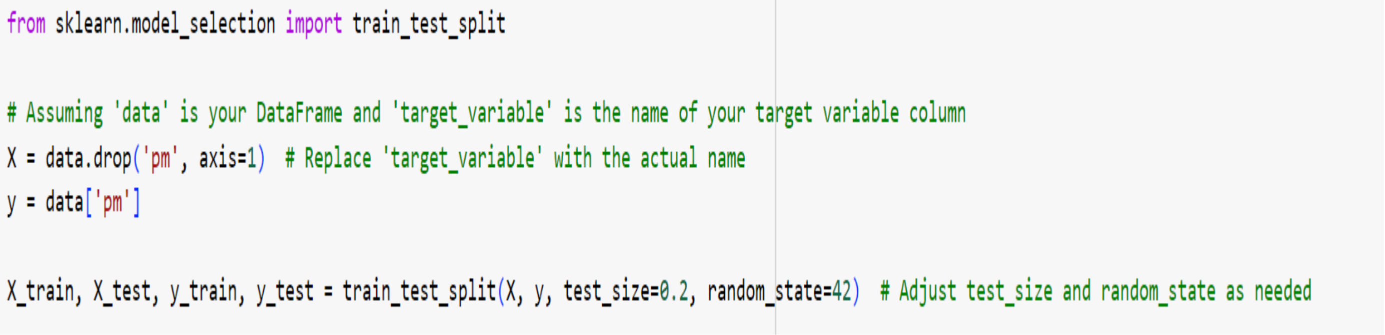
Model Development Phase Template

Date	20 June 2024
Team ID	739809
Project Title	Predicting Permanent Magnet Resistance Of Electronic Motor Using Machine Learning
Maximum Marks	4 Marks

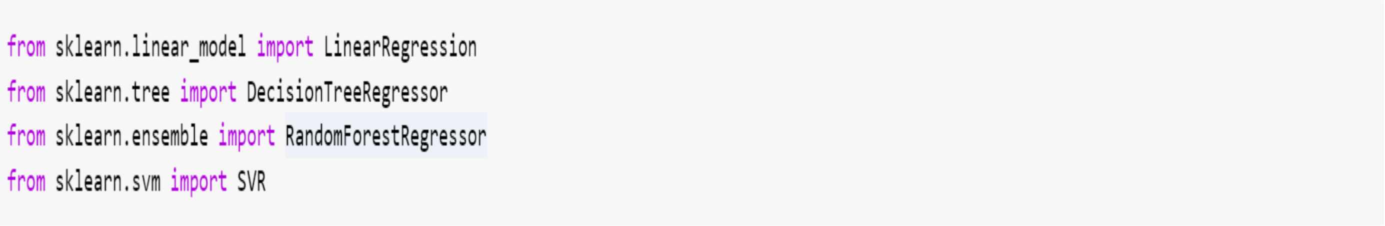
Initial Model Training Code, Model Validation and Evaluation Report

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 screenshots.

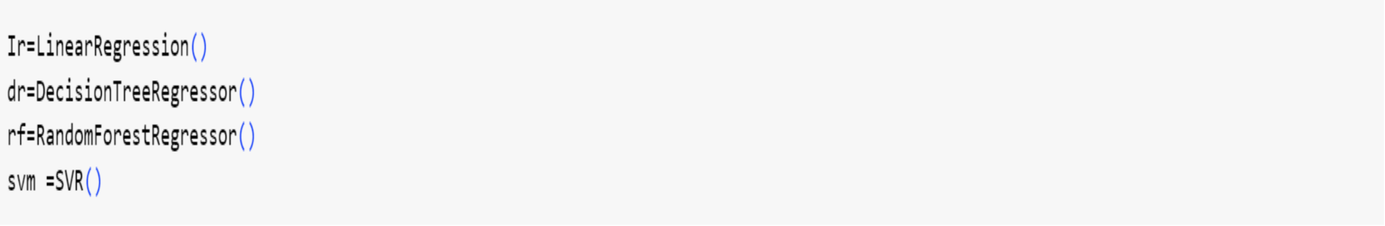
Initial Model Training Code:



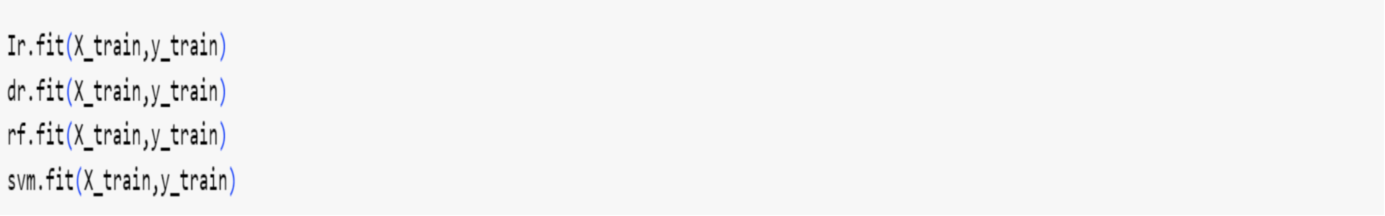
```
from sklearn.model_selection import train_test_split

# Assuming 'data' is your DataFrame and 'target_variable' is the name of your target variable column
X = data.drop('pm', axis=1)  # Replace 'target_variable' with the actual name
y = data['pm']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)  # Adjust test_size and random_state as needed
```



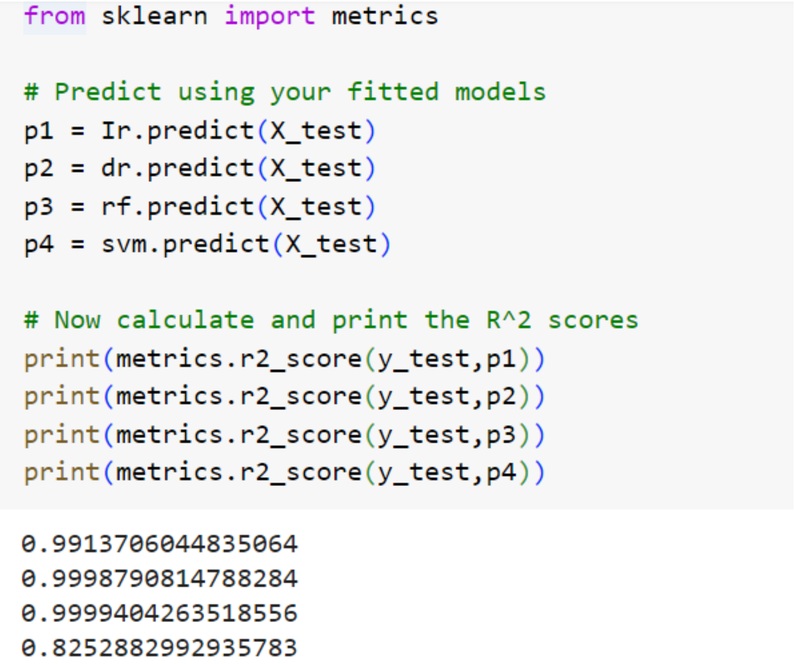
```
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
```



```
Ir=LinearRegression()
dr=DecisionTreeRegressor()
rf=RandomForestRegressor()
svm =SVR()
```

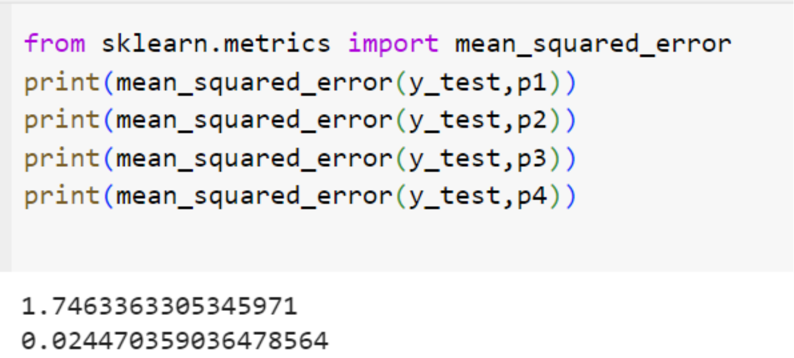


```
Ir.fit(X_train,y_train)
dr.fit(X_train,y_train)
rf.fit(X_train,y_train)
svm.fit(X_train,y_train)
```



```
from sklearn import metrics

# Predict using your fitted models
p1 = Ir.predict(X_test)
p2 = dr.predict(X_test)
p3 = rf.predict(X_test)
p4 = svm.predict(X_test)

# Now calculate and print the R^2 scores
print(metrics.r2_score(y_test,p1))
print(metrics.r2_score(y_test,p2))
print(metrics.r2_score(y_test,p3))
print(metrics.r2_score(y_test,p4))
```

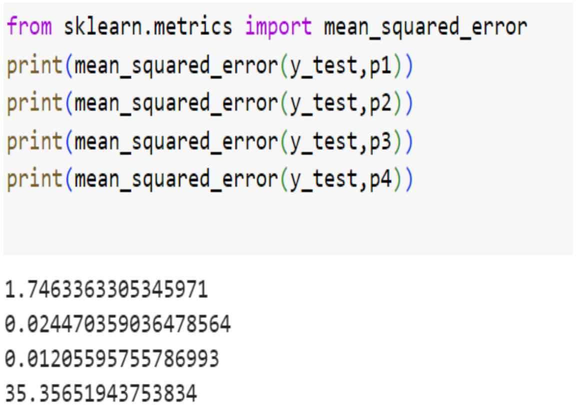
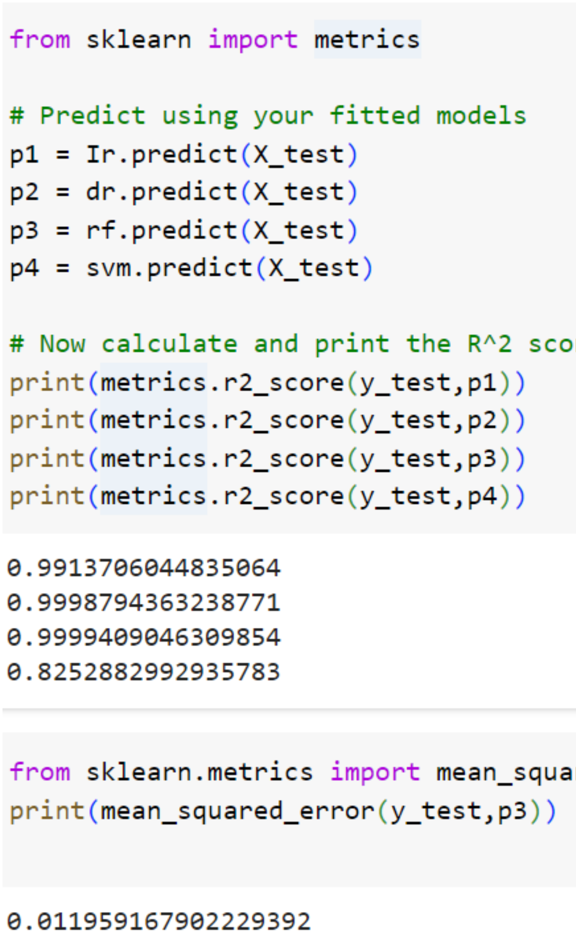
```
0.9913706044835064
0.9998790814788284
0.9999404263518556
0.8252882992935783
```



```
from sklearn.metrics import mean_squared_error
print(mean_squared_error(y_test,p1))
print(mean_squared_error(y_test,p2))
print(mean_squared_error(y_test,p3))
print(mean_squared_error(y_test,p4))
```

```
1.7463363305345971
0.024470359036478564
```

Model	Mean Squared Error	R2-Score

Decision tree,linear regression,randomforest regression	
	`from sklearn.metrics import mean_squared_error`
	`print(mean_squared_error(y_test,p1))`
	`print(mean_squared_error(y_test,p2))`
	`print(mean_squared_error(y_test,p3))`
	`print(mean_squared_error(y_test,p4))`
	
	`1.7463363305345971`
	`0.024470359036478564`
	`0.01205595755786993`
	`35.35651943753834`
		
		`from sklearn import metrics`
		
		`# Predict using your fitted models`
		`p1 = Ir.predict(X_test)`
		`p2 = dr.predict(X_test)`
		`p3 = rf.predict(X_test)`
		`p4 = svm.predict(X_test)`
		
		`# Now calculate and print the R^2 sco`
		`print(metrics.r2_score(y_test,p1))`
		`print(metrics.r2_score(y_test,p2))`
		`print(metrics.r2_score(y_test,p3))`
		`print(metrics.r2_score(y_test,p4))`
		
		`0.9913706044835064`
		`0.9998794363238771`
		`0.9999409046309854`
		`0.8252882992935783`
		
		`from sklearn.metrics import mean_squa`
		`print(mean_squared_error(y_test,p3))`
		
		`0.011959167902229392`