Project Development Phase
Model Performance Test

Date	10 November 2022
Team ID	PNT2022TMID31390
Project Name	Project - Predicting the energy output of wind turbine based on weather condition
Maximum Marks	10 Marks

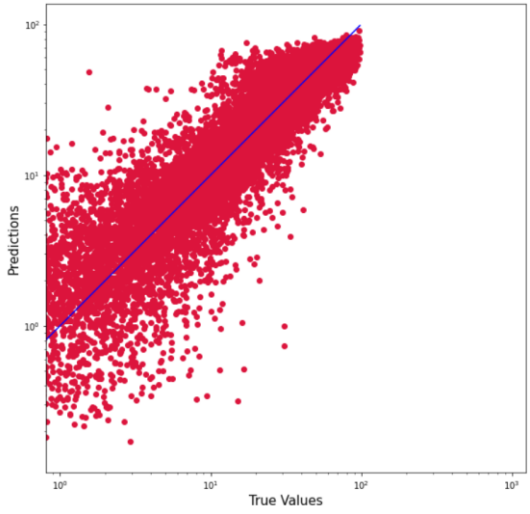
Model Performance Testing:

Random Forest Regressor:

S.No.	Parameter	Values	Screenshot

1.	Model Summary	Gives the highest accuracy. Random forest is a supervised learning algorithm that uses an ensemble learning method for classification and regression.	`from sklearn.ensemble import RandomForestRegressor`
			`regressor = RandomForestRegressor (n_estimators=30, random_state=0)`
			`regressor.fit(X_train,y_train)`
			
			`RandomForestRegressor(n_estimators=30, random_state=0)`

2.	Accuracy	Training Accuracy – 100%
		
		Validation Accuracy – 88.4%
			`from sklearn.metrics import r2_score`
			
			`r2_score(y_test, y_pred)`
			
			`0.8849999426996336`
			Ref:Model1_RandomForestRegressor_VIDHYAMBIKA.ipynb

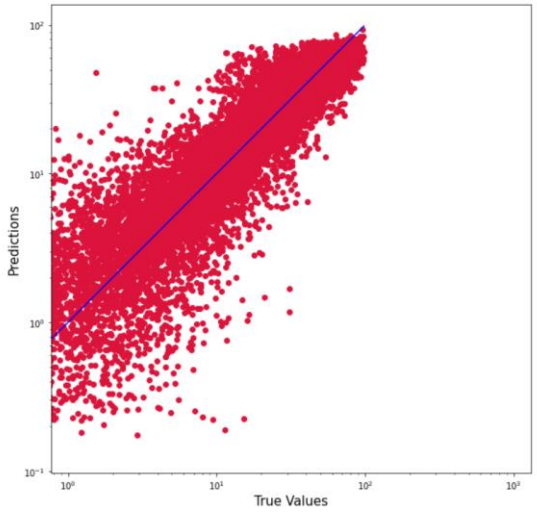
3.	Visualizing the model	X axis-True values Y axis-Predicted values	 Ref: VISUALISING_M1ipynb.ipynb

Random Forest Regressor but with two parameters eliminated:

S.No.	Parameter	Values	Screenshot

1.	Model Summary	Gives accuracy less than model 1. Random forest is a bagging technique and not a boosting technique.	`from sklearn.ensemble import RandomForestRegressor`
			`regressor = RandomForestRegressor (n_estimators=20, random_state=0)`
			`regressor.fit(X_train,y_train)`
			`RandomForestRegressor(n_estimators=20, random_state=0)`

2.	Accuracy	Training Accuracy – 100%
		
		Validation Accuracy – 87.7%
			`from sklearn.metrics import r2_score`
			`r2_score(y_test, y_pred)`
			`0.8773928979826746`
			Ref:Model2RandomForestRegeliminating2par.ipynb

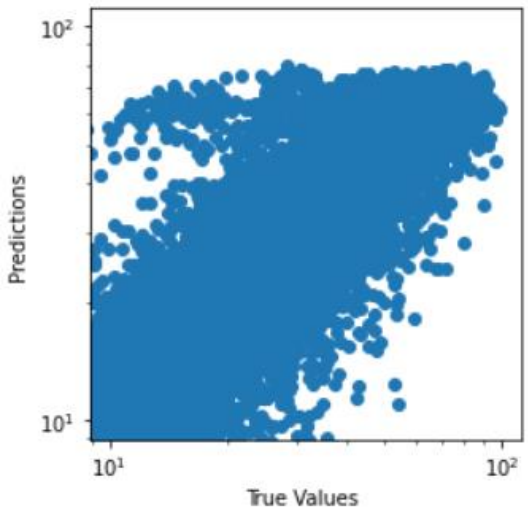
3.	Visualizing the model	X axis-True values Y axis-Predicted values	 Ref: VISUALISINGM2.ipynb

Support Vector Regression (SVR):

S.No.	Parameter	Values	Screenshot

1.	Model Summary	Gives accuracy less than model 2. SVR uses the same principle as SVM, but this SVR is applied for regression problems.	`from sklearn.svm import SVR`
			
			`# Choose regression method and set hyperparameter`
			`svr_rbf=SVR(C=1.0, epsilon=0.2, kernel='rbf')`
			
			`# Training of the regression model`
			`svr_rbf.fit(X_train, y_train)`
			`y_pred = svr_rbf.predict(X_test)`

2.	Accuracy	Training Accuracy – 100%
		
		Validation Accuracy – 84.6%
			`from sklearn.metrics import r2_score`
			
			`r2_score(y_test, y_pred)`
			
			`0.8466893820599397`
			
			Ref: Model3SVR.ipynb

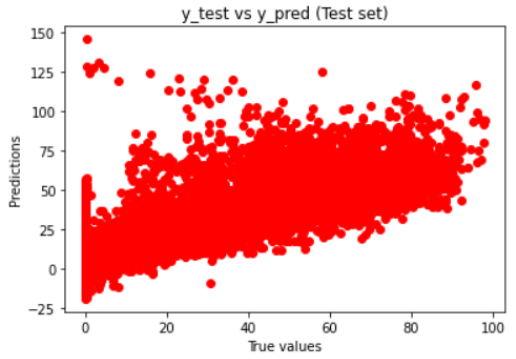
3.	Visualizing the model	X axis-True values Y axis-Predicted values	 Ref: VISUALISING_M3.ipynb

Multi linear Regression:

S.No.	Parameter	Values	Screenshot

1.	Model Summary	Gives accuracy less than model 3. Multiple linear regression (MLR), also known simply as multiple regression, is a statistical technique that uses several explanatory variables to predict the outcome of a response variable. Multiple regression is an extension of linear (OLS) regression that uses just one explanatory variable.	`from sklearn.linear_model import LinearRegression`
			`lr=LinearRegression()`
			
			`lr.fit(X_train,y_train)`
			
			`LinearRegression()`
			
			`y_pred=lr.predict(X_test)`
			`y_pred`
			
			Ref: Model4multilinearregression.ipynb

2.	Accuracy	Training Accuracy – 100%
		
		Validation Accuracy – 75%
			`print("Mean squared error =", round(sm.mean_squared_error(y_test, y_pred), 2))`
			`print("Median absolute error =", round(sm.median_absolute_error(y_test, y_pred), 2))`
			`print("Explain variance score =", round(sm.explained_variance_score(y_test, y_pred), 2))`
			`print("R2 score =", round(sm.r2_score(y_test, y_pred), 2))`
			
			`Mean squared error = 149.42`
			`Median absolute error = 6.21`
			`Explain variance score = 0.75`
			`R2 score = 0.75`

3.	Visualizing the model	X axis-True values Y axis-Predicted values	 Ref: VISUALISING_M4.ipynb