DWDM Submission

Roll Number:	18BCD7143
Student name:	L. Satyajit
Slot (L1/L2/L4):	L2
Title of the Project:	Prediction of Global Sales in Video game sales
Objective of the Project (What exactly the project is about?)	Prediction of Global Sales(Units) in Video game sales using attributes like genre, critic ratings, critic scores, user score and user ratings.

Dataset Link	**Number of rows and Columns**	**About columns**
https://www.kaggle.com/rush4ratio/video-game-sales-with-ratings	**Rows : 16719** **Columns : 16**	**Number of Categorical columns: 5** **Number of Integer/Float Columns: 10**
		Unique Values in each Column: Name : 11563 Platform : 31 Year of game : 40 Genre : 12 Publisher : 582

Challenges identified in the project	**How did you address that challenge?**	**References**
Missing values	Replacing with mean (basic)	**https://www.kaggle.com/arthurtok/the-console-wars-ps-vs-xbox-vs-wii**
	Dropping rows with NA values	**https://www.kaggle.com/arthurtok/the-console-wars-ps-vs-xbox-vs-wii**
	Replacing with median	**https://www.kaggle.com/arthurtok/the-console-wars-ps-vs-xbox-vs-wii**
	Replacing with mode	**https://www.kaggle.com/arthurtok/the-console-wars-ps-vs-xbox-vs-wii**
Noisy data	Normalization	**https://www.kaggle.com/serigne/stacked-regressions-top-4-on-leaderboard**
	Standardization	**https://www.kaggle.com/serigne/stacked-regressions-top-4-on-leaderboard**
Abnormal distribution	Log transformation	**https://www.kaggle.com/serigne/stacked-regressions-top-4-on-leaderboard**

Outliers	Dropping rows with outliers	https://www.kaggle.com/serigne/stacked-regressions-top-4-on-leaderboard
Handling Publisher, Genre and Rating columns	One-Hot encoding	https://www.geeksforgeeks.org/ml-one-hot-encoding-of-datasets-in-python/

Without Pre-processing- Different Algorithms	**Performance(Accuracy/R-Squared and RMSE)**	**Which model worked well on the test data and WHY?**
Linear Regression	R2 : 0.3671 RMSE : 0.3770	Random Forest Regressor worked fairly well with the test data as it showed the highest R Squared value of all the others. And showed the least RMSE value as well.
Decision Tree Regression	R2 : 0.2382 RMSE : 0.4136	
Ridge	R2 : 0.3672 RMSE : 0.3769	
SVR	R2 : 0.1563 RMSE : 0.4353	
K Neighbours	R2 : 0.2542 RMSE : 0.4092	
ADA Boosting	R2 : 0.2271 RMSE : 0.4166	
Gradient boosting regressor	R2 : 0.5486 RMSE : 0.3184	
Random Forest Regressor	R2 : 0.5645 RMSE : 0.3127	

Which Pre-processing technique you applied?	**Why you applied that pre-processing Technique?**	**References**
Replacing rows with NA values with mean (basic)	To handle missing values	https://machinelearningmastery.com/handle-missing-data-python/
Dropping rows with NA values	To handle missing values	https://machinelearningmastery.com/handle-missing-data-python/
Replacing rows with NA values with median	To handle missing values	https://machinelearningmastery.com/handle-missing-data-python/
Replacing rows with NA values with mode	To handle missing values	https://machinelearningmastery.com/handle-missing-data-python/
Min-Max Normalization	To handle noisy data	https://www.analyticsvidhya.com/blog/2020/04/featur

		e-scaling-machine-learning-normalization-standardization/
Standardization	To handle noisy data	https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/
Log transformation (basic)	To handle abnormal distribution	https://www.kaggle.com/jruots/forecasting-video-game-sales/notebook
Dropping rows with outliers	To handle outliers	https://www.kaggle.com/jruots/forecasting-video-game-sales/notebook

Pre-processing technique name?	**Data Mining Algorithms you applied?**	**Performance(Accuracy/other confusion matrix measures)** **(Before pre-processing)**	**(After pre-processing)**	**Which model worked well on the test data and WHY?**
Replacing rows with NA values with mean (basic)	Linear Regression	R2 : 0.3671 RMSE : 0.3770	R2 : 0.3671 RMSE : 0.3770	Random Forest Regressor showed the best prediction accuracy out of all algorithms used, the most accuracy being the result of **removal of outliers** with the highest R Squared value and lowest Root Mean Squared error.
	Decision Tree Regression	R2 : 0.2382 RMSE : 0.4136	R2 : 0.2382 RMSE : 0.4136	
	Ridge	R2 : 0.3672 RMSE : 0.3769	R2 : 0.3672 RMSE : 0.3769	
	SVR	R2 : 0.1563 RMSE : 0.4353	R2 : 0.1563 RMSE : 0.4353	
	K Neighbours	R2 : 0.2542 RMSE : 0.4092	R2 : 0.2542 RMSE : 0.4092	
	ADA Boosting	R2 : 0.2271 RMSE : 0.4166	R2 : 0.2271 RMSE : 0.4166	
	Gradient boosting regressor	R2 : 0.5486 RMSE : 0.3184	R2 : 0.5486 RMSE : 0.3184	
	Random Forest Regressor	**R2 : 0.5645 RMSE : 0.3127**	**R2 : 0.5645 RMSE : 0.3127**	
Dropping rows with NA values	Linear Regression	R2 : 0.3671 RMSE : 0.3770	R2 : 0.4397 RMSE : 0.3919	
	Decision Tree Regression	R2 : 0.2382 RMSE : 0.4136	R2 : 0.3163 RMSE : 0.4329	

	Ridge	R2 : 0.3672 RMSE : 0.3769	R2 : 0.4396 RMSE : 0.3919	
	SVR	R2 : 0.1563 RMSE : 0.4353	R2 : 0.1724 RMSE : 0.4763	
	K Neighbours	R2 : 0.2542 RMSE : 0.4092	R2 : 0.2154 RMSE : 0.4638	
	ADA Boosting	R2 : 0.2271 RMSE : 0.4166	R2 : 0.3436 RMSE : 0.4242	
	Random Forest Regressor	**R2 : 0.5645 RMSE : 0.3127**	**R2 : 0.5900 RMSE : 0.3352**	
Replacing rows with NA values with median	Linear Regression	R2 : 0.3671 RMSE : 0.3770	R2 : 0.3780 RMSE : 0.3737	
	Decision Tree Regression	R2 : 0.2382 RMSE : 0.4136	R2 : 0.2433 RMSE : 0.4122	
	Ridge	R2 : 0.3672 RMSE : 0.3769	R2 : 0.3780 RMSE : 0.3737	
	SVR	R2 : 0.1563 RMSE : 0.4353	R2 : 0.2137 RMSE : 0.4202	
	K Neighbours	R2 : 0.2542 RMSE : 0.4092	R2 : 0.2453 RMSE : 0.4117	
	ADA Boosting	R2 : 0.2271 RMSE : 0.4166	R2 : 0.2865 RMSE : 0.4003	
	Random Forest Regressor	**R2 : 0.5645 RMSE : 0.3127**	**R2 : 0.5768 RMSE : 0.3083**	
Replacing rows with NA values with mode	Linear Regression	R2 : 0.3671 RMSE : 0.3770	R2 : 0.4163 RMSE : 0.3450	
	Decision Tree Regression	R2 : 0.2382 RMSE : 0.4136	R2 : 0.1376 RMSE : 0.4194	
	Ridge	R2 : 0.3672 RMSE : 0.3769	R2 : 0.4166 RMSE : 0.3449	
	SVR	R2 : 0.1563 RMSE : 0.4353	R2 : 0.2672 RMSE : 0.3866	
	K Neighbours	R2 : 0.2542 RMSE : 0.4092	R2 : 0.2401 RMSE : 0.3937	
	ADA Boosting	R2 : 0.2271 RMSE : 0.4166	R2 : -0.4766 RMSE : 0.5488	

	Random Forest Regressor	**R2 : 0.5645** **RMSE : 0.3127**	**R2 : 0.5874** **RMSE : 0.2901**	
Min-Max Normalization	Linear Regression	R2 : 0.3671 RMSE : 0.3770	R2 : 0.3671 RMSE : 0.3770	
	Decision Tree Regression	R2 : 0.2382 RMSE : 0.4136	R2 : 0.1893 RMSE : 0.4267	
	Ridge	R2 : 0.3672 RMSE : 0.3769	R2 : 0.3667 RMSE : 0.3771	
	SVR	R2 : 0.1563 RMSE : 0.4353	R2 : 0.4130 RMSE : 0.3631	
	K Neighbours	R2 : 0.2542 RMSE : 0.4092	R2 : 0.3310 RMSE : 0.3876	
	ADA Boosting	R2 : 0.2271 RMSE : 0.4166	R2 : 0.1849 RMSE : 0.4278	
	Random Forest Regressor	**R2 : 0.5645** **RMSE : 0.3127**	**R2 : 0.5703** **RMSE : 0.3106**	
Standardization	Linear Regression	R2 : 0.3671 RMSE : 0.3770	R2 : 0.3669 RMSE : 0.3770	
	Decision Tree Regression	R2 : 0.2382 RMSE : 0.4136	R2 : 0.2511 RMSE : 0.4101	
	Ridge	R2 : 0.3672 RMSE : 0.3769	R2 : 0.3671 RMSE : 0.3770	
	SVR	R2 : 0.1563 RMSE : 0.4353	R2 : 0.4997 RMSE : 0.3352	
	K Neighbours	R2 : 0.2542 RMSE : 0.4092	R2 : 0.3940 RMSE : 0.3689	
	ADA Boosting	R2 : 0.2271 RMSE : 0.4166	R2 : 0.2867 RMSE : 0.4002	
	Random Forest Regressor	**R2 : 0.5645** **RMSE : 0.3127**	**R2 : 0.5612** **RMSE : 0.3139**	
Dropping rows with outliers	Linear Regression	R2 : 0.3671 RMSE : 0.3770	R2 : 0.4308 RMSE : 0.3565	
	Decision Tree Regression	R2 : 0.2382 RMSE : 0.4136	R2 : 0.3147 RMSE : 0.3912	
	Ridge	R2 : 0.3672 RMSE : 0.3769	R2 : 0.4314 RMSE : 0.3563	
	SVR	R2 : 0.1563 RMSE : 0.4353	R2 : 0.1948 RMSE : 0.4240	

	K Neighbours	R2 : 0.2542 RMSE : 0.4092	R2 : 0.3004 RMSE : 0.3953	
	ADA Boosting	R2 : 0.2271 RMSE : 0.4166	R2 : 0.3401 RMSE : 0.3839	
	Random Forest Regressor	**R2 : 0.5645 RMSE : 0.3127**	**R2 : 0.6689 RMSE : 0.2719**	
Using suitable combination of preprocessing techniques	Linear Regression	R2 : 0.3671 RMSE : 0.3770	R2 : 0.4919 RMSE : 0.3610	
	Decision Tree Regression	R2 : 0.2382 RMSE : 0.4136	R2 : 0.3978 RMSE : 0.3931	
	Ridge	R2 : 0.3672 RMSE : 0.3769	R2 : 0.4872 RMSE : 0.3627	
	SVR	R2 : 0.1563 RMSE : 0.4353	R2 : 0.5267 RMSE : 0.3484	
	K Neighbours	R2 : 0.2542 RMSE : 0.4092	R2 : 0.4341 RMSE : 0.3810	
	ADA Boosting	R2 : 0.2271 RMSE : 0.4166	R2 : 0.4613 RMSE : 0.3718	
	Random Forest Regressor	**R2 : 0.5645 RMSE : 0.3127**	**R2 : 0.6361 RMSE : 0.3056**	

Summary:

Number of Pre-processing Techniques applied with their names:

Number : 8

Names :

Replacing rows with NA values with mean (basic)
Dropping rows with NA values
Replacing rows with NA values with median
Replacing rows with NA values with mode
Min-Max Normalization
Standardization
Log transformation
Dropping rows with outliers

Number of Data Mining Algorithms applied with their names:

Number : 8

Names :

Linear Regression
Decision Tree Regression
Ridge
SVR
K-Neighbours
ADA Boosting
Gradient boosting regressor
Random Forest Regressor

Which algorithm showed highest performance after "All" pre-processing techniques and WHY?:

Random Forest Regressor showed the best prediction accuracy out of all algorithms used, the most accuracy being the result of **removal of outliers** with the highest R Squared value and lowest Root Mean Squared error.

Conclusion-Write in your own words:

The best model for this dataset is **Random Forest Regressor**, both before and after preprocessing. This can be concluded from the R Squared value and Root mean squared error. The closer the **R Squared value** is to 1 the more accurate the predictions are. Similarly the less the **RMSE** is the better the predictions are.

We have also observed that depending on the preprocessing, each model behaves very differently, for instance, the performance of the **SVR** model when **Normalization** and **Standardization** was applied and not applied differed significantly compared to other models.

And we also observed a fairly high increase in accuracy in all the models when appropriate preprocessing techniques were applied.

The most impactful preprocessing technique that we observed for our dataset is **removal of outliers.**

From all the observations, we can conclude that **preprocessing techniques** play a significant role in prediction algorithms and appropriate techniques should be employed for the best results.