Model Development Phase Template

Date	7 July 2024
Team ID	team-740657
Project Title	Medical Cost Prediction
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshot

Initial model training code

Linear Regression

```
[ ] from sklearn.linear_model import LinearRegression
```

```
lr= LinearRegression()
lr.fit(X_train,y_train)
```

```
LinearRegression
LinearRegression()
```

```
[ ] y_pred1=lr.predict(X_test)
```

```
from sklearn import  metrics
```

```
[ ] score1=metrics.r2_score(y_test,y_pred1)
    print(score1)
```

```
0.7837015388200166
```

```
[ ] s1=metrics.mean_absolute_error( y_test,y_pred1)
    print(s1)
```

```
3320.557034987548
```

```
[ ] rmse_lr=np.sqrt(metrics.mean_squared_error(y_test,y_pred1))
```

```
rmse_lr=np.sqrt(metrics.mean_squared_error(y_test,y_pred1))
print("mean_squared_error",rmse_lr)
```

```
mean_squared_error 4845.6792366495965
```

```
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%
```

Support Vector MachineRegressor

```
[ ] from sklearn.svm import SVR
```

```
[ ] svm= SVR()
    svm.fit(X_train,y_train)
```

```
SVR
SVR()
```

```
[ ] y_pred2=svm.predict(X_test)
```

```
score2=metrics.r2_score(y_test,y_pred2)
print(score2)
```

```
-0.057306433750309305
```

```
[ ] s2=metrics.mean_absolute_error( y_test,y_pred2)
    print(s2)
```

```
7754.513457705959
```

```
[ ] rmse_svm=np.sqrt(metrics.mean_squared_error(y_test,y_pred2))
    print("root_mean_squared_error",rmse_svm)
```

```
root_mean_squared_error 10713.4262641038
```

```
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          -0.057306433750309305
Accuracy in percentage  :-5.7%
```

RandomForest Regressor

```
[ ] from sklearn.ensemble import RandomForestRegressor
```

```
rf= RandomForestRegressor()
rf.fit(X_train,y_train)
```

```
RandomForestRegressor
RandomForestRegressor()
```

```
[ ] y_pred3=rf.predict(X_test)
```

```
[ ] score3=metrics.r2_score(y_test,y_pred3)
    print(score3)
```

```
0.8302918166174308
```

```
[ ] s3=metrics.mean_absolute_error( y_test,y_pred3)
    print(s3)
```

```
2158.311786770744
```

```
[ ] rmse_rf=np.sqrt(metrics.mean_squared_error(y_test,y_pred3))
    print("root_mean_squared_error",rmse_rf)
```

```
root_mean_squared_error 4292.193966762153
```

```
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%
```

GradientBoostingRegressor

```
from sklearn.ensemble import GradientBoostingRegressor
```

```
[ ] gb= GradientBoostingRegressor()
    gb.fit(X_train,y_train)
```

```
GradientBoostingRegressor
GradientBoostingRegressor()
```

```
[ ] y_pred4=gb.predict(X_test)
```

```
[ ] score4=metrics.r2_score(y_test,y_pred4)
    print(score4)
```

```
0.8451154840835637
```

```
[ ] s4=metrics.mean_absolute_error( y_test,y_pred4)
    print(s4)
```

```
2174.9371457221414
```

```
[ ] rmse_gb=np.sqrt(metrics.mean_squared_error(y_test,y_pred4))
    print("root_mean_squared_error",rmse_gb)
```

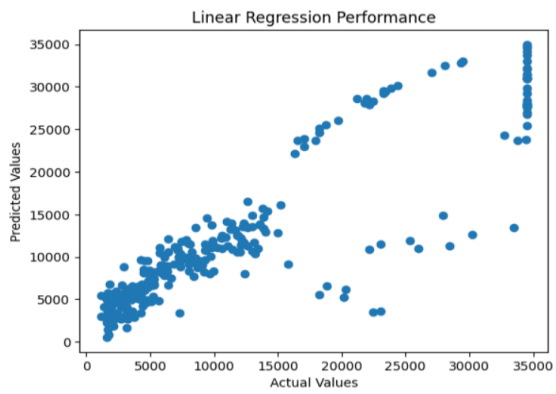
```
root_mean_squared_error 4100.4540432147405
```

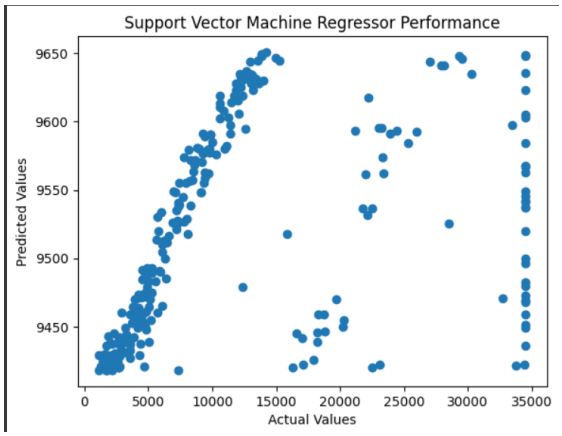
```
accuracy=gb.score(X_test,y_test)
print("-------GradientBoostingRegressor-------")
print("model accuracy \t\t",accuracy)
print(f'Accuracy in percentage\t:{accuracy:.1%}')
```

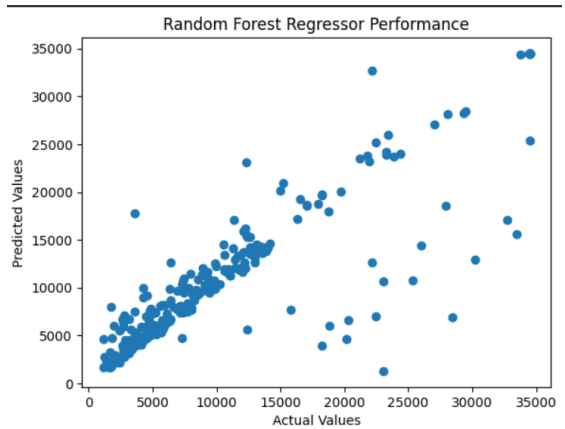
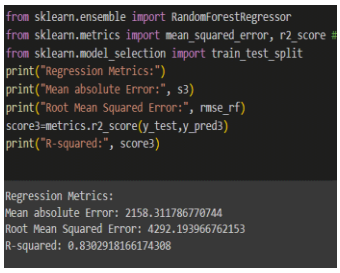
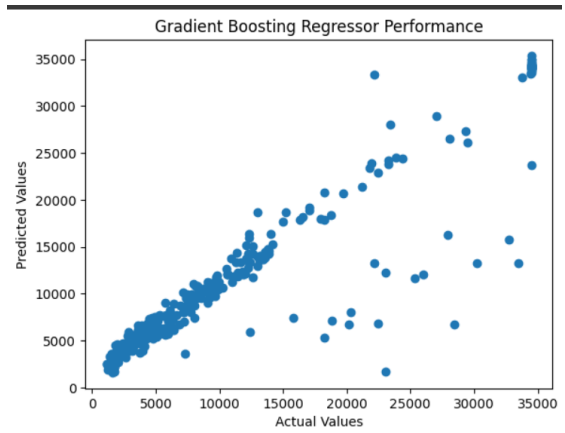
```
-------GradientBoostingRegressor-------
model accuracy          0.8451154840835637
Accuracy in percentage  :84.5%
```

Model Validation and Evaluation Report:

Model	Performance Report	Accuracy	Regression matrix

Linear Regression		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

Support Vector Machine Regressor		-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: -0.057306433750309305

Random Forest Regressor		83%	
Gradient Boosting Regresssor		84.5%