MODEL OPTIMIZATION AND TUNING PHASE

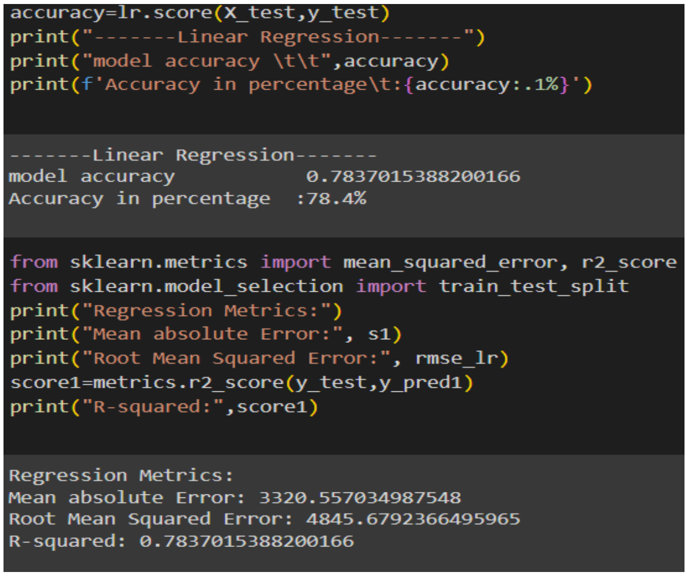
Date	6 July 2024
Team ID	740102
Project Title	Medical Cost Prediction
Maximum Marks	10 Marks

Model Optimization and Tuning Phase

The Model Optimization and Tuning Phase involves refining machine learning models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency. The model optimization and tuning phase for medical cost prediction involves refining predictive models by fine-tuning their parameters through techniques like Grid Search or Random Search. This phase aims to enhance model performance by selecting optimal hyper parameters and evaluating them rigorously using cross-validation methods. By systematically optimizing models and selecting the best-performing configurations, this process ensures accurate predictions of medical costs, crucial for effective healthcare resource allocation and planning.

Performance Metrics Comparison Report (2 Marks):

<table>
<tr><th>Model</th><th>Optimized Metric</th></tr>
<tr><td>Linear Regression</td><td>



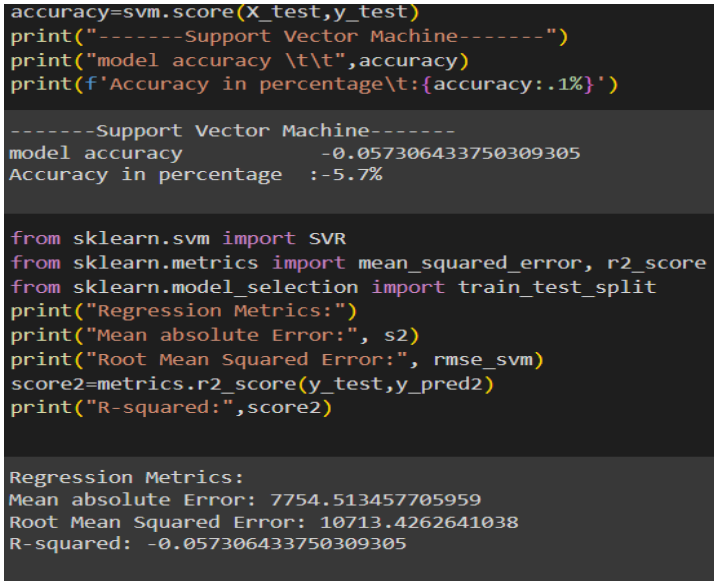
```
accuracy=lr.score(X_test,y_test)
print("-------Linear Regression-------")
print("model accuracy \t\t",accuracy)
print(f'Accuracy in percentage\t:{accuracy:.1%}')
```

```
-------Linear Regression-------
model accuracy          0.7837015388200166
Accuracy in percentage  :78.4%
```

```
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split
print("Regression Metrics:")
print("Mean absolute Error:", s1)
print("Root Mean Squared Error:", rmse_lr)
score1=metrics.r2_score(y_test,y_pred1)
print("R-squared:",score1)
```

```
Regression Metrics:
Mean absolute Error: 3320.557034987548
Root Mean Squared Error: 4845.6792366495965
R-squared: 0.7837015388200166
```

</td></tr>
<tr><td>Support Vector Machine
Regressor</td><td>



```
accuracy=svm.score(X_test,y_test)
print("-------Support Vector Machine-------")
print("model accuracy \t\t",accuracy)
print(f'Accuracy in percentage\t:{accuracy:.1%}')
```

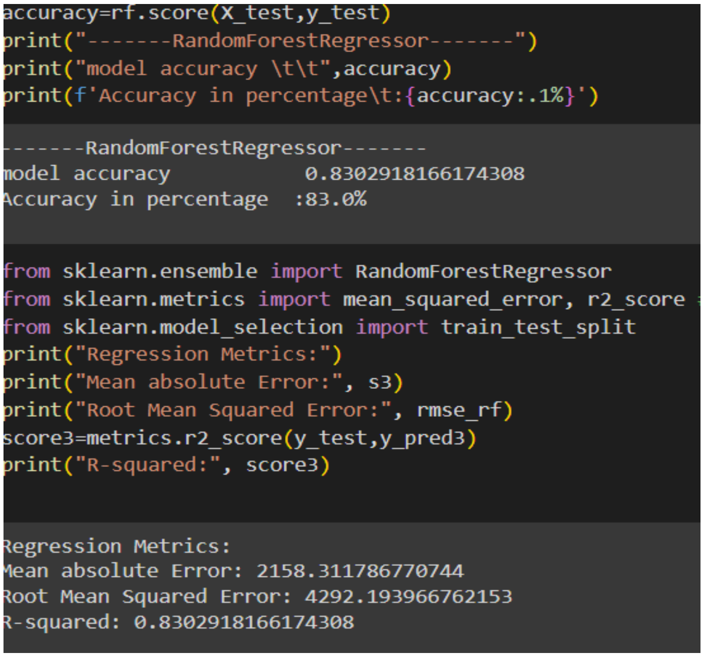
```
-------Support Vector Machine-------
model accuracy          -0.0573064337503093 05
Accuracy in percentage  :-5.7%
```

```
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split
print("Regression Metrics:")
print("Mean absolute Error:", s2)
print("Root Mean Squared Error:", rmse_svm)
score2=metrics.r2_score(y_test,y_pred2)
print("R-squared:",score2)
```

```
Regression Metrics:
Mean absolute Error: 7754.513457705959
Root Mean Squared Error: 10713.4262641038
R-squared: -0.0573064337503093 05
```

</td></tr>
</table>

Random Forest Regressor



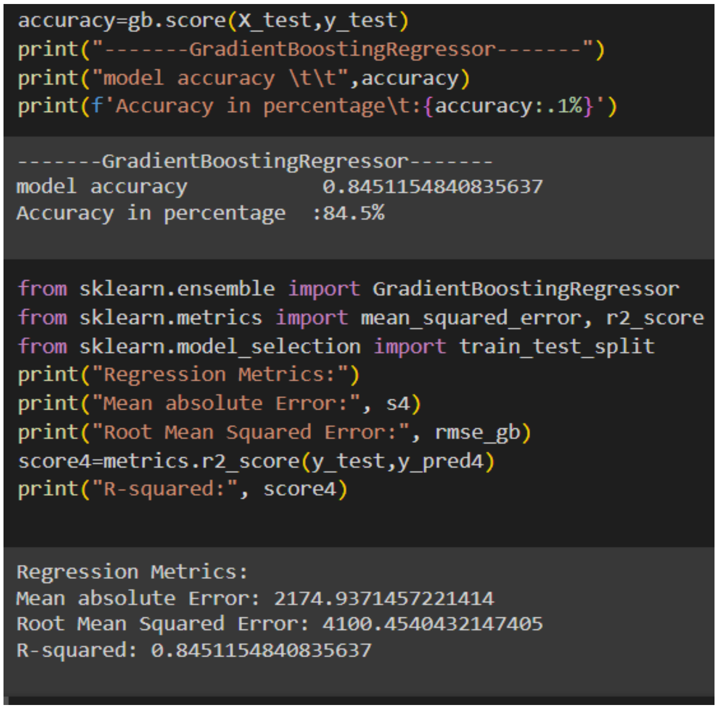
```
accuracy=rf.score(X_test,y_test)
print("-------RandomForestRegressor-------")
print("model accuracy \t\t",accuracy)
print(f'Accuracy in percentage\t:{accuracy:.1%}')
```

```
-------RandomForestRegressor-------
model accuracy           0.8302918166174308
Accuracy in percentage  :83.0%
```

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split
print("Regression Metrics:")
print("Mean absolute Error:", s3)
print("Root Mean Squared Error:", rmse_rf)
score3=metrics.r2_score(y_test,y_pred3)
print("R-squared:", score3)
```

```
Regression Metrics:
Mean absolute Error: 2158.311786770744
Root Mean Squared Error: 4292.193966762153
R-squared: 0.8302918166174308
```

Gradient Boosting Regressor



```
accuracy=gb.score(X_test,y_test)
print("-------GradientBoostingRegressor-------")
print("model accuracy \t\t",accuracy)
print(f'Accuracy in percentage\t:{accuracy:.1%}')
```

```
-------GradientBoostingRegressor-------
model accuracy           0.8451154840835637
Accuracy in percentage  :84.5%
```

```
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split
print("Regression Metrics:")
print("Mean absolute Error:", s4)
print("Root Mean Squared Error:", rmse_gb)
score4=metrics.r2_score(y_test,y_pred4)
print("R-squared:", score4)
```

```
Regression Metrics:
Mean absolute Error: 2174.9371457221414
Root Mean Squared Error: 4100.4540432147405
R-squared: 0.8451154840835637
```

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Random Forest Regressor	Random Forest Regressor is ideal for predicting cost due to its high accuracy and ability to capture complex, non-linear relationships in data. It provides feature importance insights, helping to understand key cost drivers. The method is robust to outliers and noise, common in medical data. Random Forests reduce overfitting by averaging multiple decision trees and require minimal preprocessing. Additionally, they support both continuous and categorical variables and allow parallel processing, enhancing efficiency in large datasets.