FINAL PROJECT

CS536: DATA COMPLETION AND INTERPOLATION

Rithvik Reddy Ananth(ra830)
Srikar Reddy Nomula(sn671)
Guided By - Prof. Cowan Charles
Computer Science Department, Rutgers University

ABSTRACT

To take a data set and construct a system to interpolate the missing features from the present ones. Any feature might be missing, and that it should be predicted from the existing ones. For this assignment, we have taken the Portuguese "Vinho Verde" wine rating data set.

PROBLEMS ADDRESSED:

- **Data:** The data set is a CSV file and has 11 features and an output rating column. It contains a little less than 5000 rows. The entire data is numerical, and it is read using the panda's library as a data frame. Pandas and Numpy also provide flexibility to read and manipulate the data efficiently.
- **Interpolation - Input and output:** The loaded data set doesn't have any missing values in any of the features. So, we simulated the missing values in a unpredictable and completely random manner to avoid any pattern in missing values. This missing data is given input to a function, which selects a column with the least missing values and neglects the remaining missing value features columns. The meaning of interpolation or predicting the missing values here is to make sure that our data does not have any missing values for predicting the output Y. Hence we use interpolation to fill in all the missing values.
- **Model:** We have started with the model using mean imputation of values and later implemented a few complex models such as Linear regression, KNN, Lasso regression, and Ridge Regression models to find the data set's missing values.
- **Quantifying Error:** After predicting the missing values, we found the mean square error value to check the quantity of error by our prediction models as we already knew the original data. A detailed analysis of it is further explained below.
- **Training:** We first took the columns that had all the row values and one column which had the least missing features for training the models. This data was now split into two (based on the column having missing values). Let us say X was the column having missing features. The rows where X column values were available were taken as training data to train our model, and the weights were found. The testing data, which consisted of the rows where X column values were missing, were predicted using these weights.

- **Feature Selection:** We have chosen all the features which had all the row values for predicting the unknown values. As this was a wine data set, we felt that each ingredient had equal importance to contribute to the taste of the wine, which would affect the rating, which was the final output.
- **Validation and evaluation** is further explained in detail with the graphs.

DESCRIBE YOUR MODEL:

Our data set is all numerical with no categorical values and also doesn't consist of Boolean variables. So, we employed basic models such as naive mean and also regression models learnt such as linear regression, ridge regression and lasso regression. We also tried KNN to compare with this regression results. As stated, we had to choose regression over classification models based on our data set. Since, we need only to predict features we dropped the rating in the final data set. Our data set has no missing values. So, we thought it is better to simulate the missing values as close to real data as possible. We had employed a selection of random choice to select a random feature among the current features. This selection of features and also the number of features to be selected are in two ways. One is in random manner, the n number of features are selected randomly from the given features. Second, is in selective manner, values are removed in random percentage of values at random places.

To evaluate this model, we have an advantage of comparing it with real data values. We choose some standard statistical functions to represent the deviations along with the mean square error.

DESCRIBE YOUR TRAINING ALGORITHM:

We will discuss the implementation of different models in detail below.

- **Naive mean:** This is the basic model we have implemented to predict and interpolate missing feature values. This implementation helps us to predict the features. This can be used as base model to evaluate the performance of other models as well. This is computationally efficient in comparison to other models.
- **linear regression:** This is the first model we have tried to train and implement which can learn the dependencies between features. We haven't removed any features in this model. In most of the scenarios we tried to extract the relations and to use them in predicting the missing values.

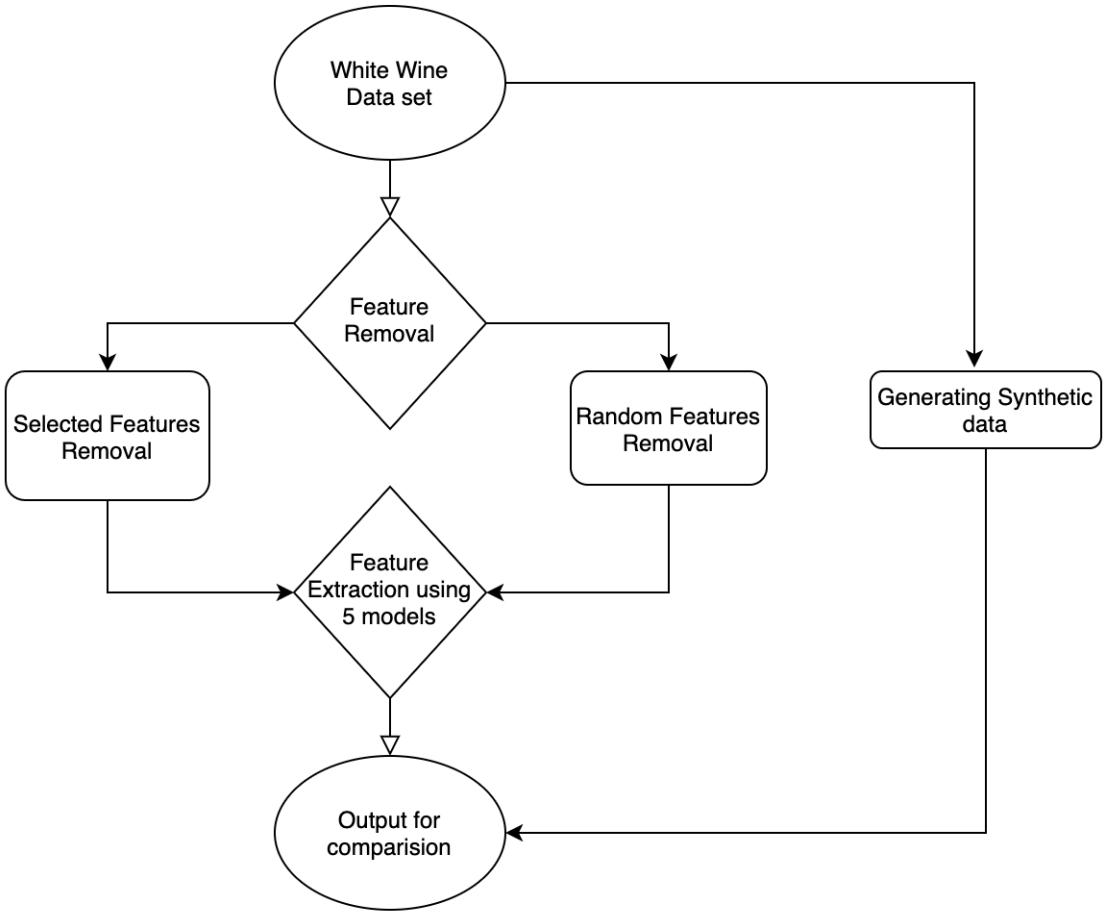


Figure 1: Flowchart of the Model

- **ridge regression:** We have implemented this model to check if we can improve the predicting accuracy over the above two models. This implementation is similar to linear regression. Ridge regression constant is tried on with different values and settled with 2

- **lasso regression:** We have implemented this model to check if we can improve the predicting accuracy over the above models. The constants are chosen based on the resulting mean square error and settled with 0.1.

- **KNN:** This is one model we have tried to implement other than regression methods. As this can be used both for classification and regression models, we wanted to try it and check if it can provide a more similar data in comparison to the original .This is slightly computationally expensive as it has to look for the entire training data set for each sample in test data. We have chosen the distance metric to be euclidean distance. We have chosen 10 nearest neighbours, and the mean of the values is used to replace the missing value.

Training algorithm: Other than mean imputation, where the prediction is not depended on features, we have selected the feature with least missing values to predict on. We thought this is the best approach to predict and build upon the features with complete values to predict the missing values in other feature. This can also propagate less MSE if we predict more accurately the initial feature. As the next features are trained on the updated training data. This updating the training data with the predicted column gives more samples to train and can reduce the error on further predicting. This method is employed until all the missing values are filled.

DESCRIBE YOUR MODEL VALIDATION:

In the implemented models other than mean imputation and KNN, we used regression methods to make a prediction. We calculated the MSE and deviations from the original data to measure the model accuracy. The regularisation models helps us to prevent over fitting and compare with the other models. As the data samples are increased by each iteration of filling missing values, the next features are trained on more number of samples, the more number of features are filled in, it is less likely to over fit to the certain part of the data.

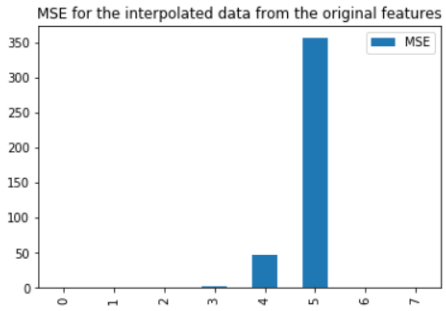


Figure 2: MSE of the model with 25 % of missing values

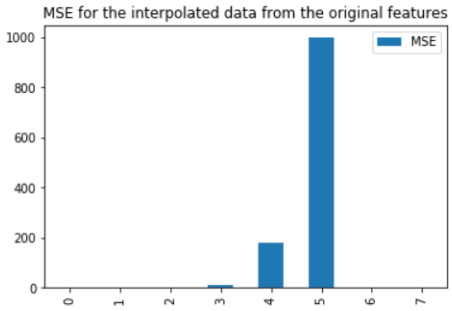


Figure 3: MSE of the model with around 60% of values missing

EVALUATE YOUR MODEL:

To evaluate our model, we started with simulation of missing random values. We increased the percentage of missing values to check the capability of our model. Upon increasing the percentage of missing values the errors gradually increased. Below are the results on increasing the percentage of missing values from 25 percent to 60 percent.

We also tried to check if change in the number of features with missing values can show an effect on overall accuracy. To do this, random removal is modified to select a set of features to compare the same features. There is not a considerable increase in error, which would tell us that the features are not completely correlated.

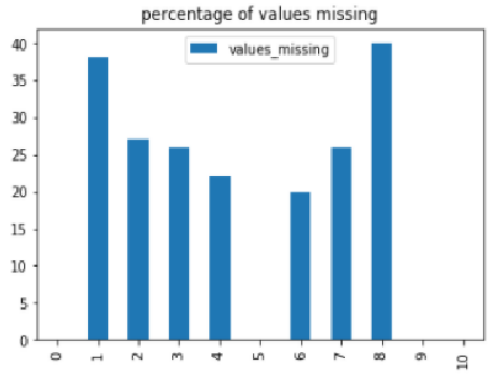


Figure 4: Missing values in percentage by feature

Naive Mean method for missing value prediction:

As the values of the 4th column are higher compared to other values, there is a difference in standard deviation and variance. Mean square error, mean difference and variance difference plots indicate that they are the highest for column 4 (total sulfur dioxide). The negative values in mean difference and variance graphs shows that the mean imputation has reduced the spread of the overall distribution of the data.

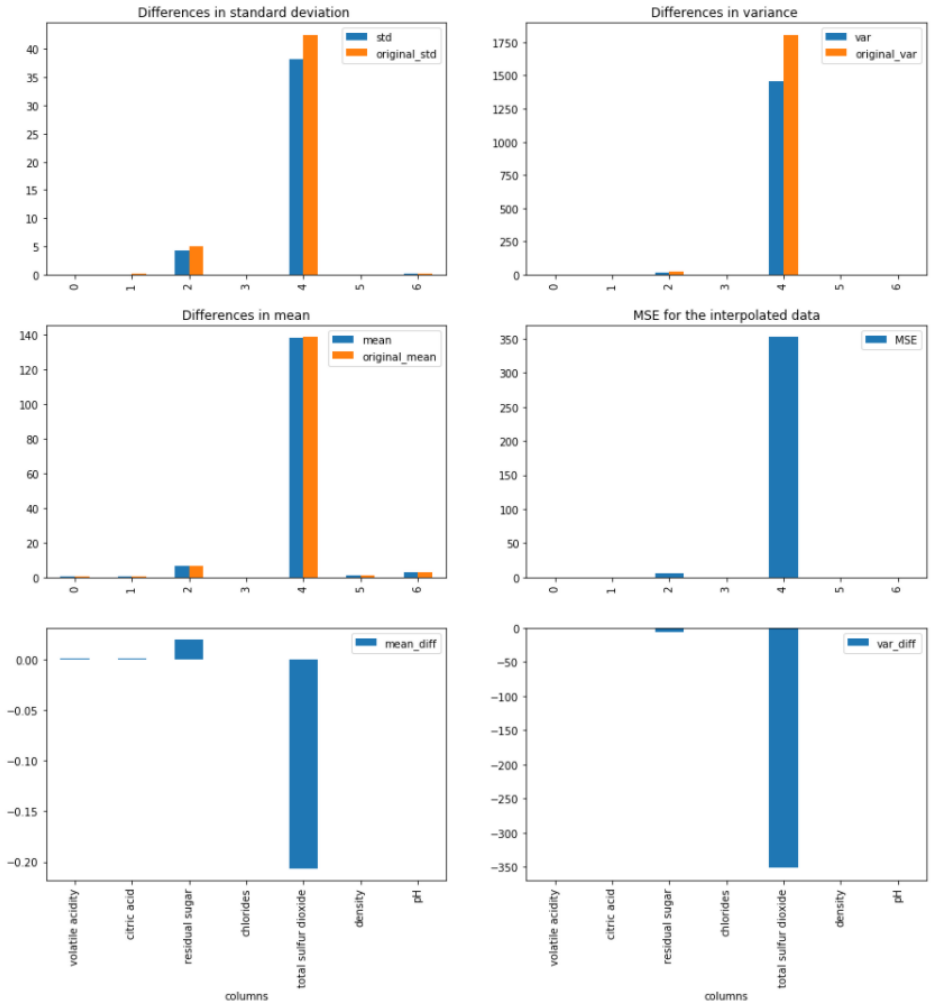


Figure 5: Missing value prediction using Naive Mean method

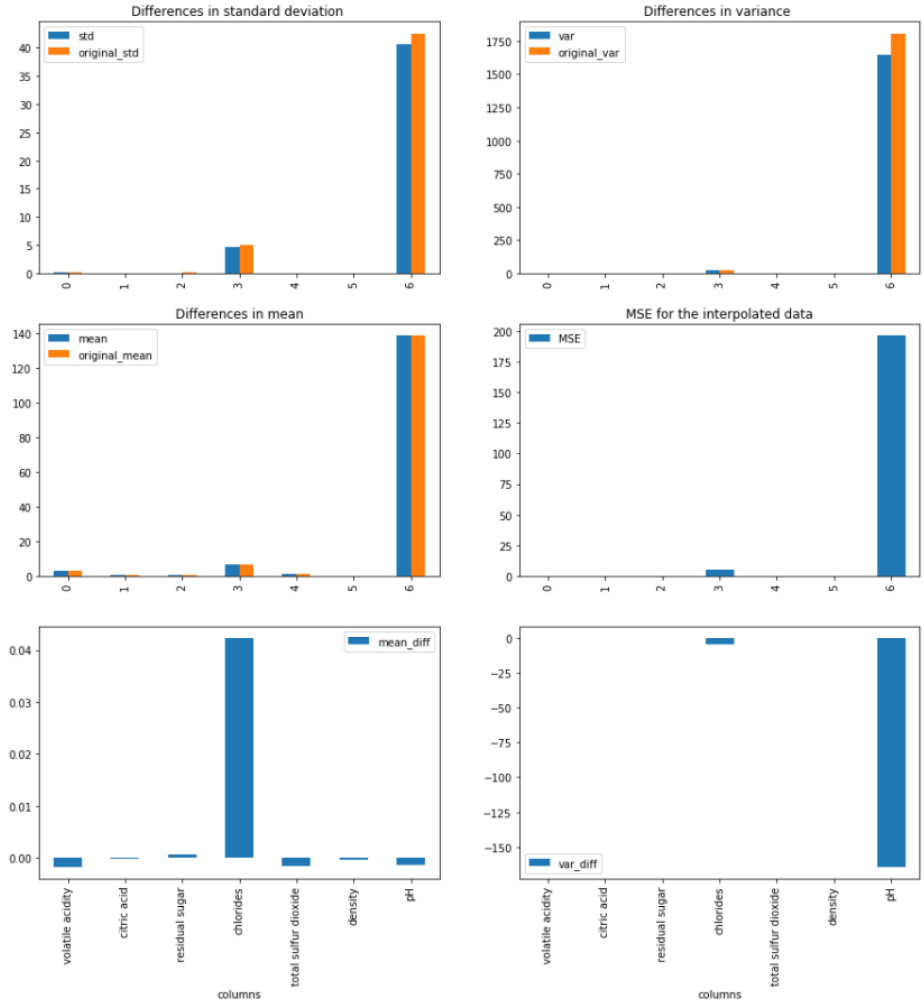


Figure 6: Missing value prediction using Linear-Regression method

Linear-Regression method for missing value prediction:

The difference in original mean and standard deviation is not much but there is a slight difference in original and predicted value's variance. The mean square error and difference in variance is highest while predicting the pH values, whereas the mean difference is higher for chlorides. This is because the pH attribute is not dependent on any other feature and this can be explained using the correlation matrix where the values are within +0.15 to -0.15.

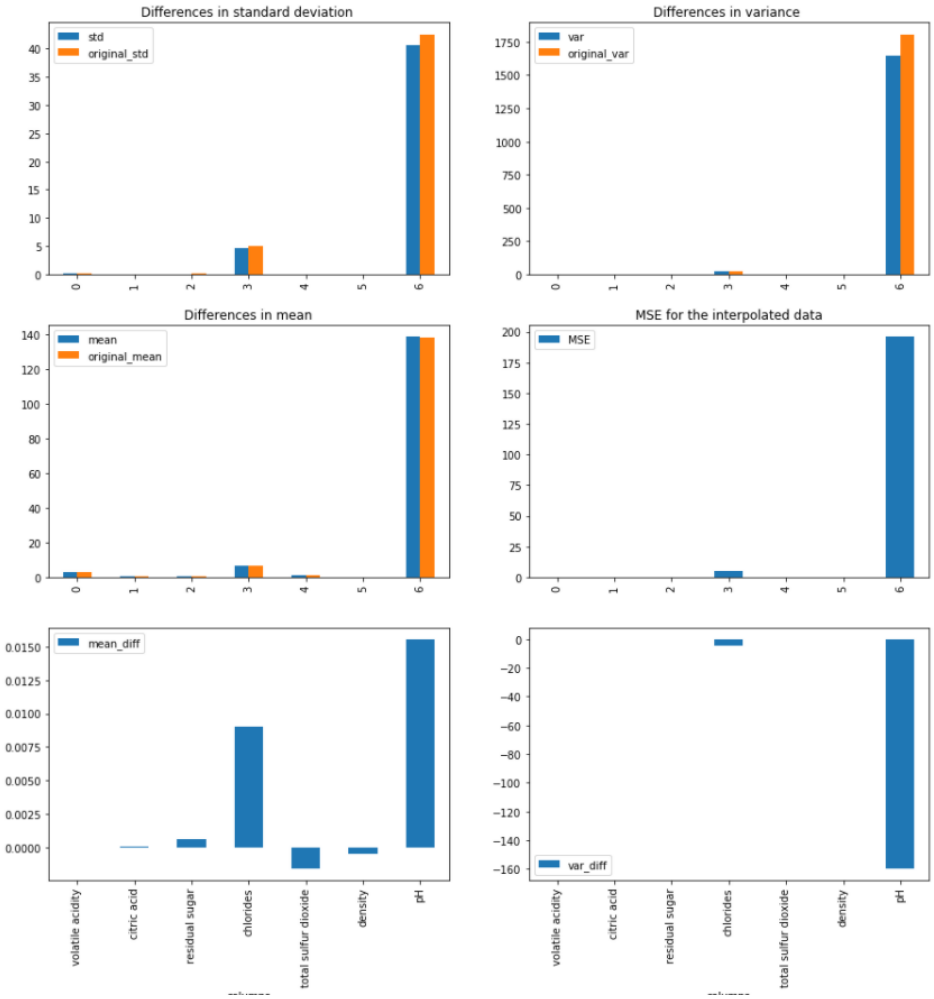


Figure 7: Missing value prediction using Lasso-Regression method

Lasso-Regression method for missing value prediction:

There is not much of a difference in original and predicted value's standard deviation, mean and variance. Among the features who's values are missing, pH has the highest difference in mean square error, mean, and variance.

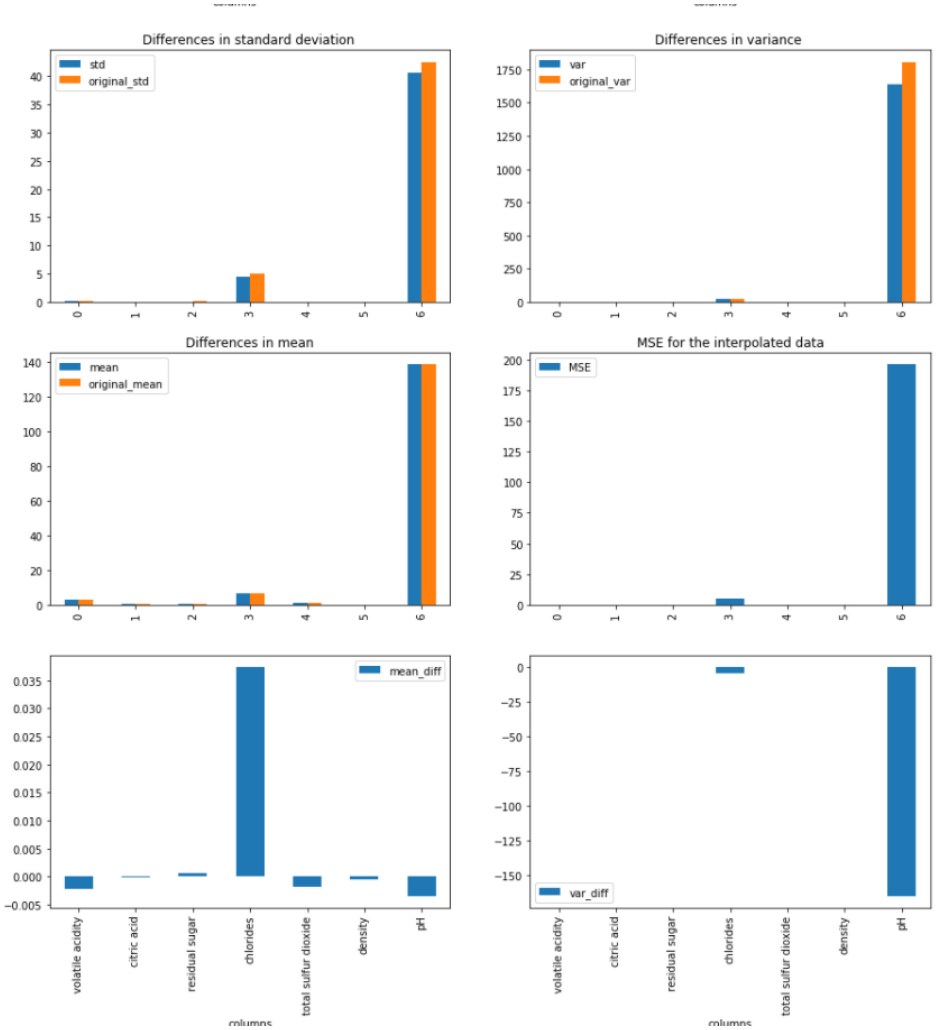


Figure 8: Missing value prediction using Ridge-Regression method

Ridge-Regression method for missing value prediction:

The ridge regression was mostly similar to the lasso regression model but the only difference being mean difference is highest for chlorides than for pH columns.

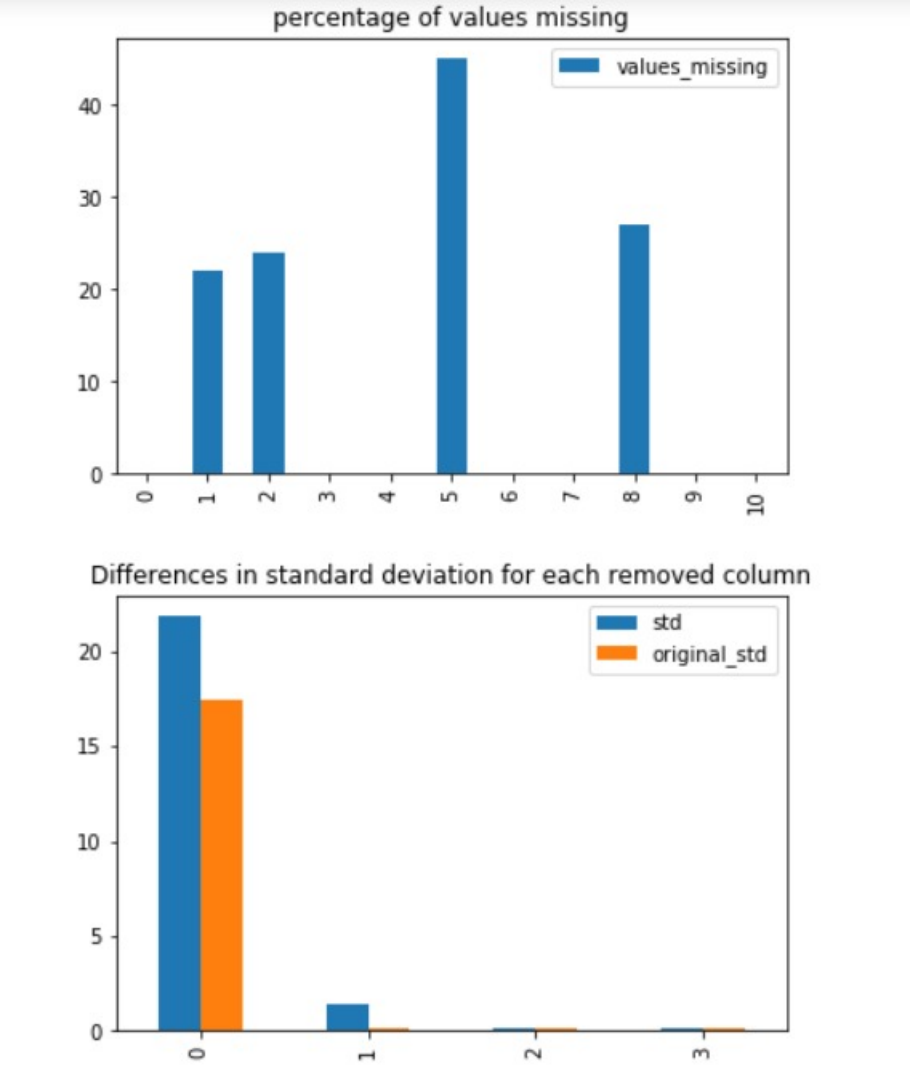


Figure 9: Missing values prediction using KNN

KNN for missing value prediction:

We gave a range of 20 - 45 percent and number of columns as 4 and the function randomly removed 22, 25, 45, 27 percent of values from columns 1, 2, 5 , and 8. We can observe a little standard deviation in the first column and not much in the other 3. This is also the same for difference in variance and mean. This indicates that even though the percentage of missing values is the least in the first column the difference in mean, standard deviation, and variance is high.

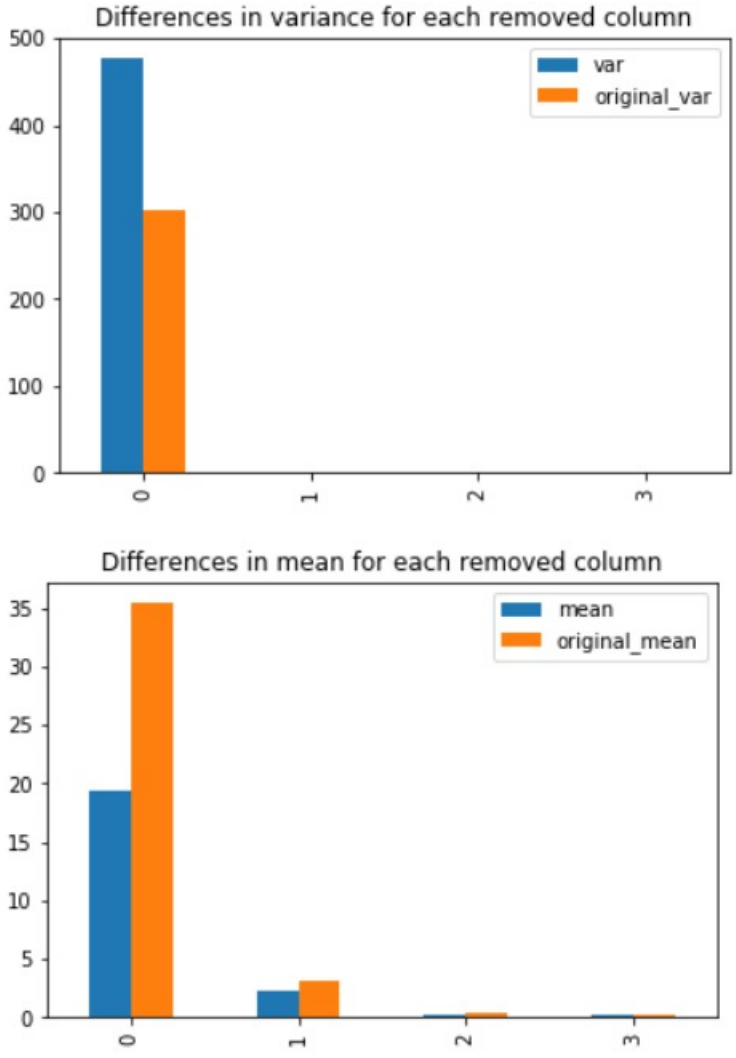
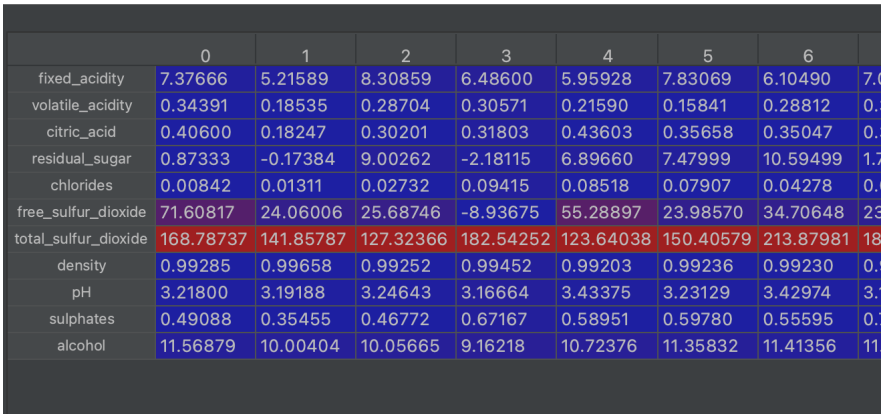


Figure 10: Difference in variance and mean using KNN



	0	1	2	3	4	5	6	
fixed_acidity	7.37666	5.21589	8.30859	6.48600	5.95928	7.83069	6.10490	7.(
volatile_acidity	0.34391	0.18535	0.28704	0.30571	0.21590	0.15841	0.28812	0.
citric_acid	0.40600	0.18247	0.30201	0.31803	0.43603	0.35658	0.35047	0.
residual_sugar	0.87333	-0.17384	9.00262	-2.18115	6.89660	7.47999	10.59499	1.7
chlorides	0.00842	0.01311	0.02732	0.09415	0.08518	0.07907	0.04278	0.
free_sulfur_dioxide	71.60817	24.06006	25.68746	-8.93675	55.28897	23.98570	34.70648	23
total_sulfur_dioxide	168.78737	141.85787	127.32366	182.54252	123.64038	150.40579	213.87981	18
density	0.99285	0.99658	0.99252	0.99452	0.99203	0.99236	0.99230	0.
pH	3.21800	3.19188	3.24643	3.16664	3.43375	3.23129	3.42974	3.1
sulphates	0.49088	0.35455	0.46772	0.67167	0.58951	0.59780	0.55595	0.1
alcohol	11.56879	10.00404	10.05665	9.16218	10.72376	11.35832	11.41356	11

Figure 11: Synthetic Data

GENERATING SYNTHETIC DATA

Synthetic data is a data that is generated in a programmatic way using the existing data. Its main purpose is to be rich enough to help ML engineers to experiment with various classifications, regressions and clustering algorithms. It should be random and one can choose a wide variety of statistical distributions to base this data upon. The challenge would be to control and tune this random data generation process. We have a library called SK learn to generate synthetic data. For regression models we can use dataset.make-regression, for classification dataset.make-classification, for clustering algorithms datasets.make-blobs etc can be used to generate the artificial data.
As the entire data set is numeric we generated our synthetic data in the following way:

- From the available data set, find the mean and standard deviation of each column.
- By using normal distribution in numpy, the inputs mean and standard deviation were given along with the sample size.
- We thus generate the data for all columns keeping the size of each column same.
- All the columns are merged and the entire data generated is converted to a data frame using pandas.

ACKNOWLEDGMENT

We would like to thank Professor Cowan W. Charles and fellow students who have clarified doubts on the discussion board and helped us understand the problem statement better.

REFERENCES

Class notes

Lecture videos

Discussion board on Canvas.

Towards Data Science website.