1009-INSURANCE PREMIUM PREDICTION

Team Nash

- Amirtesh Raghuram
- Nigam Parida
- Shreya Pattanayak
- Harish Sathyanandan

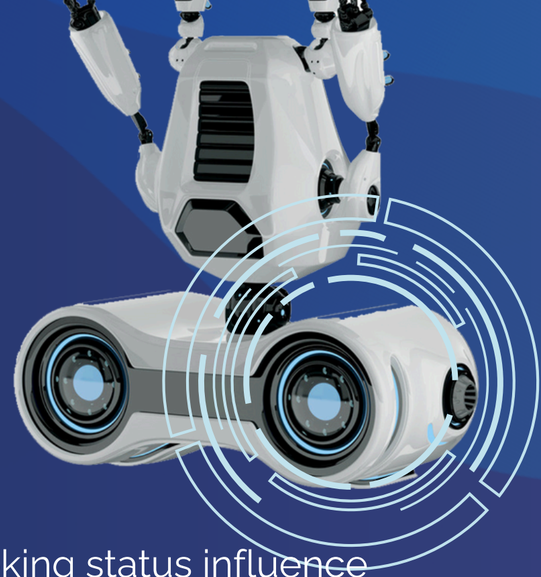
Transforming the Future

Guide’s Name

Nikhil Maurya

02

Goal



1. Risk Assessment: By analyzing how various factors like age, BMI, and smoking status influence insurance premiums, insurers can better assess risk profiles and set appropriate premium rates.

2. Personalized Pricing: Developing predictive models allows for more tailored insurance plans that reflect individual characteristics, leading to fairer pricing for customers based on their unique risk factors.

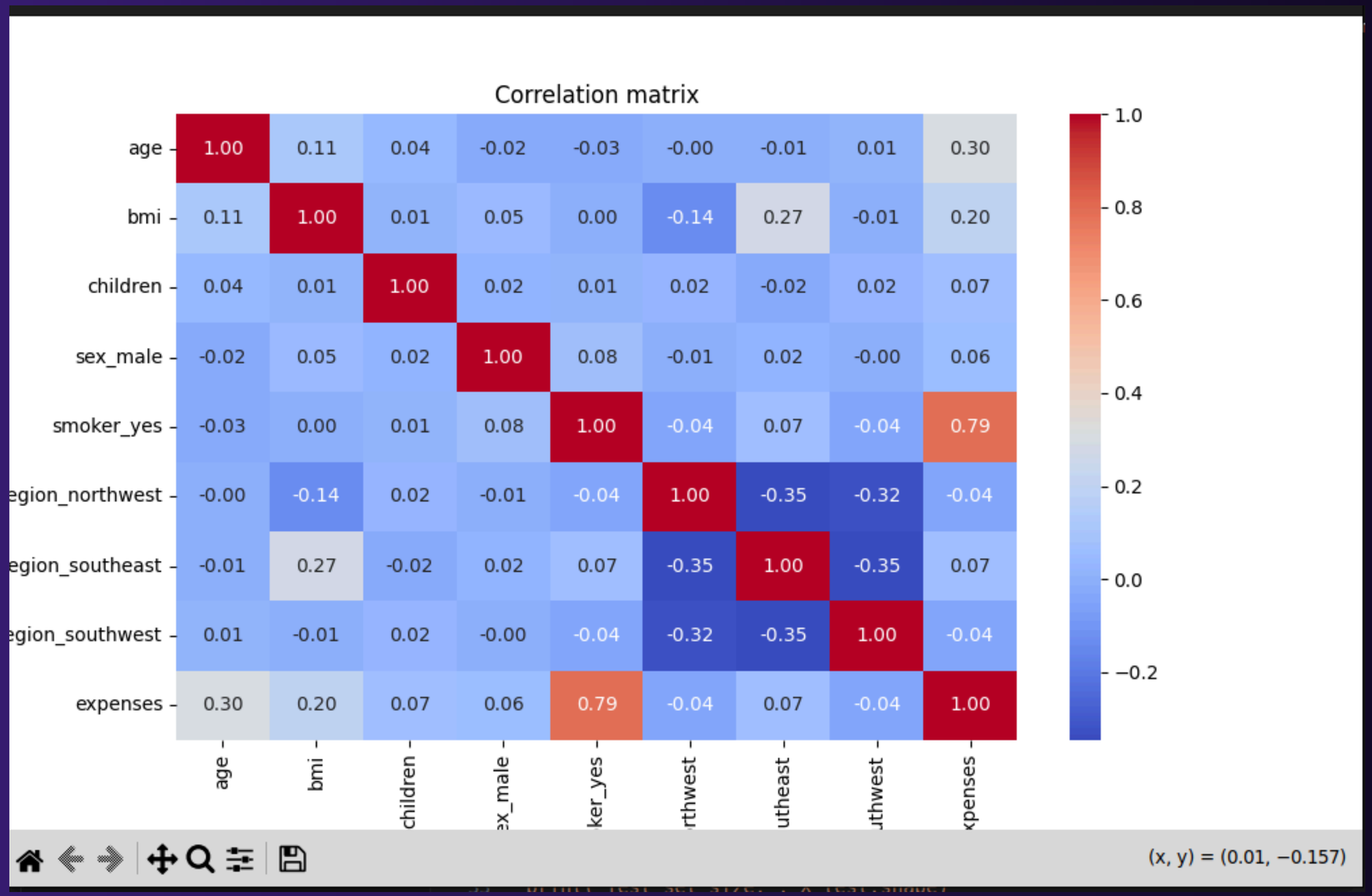
3. Operational Efficiency: Accurate premium prediction improves operational efficiency for insurance companies by streamlining the underwriting process, making it quicker and more cost-effective.

4. Market Competitiveness: Companies that leverage data-driven insights can remain competitive in the insurance market, attracting customers with fairer and more transparent pricing models based on predictive analytics.

5. Improved Customer Satisfaction: By providing accurate quotes and tailored insurance products, the project can enhance customer satisfaction and trust, fostering long-term relationships between insurers and clients.

Exploratory Data Analysis

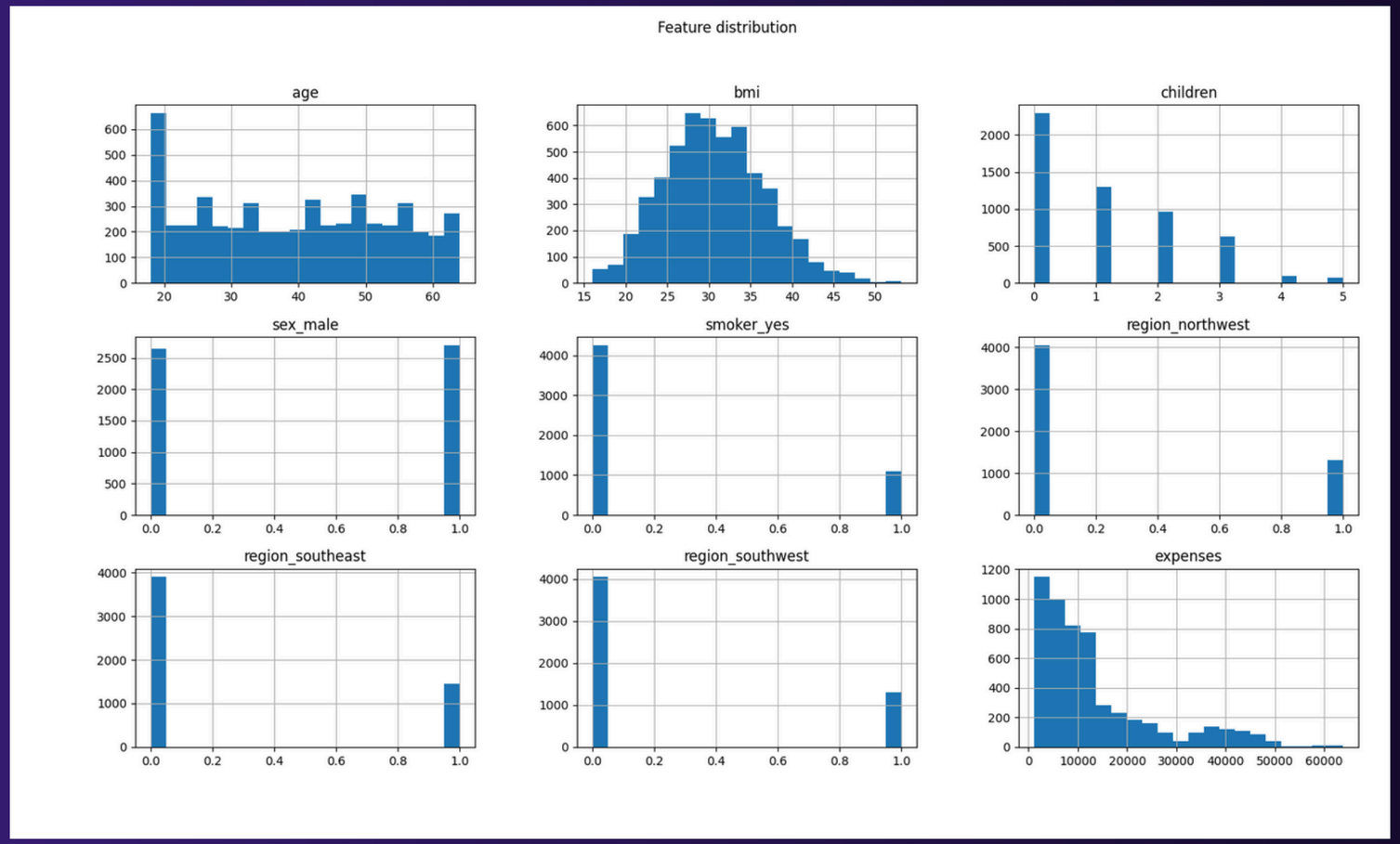
Basic Information



This correlation matrix visualizes relationships between different features in a dataset:

- Smoker_yes and expenses (0.79): Strong positive correlation, indicating that smokers tend to have much higher expenses.
- Age and expenses (0.30): Moderate positive correlation, suggesting that older people have higher expenses.
- BMI and expenses (0.20): Moderate positive correlation, indicating that individuals with higher BMI also have higher expenses.
- Other correlations, like between regions and expenses, are weak, showing minimal relationships. Most other variables have weak or no significant correlations.

Exploratory Graphs



The image shows several histograms representing the distribution of features in a dataset. Key takeaways include:

- Age: Even distribution with slightly more individuals around the age of 20 and 60.
- BMI: Normal distribution with most values between 25 and 35.
- Children: Most individuals have 0 or 1 child, with fewer people having more children.
- Sex: More males than females in the dataset.
- Smoker: Most individuals are non-smokers.
- Regions: Certain regions, like the Southeast and Northwest, have a higher representation.
- Expenses: Skewed distribution, with most people having lower expenses, but a few have high expenses.

These distributions provide insights into health and insurance-related variables.

Data Understanding

The dataset contains health insurance information for individuals, including their age, sex, BMI, number of children, smoking status, region, and medical expenses. Key insights can be drawn by analyzing how factors such as BMI, smoking habits, and age influence medical costs. Smokers generally tend to have significantly higher medical expenses compared to non-smokers. Similarly, individuals with a higher BMI may incur more expenses due to associated health risks. The dataset allows for a comprehensive understanding of how lifestyle factors and demographics impact insurance costs, which could inform future health insurance policies or preventive healthcare strategies.

Training of the Model

To create a well performing model we have tried to train the model on various classes present in scikit learn module of Python and XGBoost module of Python. The following models were used:

- Linear regression
- KNN Regressor
- Random Forest Regressor
- XGB Regressor
- Voting Regressor

Parameters Used in Every Model

1.Linear Regression: Proceeded with default parameters parameters

2.KNN Regression: n_neighbors=2

3.Random Forest Regression: n_estimators=300, max_depth=10

4.XGBoost Regressor: n_estimators=1000, learning_rate=0.01, max_depth=10, gamma=0.1

5.Voting Regressor: estimators=[('knn', knn_model), ('rf', rf_model), ('xgb', xgb_model)],weights=[1,1,1]

Training Progress

```
Linear regression results on training data:
Mean absolute error: 0.34392887
Mean square error: 0.24709422
Root mean squared error: 0.49708572
R2 score: 0.75290578

Linear regression results on testing data:
Mean absolute error: 0.34027074
Mean square error: 0.24597460
Root mean squared error: 0.49595827
R2 score: 0.74207573
```

```
Random forest results on training data:
Mean absolute error: 0.06312431
Mean square error: 0.01893405
Root mean squared error: 0.13760106
R2 score: 0.98106595

Random forest results on testing data:
Mean absolute error: 0.08125287
Mean square error: 0.03204107
Root mean squared error: 0.17900021
R2 score: 0.97094148
```

```
Voting regressor results on training data:
Mean absolute error: 0.04424650
Mean square error: 0.00808906
Root mean squared error: 0.08993923
R2 score: 0.99191094

Voting regressor results on testing data:
Mean absolute error: 0.06092070
Mean square error: 0.01507375
Root mean squared error: 0.12277520
R2 score: 0.98457666
```

```
KNN results on training data :
Mean absolute error: 0.00357744
Mean square error: 0.00271602
Root mean squared error: 0.05211546
R2 score: 0.99728398

KNN results on testing data:
Mean absolute error: 0.01939832
Mean square error: 0.01106513
Root mean squared error: 0.10519092
R2 score: 0.98947117
```

```
XGBoost results on training data:
Mean absolute error: 0.07218220
Mean square error: 0.01650569
Root mean squared error: 0.12847448
R2 score: 0.98349431

XGBoost results on testing data:
Mean absolute error: 0.08996176
Mean square error: 0.03017756
Root mean squared error: 0.17371690
R2 score: 0.96665006
```

Process of Evaluation

1. MAE (Mean Absolute Error): This measures the average of the absolute differences between predicted and actual values. It gives an idea of how far, on average, the predictions are from the true values. The benefit is that it's easy to interpret since it's in the same units as the target variable. However, it doesn’t penalize large errors as much as other metrics.

2. MSE (Mean Squared Error): This calculates the average of the squared differences between the predicted and actual values. By squaring the errors, MSE heavily penalizes larger mistakes, making it sensitive to outliers. It’s useful when you want to focus on minimizing large errors.

3. RMSE (Root Mean Squared Error): This is the square root of the MSE. It restores the scale of the errors to the same unit as the target variable, which makes it more interpretable. RMSE still penalizes large errors more heavily but provides a more balanced view of overall error magnitude.

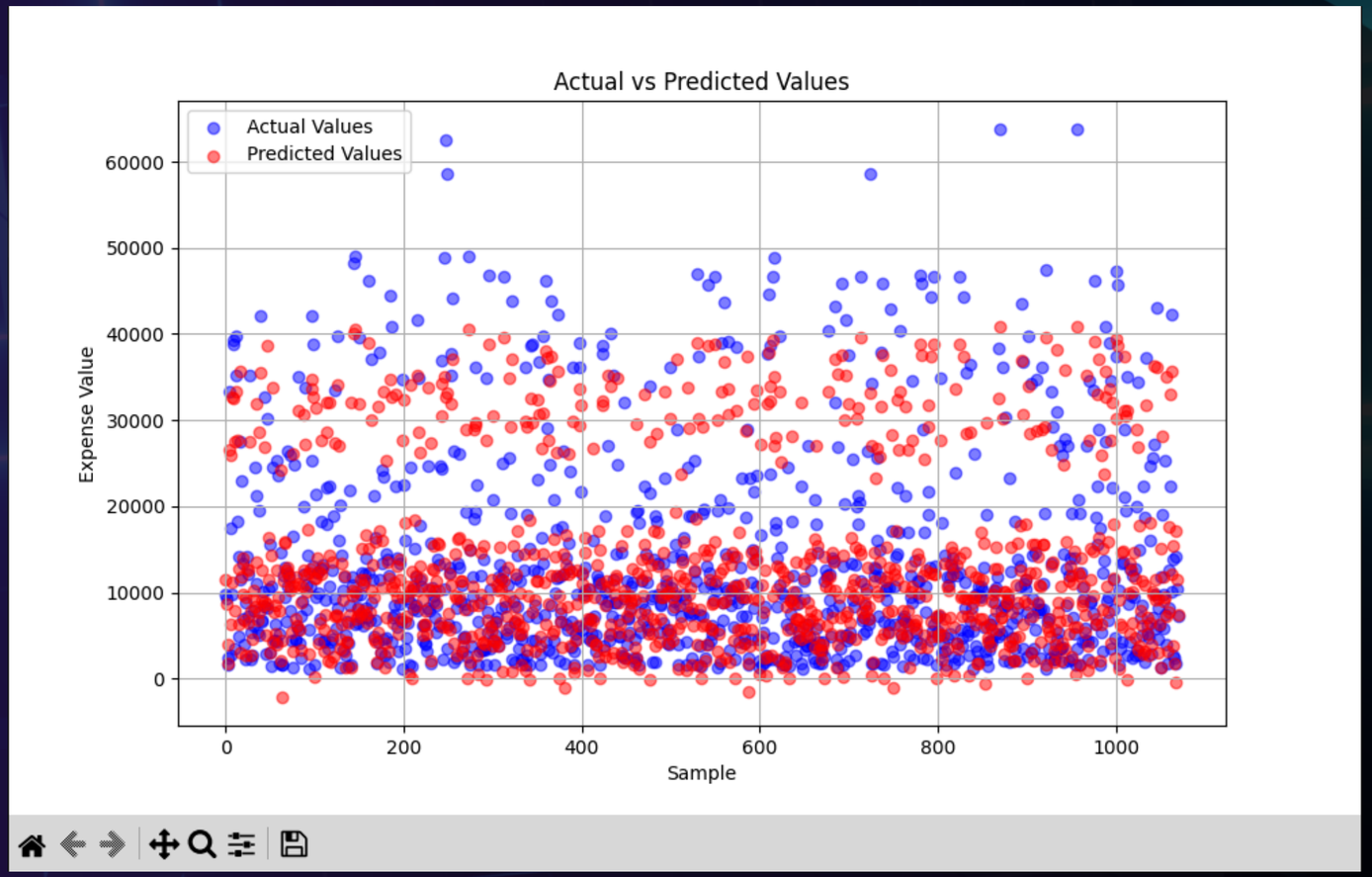
4. R^2 Score (Coefficient of Determination): This evaluates how well the model explains the variance in the target variable. An R^2 score of 1 means the model perfectly predicts the data, while 0 means the model predicts no better than the average of the data. It helps assess the goodness of fit.

Together, these metrics help assess model performance from different perspectives—error magnitude, sensitivity to large errors, and overall fit.

Evaluating Models

Linear Regression Model

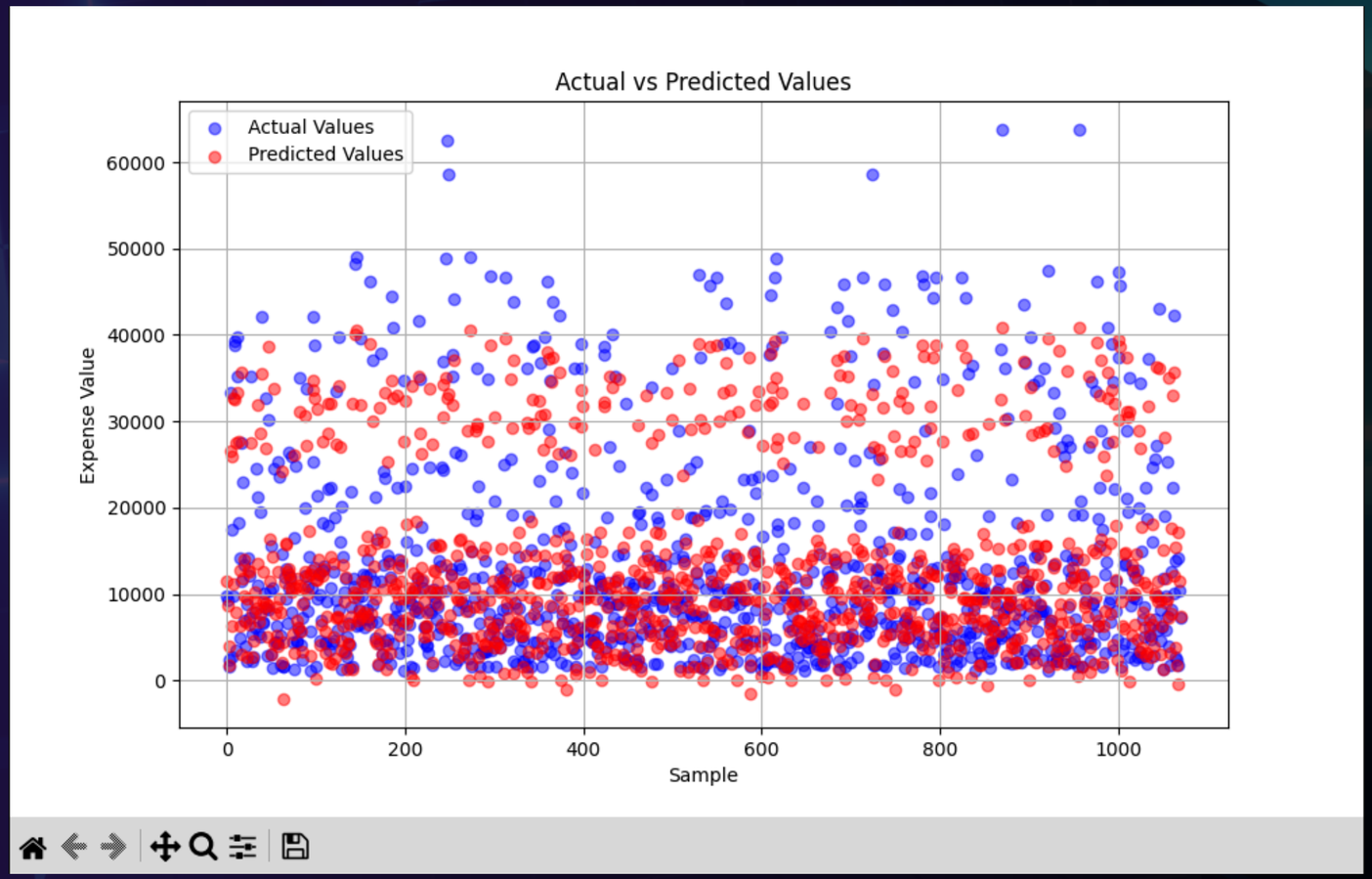
```
Mean absolute error: 0.34027074
Mean square error: 0.24597460
Root mean squared error: 0.49595827
R2 score: 0.74207573
```



Evaluating Models

KNN Model

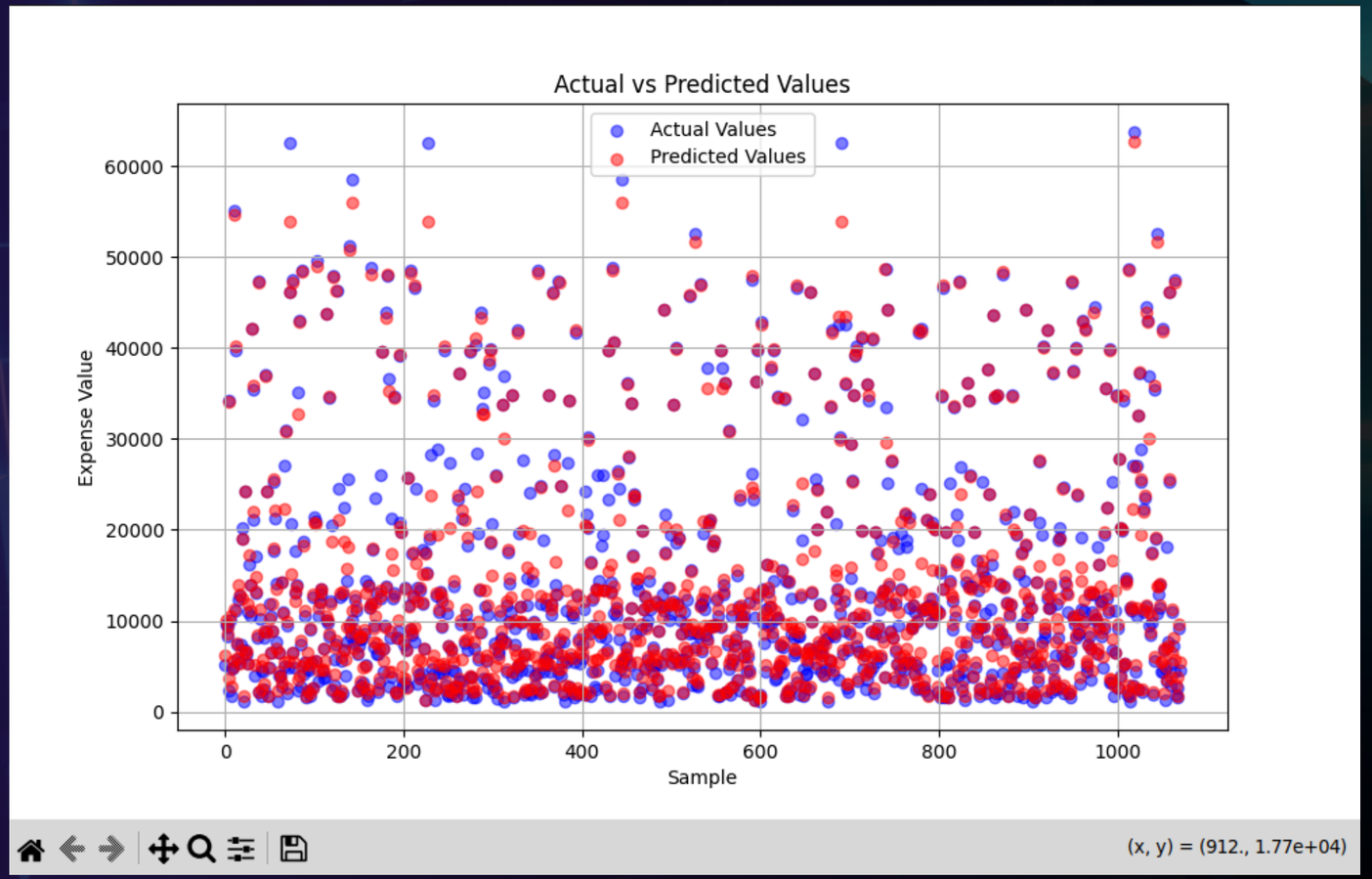
```
Mean absolute error: 0.34027074
Mean square error: 0.24597460
Root mean squared error: 0.49595827
R2 score: 0.74207573
```



Evaluating Models

Random Forest model

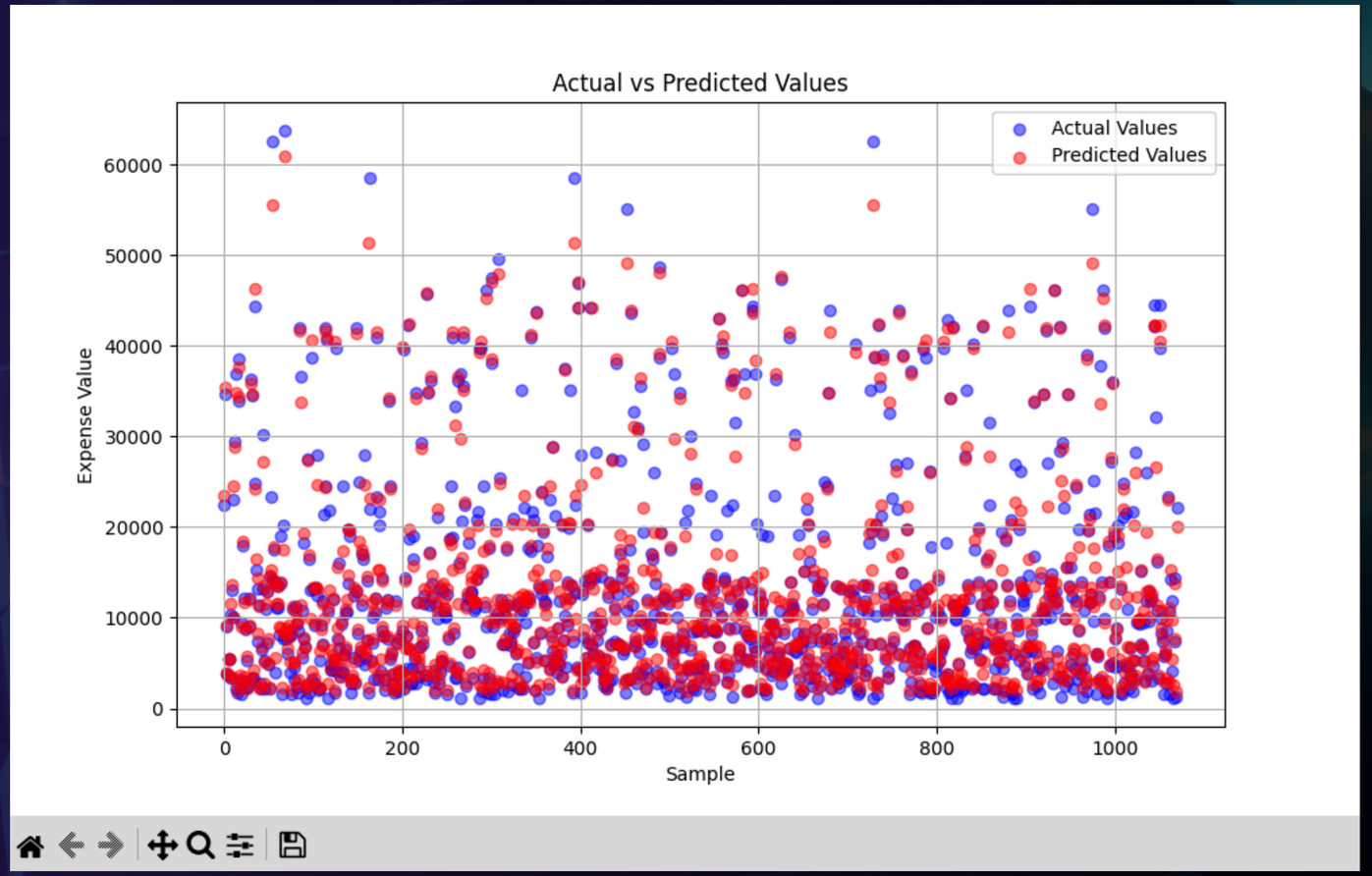
```
Mean absolute error: 0.08125287
Mean square error: 0.03204107
Root mean squared error: 0.17900021
R2 score: 0.97094148
```



Evaluating Models

XGBoost Model

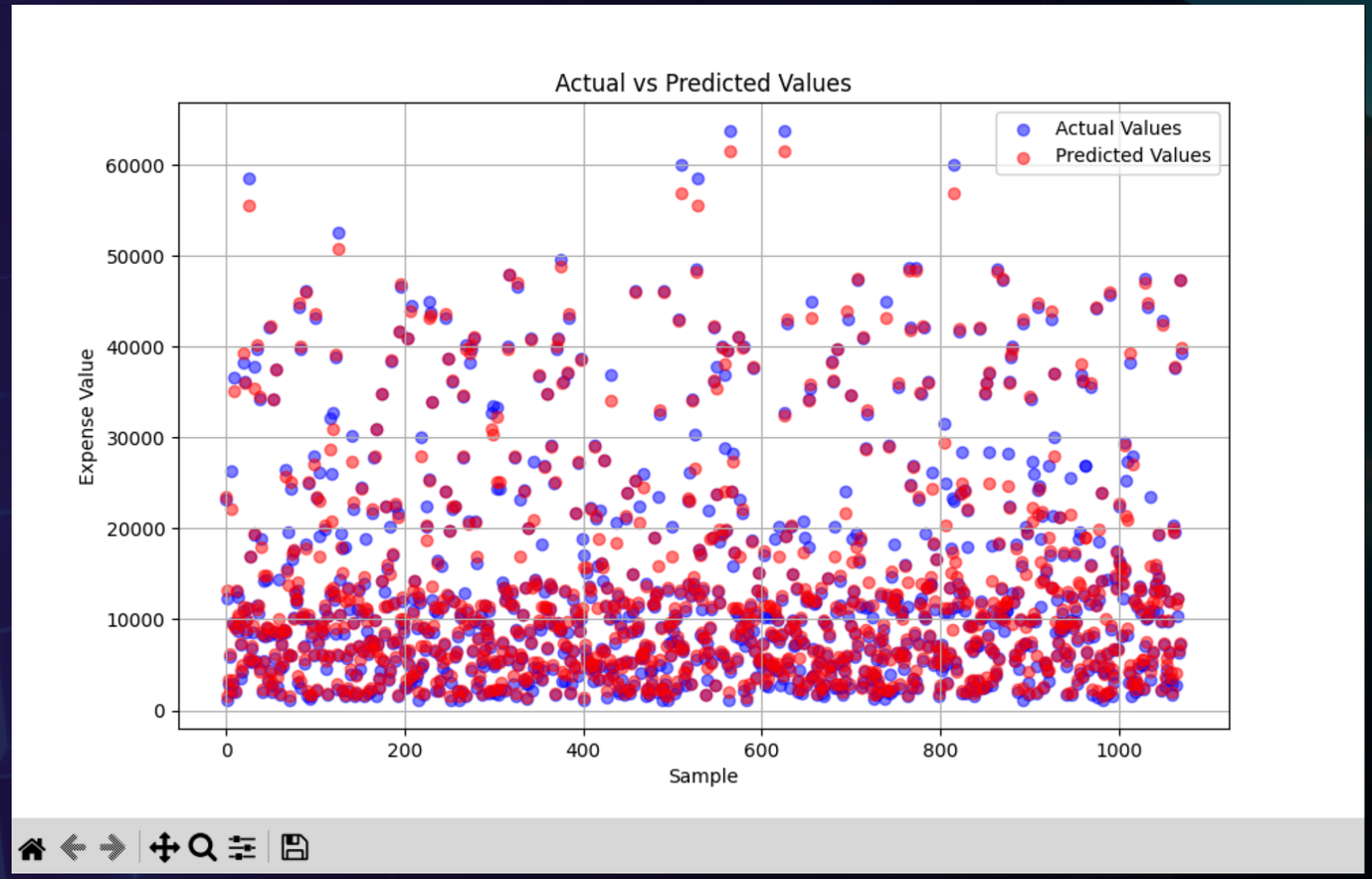
```
Mean absolute error: 0.08996176
Mean square error: 0.03017756
Root mean squared error: 0.17371690
R2 score: 0.96665006
```



Evaluating Models

Voting Regressor Model

```
Mean absolute error: 0.06092070
Mean square error: 0.01507375
Root mean squared error: 0.12277520
R2 score: 0.98457666
```



Comparing all the Models

Model	Data Type	MAE	MSE	RMSE	R^2 Score
KNN	Testing	0.01648269	0.01539224	0.12406548	**0.98606959**
Voting Regressor	Testing	0.06515148	0.02383696	0.15439223	0.97842688
XGBoost	Testing	0.09386532	0.03635291	0.19066440	0.96709959
Random Forest	Testing	0.08913031	0.03820744	0.19546722	0.96542120
Linear Regression	Testing	0.36298774	0.26932241	0.51896282	0.75625567

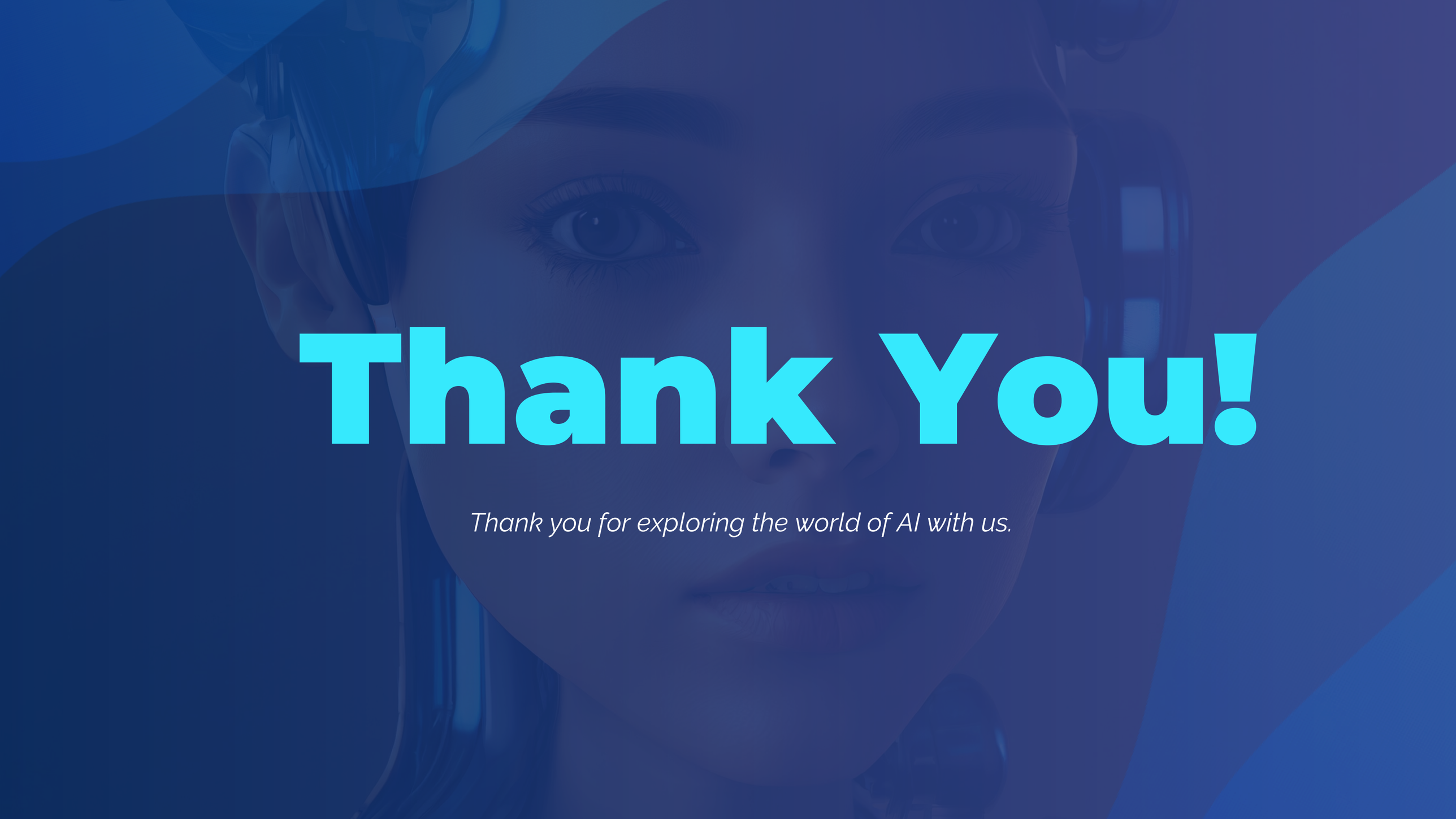
Choosing Best Model

KNN is the best performer because it achieves the highest R^2 score and the lowest error rates on testing data, indicating it generalizes well to unseen data. This means KNN provides the most accurate and consistent predictions in this context.

Testing Best Model

	Actual	Predicted
0	5693.43	5693.430
1	14410.93	14410.930
2	25992.82	25992.820
3	4243.59	4243.590
4	8988.16	8988.160
5	39836.52	39836.520
6	3756.62	3756.620
7	44423.80	44423.800
8	32108.66	32108.660
9	4185.10	4185.100
10	29186.48	29186.480
11	27375.90	27375.900
12	8059.68	7670.595
13	4347.02	4347.020
14	4189.11	4189.110
15	16085.13	16085.130
16	5910.94	5910.940
17	3353.28	3353.280
18	7196.87	8379.950
19	4005.42	4005.420

As seen, the KNN model is able to perform very well on unseen data, as out of 20 test cases it has given a different prediction for only 2 test cases, indicating that the model is able to perform well on unseen data.



Thank You!

Thank you for exploring the world of AI with us.