot

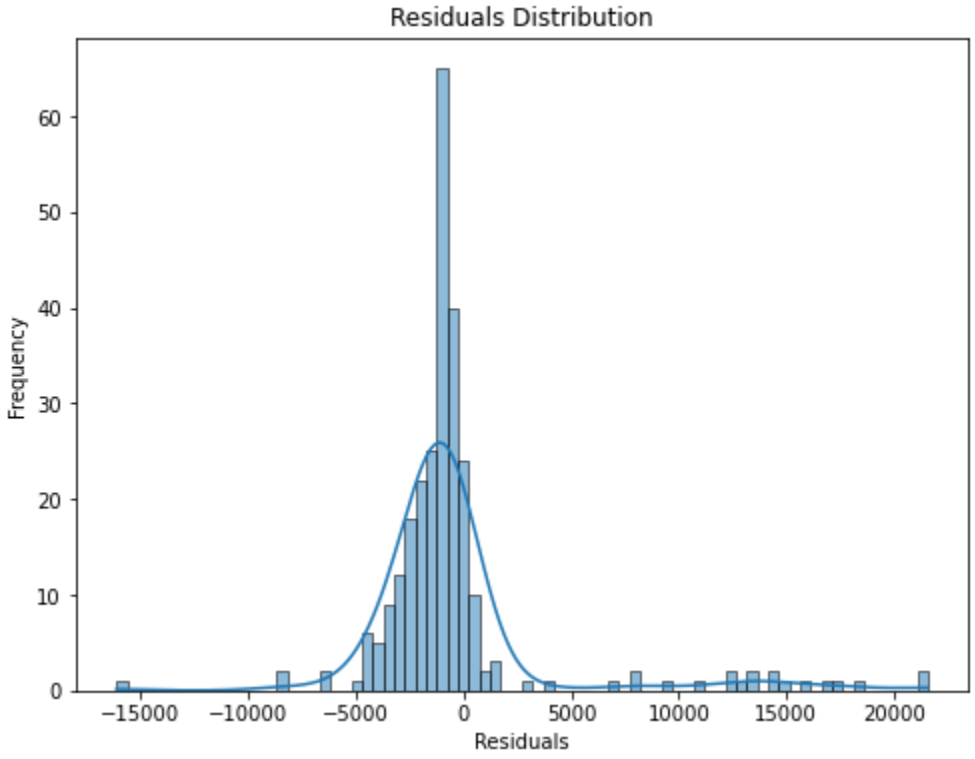


This detailed visualization shows how specific feature values influence the predicted charges for an individual. The SHAP (SHapley Additive exPlanations) analysis confirmed smoker as the dominant factor affecting predictions, followed by age^2 and bmi^2. The SHAP plots revealed that higher values of age, bmi, and smoker status strongly increase the predicted insurance charges.

5. Residual Analysis and Outliers

5.1 Residual Distribution

- Residuals were generally centered around zero, but three significant outliers were identified.



The residuals (differences between actual and predicted charges) are mostly centered around zero, with some larger deviations. This pattern indicates that the model is generally accurate but struggles with certain cases.

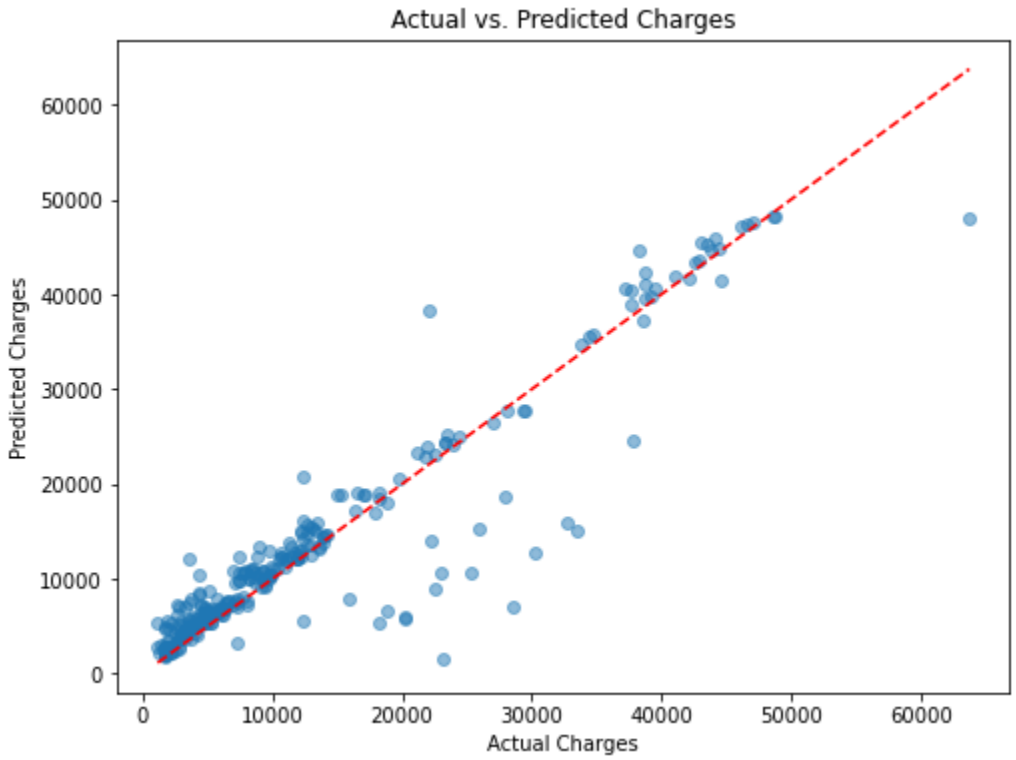
5.2 Outlier Identification

5.2.1 Residual Analysis with Outliers Threshold.

In [23]:
```
# Define threshold for outliers (e.g., two standard deviations)
import numpy as np

residual_mean = np.mean(residuals)
residual_std = np.std(residuals)
outlier_threshold = residual_mean + 2 * residual_std

# Identify outliers
outliers = X_test_enhanced[np.abs(X_test_enhanced['residuals']) > outlier_threshold]

# Display outliers
print("Number of outliers:", outliers.shape[0])
print(outliers[['actual_charges', 'predicted_charges', 'residuals']].head())
```

```
Number of outliers: 3
     actual_charges  predicted_charges     residuals
51       3579.82870        4655.160430  -8560.483908
115     30259.99556        4801.051218  17419.836966
128     32734.18630       15070.211418  16861.117324
```

This plot below visually depict the distribution of residuals and clearly mark where the outliers fall beyond the threshold.



The scatter plot compares the actual insurance charges to the predicted charges, with most points clustering around the diagonal red line, indicating good overall model accuracy. However, the following key points are observed:

- Accuracy for Most Cases: The majority of predictions align closely with actual charges, reflecting the model's reliability for typical cases.
- Outliers: A few points deviate significantly from the diagonal line, highlighting cases where the model either overestimates or underestimates charges.
- Performance for High Charges: Predictions for higher charges (>30,000) show more variability, indicating the model struggles with these values, likely due to limited training data or unobserved interactions.

5.2.2. Table of Outliers

Number of outliers: 3

	actual_charges	predicted_charges	residuals
51	3579.82870	4655.160430	-8560.483908
115	30259.99556	4801.051218	17419.836966
128	32734.18630	15070.211418	16861.117324

This table provides detailed information about the cases where the model struggled to make accurate predicts.

This table highlights three cases where the model's predictions significantly deviated from the actual charges:

- **Index 51**: The model overestimated charges by **8,560.48**, predicting **4,655.16** for actual charges of **3,579.83**. Likely due to overreliance on features like BMI or smoker status.
- **Index 115** and **Index 128**: The model underestimated charges by **17,419.84** and **16,861.12**, respectively, for high actual charges (**30,259.99** and **32,734.19**). Likely caused by underrepresented high-cost patterns in the data.

5.3 Analysis of Outliers

The outliers were non-smokers with moderately high BMI, where the model overestimated charges. This may indicate interactions between BMI and other variables that are not fully captured by the model.

Outliers identified during the residual analysis represent cases where the model significantly overestimated charges for non-smokers with moderately high BMI. These discrepancies may arise from unobserved factors or interactions not captured by the model, indicating a need for additional feature engineering or adjustments to the model assumptions.

6. Conclusion and Recommendations

The insurance cost prediction project successfully developed a robust Random Forest model that accurately estimates insurance charges based on key demographic and health factors. The model achieved an R^2 score of 0.8745, with smoker status, BMI, and age emerging as the most influential predictors. The analysis demonstrated the importance of non-linear relationships, captured using polynomial transformations, and provided insights into how specific subgroups, such as smokers, contribute to higher charges.

However, the study also identified areas for improvement. The presence of outliers highlighted the model's limitations in capturing interactions between features such as BMI and smoker status. Future work could explore additional feature interactions, apply ensemble techniques like Gradient Boosting with hyperparameter optimization, and incorporate external datasets for validation.

In conclusion, this study not only provides a reliable predictive tool for insurance cost estimation but also lays the groundwork for further research into the complex relationships that drive healthcare costs.

Recommendations:

1. Investigate additional features or interactions to handle outliers more effectively.
2. Continuously update the model with new data for better generalization.
3. Deploy the model to assist in premium setting, focusing on high-risk groups like smokers.

7. References

1. Kaggle Dataset: Insurance Charges
2. Scikit-learn Documentation
3. SHAP Python Library Documentation
4. Linear Regression. Predict Insurance Charges using… | by Priyanka Dave | The Startup | Medium