DS 203 Assignment E3

Siddharth Verma 22B2153

Regression Algorithms Analysis

Siddharth Verma[a]

[a] *Indian Institute of Technology Bombay*

Abstract

This Report is on Exercise E3, which is aimed at analyzing diverse regression algorithms applied to a given dataset with the goal of comprehending their respective characteristics, performance metrics, and inherent advantages. It is also a summary of the code in the Jupyter Notebook `E3.ipynb` and explores the learnings derived from the code. Additionally, it includes descriptions of the Sklearn functions used in the notebook, presenting them in a two-column table format ($Functionname|Description$).

1. Code Summary

The provided code in E3.ipynb performs the following tasks:

1. Reading data from an Excel file ('E3-MLR3.xlsx') into training and test datasets.
2. Separating features and target variables in both training and test datasets.
3. Using PolynomialFeatures to create polynomial features from the original features.
4. Augmenting the test data by incorporating additional features and subsequently saving the augmented dataset to a CSV file named 'augmented_data.csv'.
5. Implementing several regression algorithms, including Linear Regression, SVM Regression, RandomForest, XGBoost, KNN, and Neural Network.
6. Evaluating and comparing the performance of each algorithm using metrics such as R-squared, Mean Squared Error (MSE), Durbin-Watson statistic, Jarque-Bera statistic, and Jarque-Bera p-value.
7. Creating scatter plots and residual plots for both training and test data.

2. Learnings

The provided code gives a analytical exploration of different regression algorithms and thier performance metrics:

- Study of Comparision of Tree-based Algorithms like XGBoost and Neural Network based Algorithm is one of the most insightful gains in report.
- Understanding the use of PolynomialFeatures to capture non-linear relationships in the data.
- Application of statistical metrics like Durbin-Watson and Jarque-Bera for regression diagnostics.

3. Sklearn Function Descriptions

Table 1: Sklearn Function Descriptions

Function Name	**Description**
`PolynomialFeatures()`	Generates polynomial features for the input data up to the specified degree.
`LinearRegression()`	Implements linear regression using the Ordinary Least Squares method.
`SVR(kernel)`	Support Vector Regression with a specified kernel function (in this case, polynomial kernel).
`RandomForestRegressor()`	Random Forest regression algorithm for ensemble learning.
`GradientBoostingRegressor`	Gradient Boosting regression algorithm.
`MLPRegressor()`	Multi-layer Perceptron (Neural Network) regression with specified hidden layer sizes and maximum iterations.
`KNeighborsRegressor()`	K-Nearest Neighbors regression algorithm.
`mean_squared_error(y_true, y_pred)`	Computes the mean squared error between the true and predicted values.
`r2_score(y_true, y_pred)`	Computes the R-squared (coefficient of determination) score.

*

Email address: `22b2153@iitb.ac.in` (Siddharth Verma)

4. AUGMENTED DATA CSV Analysis

In this section, we analyze the augmented matrices generated for different polynomial degrees using the PolynomialFeatures function. Visual representations of the augmented matrices for degrees 1, 3, 6, and 10 are provided below.

Degree 1

	1	x1	y
count	152.0	152.000000	152.000000
mean	1.0	0.487143	5.527713
std	0.0	0.296484	3.532568
min	1.0	0.001429	-0.058062
25%	1.0	0.220000	2.292668
50%	1.0	0.477143	4.334332
75%	1.0	0.753571	8.801640
max	1.0	0.992857	12.217289

Figure 1: Augmented Matrix for Degree 1

Degree 1 Analysis

- The range of x1 lies in interval [0.001429,0.992857] and The values of mean and standard deviation suggest x1 is normally distributed.The mean value of x1 is almost 11 times that of mean value of y.
- y doesn't follow normal distribution

Degree 3

	1	x1	x1^2	x1^3	y
count	152.0	152.000000	152.000000	1.520000e+02	152.000000
mean	1.0	0.487143	0.324633	2.435050e-01	5.527713
std	0.0	0.296484	0.299569	2.814346e-01	3.532568
min	1.0	0.001429	0.000002	2.915452e-09	-0.058062
25%	1.0	0.220000	0.048498	1.071153e-02	2.292668
50%	1.0	0.477143	0.227667	1.086318e-01	4.334332
75%	1.0	0.753571	0.567884	4.279617e-01	8.801640
max	1.0	0.992857	0.985765	9.787241e-01	12.217289

Figure 2: Augmented Matrix for Degree 3

Degree 3 Analysis

-]Standard deviation first increases then decreases suggesting significant change in p -value from x1$\hat{2}$ to x1$\hat{3}$as well.
- x2 have range from around 0 to 0.98 and is not normally distributed but skewed.Similar in case of x3

Degree 6

	1	x1	x1^2	x1^3	x1^4	x1^5	x1^6	y
count	152.0	152.000000	152.000000	1.520000e+02	1.520000e+02	1.520000e+02	1.520000e+02	152.000000
mean	1.0	0.487143	0.324633	2.435050e-01	1.945372e-01	1.616509e-01	1.379790e-01	5.527713
std	0.0	0.296484	0.299569	2.814346e-01	2.621030e-01	2.446215e-01	2.293549e-01	3.532568
min	1.0	0.001429	0.000002	2.915452e-09	4.164931e-12	5.949902e-15	8.499860e-18	-0.058062
25%	1.0	0.220000	0.048498	1.071153e-02	2.370045e-03	5.252763e-04	1.165994e-04	2.292668
50%	1.0	0.477143	0.227667	1.086318e-01	5.183428e-02	2.473324e-02	1.180182e-02	4.334332
75%	1.0	0.753571	0.567884	4.279617e-01	3.225233e-01	2.430682e-01	1.831917e-01	8.801640
max	1.0	0.992857	0.985765	9.787241e-01	9.717332e-01	9.647923e-01	9.579009e-01	12.217289

Figure 3: Augmented Matrix for Degree 6

Degree 6 Analysis

- The variables in column have standard deviation decreasing x1$\hat{2}$ onwards till x1$\hat{6}$.

Degree 10

	1	x1	x1^2	x1^3	x1^4	x1^5
nt	152.0	152.000000	152.000000	1.520000e+02	1.520000e+02	1.520000e+02
an	1.0	0.487143	0.324633	2.435050e-01	1.945372e-01	1.616509e-01
td	0.0	0.296484	0.299569	2.814346e-01	2.621030e-01	2.446215e-01
in	1.0	0.001429	0.000002	2.915452e-09	4.164931e-12	5.949902e-15
%	1.0	0.220000	0.048498	1.071153e-02	2.370045e-03	5.252763e-04
%	1.0	0.477143	0.227667	1.086318e-01	5.183428e-02	2.473324e-02
%	1.0	0.753571	0.567884	4.279617e-01	3.225233e-01	2.430682e-01
ax	1.0	0.992857	0.985765	9.787241e-01	9.717332e-01	9.647923e-01

	x1^6	x1^7	x1^8	x1^9	x1^10	y
nt	1.520000e+02	1.520000e+02	1.520000e+02	1.520000e+02	1.520000e+02	152.000000
an	1.379790e-01	1.200933e-01	1.060908e-01	9.482863e-02	8.557701e-02	5.527713
td	2.293549e-01	2.161005e-01	2.045575e-01	1.944456e-01	1.855297e-01	3.532568
in	8.499860e-18	1.214266e-20	1.734665e-23	2.478093e-26	3.540133e-29	-0.058062
%	1.165994e-04	2.591995e-05	5.769724e-06	1.285923e-06	2.869270e-07	2.292668
%	1.180182e-02	5.631457e-03	2.687178e-03	1.282260e-03	6.118704e-04	4.334332
%	1.831917e-01	1.380684e-01	1.040624e-01	7.843394e-02	5.911877e-02	8.801640
ax	9.579009e-01	9.510588e-01	9.442655e-01	9.375207e-01	9.308242e-01	12.217289

Figure 4: Augmented Matrix for Degree 10

Degree 10 Analysis

- The range of values from x1 to x1$\hat{10}$ decreases with standard deviation increasing till x1$\hat{2}$ then decreasing

5. Qualitative Analysis of Data Across Degrees for Different Algoritms

In this section, we conduct a qualitative analysis of the regression models' performance at different polynomial degrees. The insights gained provide a deeper understanding of the impact of polynomial degree on the model's behavior.

Degree 1

Metrics - Train Data:

	R-squared	MSE	Durbin-Watson	Jarque-Bera	JB P-value
Linear Regression	0.833674	1.985063	0.059761	51.466506	6.670987e-12
SVM Regression	0.903049	1.157094	0.102699	45.292046	1.462033e-10
RandomForest	0.999025	0.011632	2.956472	0.713402	6.999818e-01
XGBoost	0.997146	0.034058	2.372918	1.046859	5.924852e-01
knn	0.995862	0.049392	2.506218	0.724274	6.961871e-01
Neural Network	0.833556	1.986476	0.059726	53.945555	1.931397e-12

Metrics - Test Data:

	R-squared	MSE	Durbin-Watson	Jarque-Bera	JB P-value
Linear Regression	0.818068	2.255404	0.079118	12.500550	0.001930
SVM Regression	0.879811	1.489974	0.120164	3.745234	0.153721
RandomForest	0.992334	0.095036	1.869900	0.017492	0.991292
XGBoost	0.993488	0.080732	1.876249	0.424332	0.808830
knn	0.992945	0.087457	1.992313	0.752572	0.686406
Neural Network	0.818194	2.253841	0.079113	13.069379	0.001452

Figure 5: Qualitative Analysis for Degree 1

- Random Forest having more $R\hat{2}$ value and lower MSE than XGBoost coincides with the fact that Random forests are often better at capturing interaction effects between features. Even with just one feature (x1), there might be implicit interaction effects with other univariate relationships that a random forest could exploit.

- SVM Regression on Test Data: The Support Vector Machine (SVM) Regression model stands out with a relatively high R-squared value on the test data. While not as extreme as Random Forest on the training data, the SVM model demonstrates good explanatory power on the test set, capturing a substantial portion of the variance in the target variable.

- Neural Network Durbin-Watson Statistic on Training Data:The Neural Network model has a remarkably low Durbin-Watson statistic on the training data (0.059726). The Durbin-Watson statistic measures the autocorrelation of residuals, and a value close to 0 indicates a high positive autocorrelation. This suggests that there might be a pattern or structure in the residuals of the Neural Network model on the training data, which could potentially impact the model's performance or signal a violation of the independence assumption..

Degree 3

- Similar to the previous set, the Random Forest model exhibits an extremely high R-squared value and a very low MSE on the training data. This indicates an almost perfect fit to the training data, capturing the majority of the variance with minimal prediction errors.

Metrics - Train Data:

	R-squared	MSE	Durbin-Watson	Jarque-Bera	JB P-value
Linear Regression	0.905822	1.123991	0.105657	16.909651	2.128707e-04
SVM Regression	0.917078	0.989658	0.119795	27.746584	9.438561e-07
RandomForest	0.999034	0.011533	2.934431	1.802293	4.061038e-01
XGBoost	0.997146	0.034058	2.372918	1.046859	5.924852e-01
knn	0.995895	0.048993	2.505231	0.733304	6.930509e-01
Neural Network	0.960628	0.469898	0.250335	190.295692	4.762287e-42

Metrics - Test Data:

	R-squared	MSE	Durbin-Watson	Jarque-Bera	JB P-value
Linear Regression	0.886990	1.400982	0.127507	1.239081	0.538192
SVM Regression	0.899065	1.251286	0.140719	2.799847	0.246616
RandomForest	0.992097	0.097968	1.865904	0.136479	0.934037
XGBoost	0.993005	0.086712	1.858895	0.660034	0.718911
knn	0.993051	0.086146	1.993466	1.266390	0.530893
Neural Network	0.947454	0.651405	0.259341	21.467475	0.000022

Figure 6: Qualitative Analysis for Degree 3

- The Neural Network model has an exceptionally high Jarque-Bera statistic on the training data, accompanied by an extremely low p-value. The Jarque-Bera test assesses whether the residuals of a model have a normal distribution. In this case, the high statistic and low p-value suggest a departure from normality in the residuals, indicating a potential violation of the normality assumption.

- While the Neural Network model performs well on the test data with a high R-squared value, the MSE is relatively higher compared to other models. This suggests that, although the model explains a significant portion of the variance, there is still some room for improvement in reducing prediction errors.

Degree 6

Metrics - Train Data:

	R-squared	MSE	Durbin-Watson	Jarque-Bera	JB P-value
Linear Regression	0.976261	0.283320	0.417086	11.050745	3.984384e-03
SVM Regression	0.917700	0.982234	0.120617	28.434280	6.692287e-07
RandomForest	0.999004	0.011886	2.939473	3.018519	2.210736e-01
XGBoost	0.997146	0.034058	2.372918	1.046859	5.924852e-01
knn	0.995895	0.048993	2.505231	0.733304	6.930509e-01
Neural Network	0.975493	0.292484	0.402465	131.610163	2.637644e-29

Metrics - Test Data:

	R-squared	MSE	Durbin-Watson	Jarque-Bera	JB P-value
Linear Regression	0.972925	0.335644	0.528190	3.916151	0.141130
SVM Regression	0.902527	1.208370	0.145266	3.932627	0.139972
RandomForest	0.991969	0.099563	1.832309	0.388490	0.823456
XGBoost	0.992956	0.087327	1.852981	0.746789	0.688394
knn	0.993051	0.086146	1.993466	1.266390	0.530893
Neural Network	0.967731	0.400034	0.411402	13.659363	0.001081

Figure 7: Qualitative Analysis for Degree 6

- The Linear Regression model exhibits a high R-squared value (0.976261) on the training data, indicating a strong linear relationship between the predictors and the target variable. The low MSE (0.283320) suggests accurate predictions with minimal errors.

- The SVM model performs well on the test data with a decent R-squared value (0.902527) and a moderate MSE (1.208370). The Durbin-Watson statistic is above 1, suggesting potential negative autocorrelation in the residuals.

- The Durbin-Watson statistic for some models, particularly Neural Network and Linear Regression, suggests potential autocorrelation in the residuals.Also note thier residual plots do show us a pattern like their own scatter plot indeed.

Degree 10

Metrics - Train Data:

	R-squared	MSE	Durbin-Watson	Jarque-Bera	JB P-value
Linear Regression	0.992987	0.083693	1.403681	28.853427	5.426976e-07
SVM Regression	0.912128	1.048728	0.112993	36.437003	1.224068e-08
RandomForest	0.999040	0.011462	2.923197	0.227198	8.926159e-01
XGBoost	0.997146	0.034058	2.372918	1.046859	5.924852e-01
knn	0.995895	0.048993	2.505231	0.733304	6.930509e-01
Neural Network	0.961131	0.463899	0.254023	192.063281	1.967834e-42

Metrics - Test Data:

	R-squared	MSE	Durbin-Watson	Jarque-Bera	JB P-value
Linear Regression	0.991865	0.100844	1.653619	3.724290	0.155339
SVM Regression	0.897935	1.265296	0.138487	5.287584	0.071091
RandomForest	0.991907	0.100323	1.851642	0.312001	0.855559
XGBoost	0.992928	0.087674	1.849816	0.844895	0.655441
knn	0.993051	0.086146	1.993466	1.266390	0.530893
Neural Network	0.947065	0.656230	0.258622	17.702392	0.000143

Figure 8: Qualitative Analysis for Degree 10

- Random Forest outshines other models with an extraordinary R-squared of 0.999040 and minimal MSE of 0.011462 on the training set. Its exceptional performance extends to the test data, with an R-squared of 0.991907 and a remarkably low MSE of 0.09. The model excels in both training and generalization.
- Exceptionally high Jarque-Bera values for Neural Network residuals (192.063281 for training) indicate a departure from the normal distribution. Non-normal residuals can affect the reliability of statistical inferences. Investigating and rectifying the non-normality in residuals is crucial for ensuring the validity of model results and improving the overall model quality.

6. Variations in Metrics Across Degrees for Different Regression Algorithms

In this section, we examine how various regression algorithms perform across different polynomial degrees. We analyze the training and testing metrics for each algorithm, providing insights into their behavior with increasing polynomial complexity.

6.0.1. Linear Regression

Degree 1:

- High R-squared (0.83) and low MSE (1.99) on the training data.
- Reasonable generalization to the test data (R-squared: 0.82, MSE: 2.26).

Degree 3:

- Improved training data performance (R-squared: 0.91, MSE: 1.12).
- Slight overfitting observed in the test data (R-squared: 0.89, MSE: 1.40).

Degree 6 and 10:

- Continued improvement in training data (R-squared: 0.98-0.99, MSE: 0.28-0.33).
- Consistent generalization to the test data with a slight increase in MSE.

6.0.2. SVM Regression

Degree 1:

- Strong performance on both training and test data.

Degree 3:

- Maintains strong predictive power on both training (R-squared: 0.92, MSE: 0.99) and test data (R-squared: 0.90, MSE: 1.25).

Degree 6 and 10:

- Consistent performance with slight variations across degrees.

6.0.3. RandomForest

Degree 1:

- Exceptional performance with a perfect fit on the training data.
- Generalizes well to the test data (R-squared: 0.99, MSE: 0.10).

Degree 3:

- Similar strong performance with high R-squared and low MSE on both datasets.

Degree 6 and 10:

- Continued excellence with minimal variations.

6.0.4. XGBoost

Degree 1:

- Strong performance on both training and test data.

Degree 3:

- Maintains high R-squared and low MSE on both datasets.

Degree 6 and 10:

- Consistent performance with slight variations across degrees.

6.0.5. k-Nearest Neighbors (KNN)

Degree 1:

- Strong performance on both training and test data.

Degree 3:

- Maintains high R-squared and low MSE on both datasets.

Degree 6 and 10:

- Consistent performance with slight variations across degrees.

6.0.6. Neural Network

Degree 1:

- Strong performance on both training and test data.

Degree 3:

- Maintains high R-squared but with an increase in MSE on the test data.

Degree 6 and 10:

- Consistent performance with slight variations across degrees.

6.0.7. Overall Trends

- Polynomial degrees generally lead to better performance on the training data.
- Higher-degree polynomials may result in overfitting, leading to a decrease in generalization on the test data.
- RandomForest and XGBoost consistently perform well across different degrees.
- Neural Network shows strong predictive power but exhibits sensitivity to the choice of the degree.

6.1. Discussion on Specific Questions

6.1.1. When degree = 1 which method(s) result in acceptable regression models? Why?

For degree 1, Linear Regression, SVM Regression, RandomForest, XGBoost, KNN, and Neural Network all demonstrate acceptable regression models. These models perform well on both training and test data, with reasonable R-squared values and MSE. Linear Regression and SVM Regression show strong performance due to the simplicity of the model, while ensemble methods like RandomForest and XGBoost also provide robust results.

6.1.2. When degree = 6 which method(s) result in acceptable regression models? Why?

When degree is increased to 6, RandomForest and XGBoost continue to demonstrate exceptional performance, maintaining high R-squared and low MSE on both training and test data. These ensemble methods effectively handle the complexity introduced by higher-degree polynomials while avoiding overfitting. Other methods like Linear Regression, SVM Regression, KNN, and Neural Network also maintain acceptable regression models, though slight variations and overfitting may occur.

6.1.3. As the value of degree is increased to 10 which regression methods show the most impact? Why?

As the degree is increased to 10, the impact on regression methods becomes more noticeable. Polynomial regression models, including Linear Regression, SVM Regression, and Neural Network, show increased accuracy on the training data but may suffer from overfitting on the test data. RandomForest and XGBoost continue to handle the complexity well, maintaining high performance with minimal impact. The non-linear relationships captured by these ensemble methods make them more robust to higher degrees.

6.1.4. Why do Non-parametric methods like KNN / Tree-based methods generate good results even without feature engineering?

Non-parametric methods like KNN and Tree-based methods (RandomForest, XGBoost) often generate good results without extensive feature engineering due to their ability to capture complex relationships and patterns in the data. These methods are less sensitive to the distribution and scale of features, making them suitable for various types of data. KNN relies on distance metrics to find relationships, while tree-based methods partition the feature space, allowing them to adapt to non-linearities without explicit feature transformations.

6.1.5. What are the limitations of the non-parametric methods?

Non-parametric methods have limitations: - **Computational Intensity:** KNN can be computationally intensive, especially with large datasets. - **Overfitting:** Tree-based methods can overfit if not properly tuned. - **Interpretability:** The black-box nature of neural networks and certain tree-based models may hinder interpretability. - **Sensitive to Noise:** Non-parametric methods may be sensitive to noisy data points.

6.1.6. Given the results, should Linear Regression be used at all? Why, when? Justify your answer.

Linear Regression can still be valuable depending on the context and requirements. It is interpretable, computationally efficient, and provides a baseline for comparison. In cases where relationships are predominantly linear, Linear Regression can offer a simple and transparent model. However, for complex, non-linear relationships, especially when degrees are increased, ensemble methods like RandomForest and XGBoost might be preferred, as they handle non-linearities effectively while maintaining good generalization. The choice depends on the trade-off between model interpretability and capturing intricate patterns in the data.

7. Key Insights from SVR and MLPRegressor Analysis

```
SVR with Kernel: linear
Train R-squared: 0.8847567044107317
Train MSE: 1.3754049628242033
Test R-squared: 0.8713749612198909
Test MSE: 1.594556751257891

SVR with Kernel: poly
Train R-squared: 0.912128493512208
Train MSE: 1.0487283055917993
Test R-squared: 0.8979347980919988
Test MSE: 1.2652960754331055

SVR with Kernel: rbf
Train R-squared: 0.9205899579513233
Train MSE: 0.9477424727690588
Test R-squared: 0.9043499842037205
Test MSE: 1.1857674049499867

MLPRegressor with Hidden Layer Sizes: (10, 10)
Train R-squared: 0.9423941036547882
Train MSE: 0.6875144911121343
Test R-squared: 0.9263505459909079
Test MSE: 0.913027783940433
```

Figure 9: Description of the test performed

In the examination of Support Vector Regression (SVR) with different kernels and Multi-layer Perceptron (MLP) Regressor models with varying hidden layer sizes, several key insights emerged.

7.0.1. Support Vector Regression (SVR)

Three different kernels were applied to SVR to capture diverse degrees of complexity in the data relationships:

Linear Kernel:. The linear kernel exhibited robust performance, achieving approximately 88.5% and 87.1% R-squared on the training and test data, respectively. The corresponding Mean Squared Errors (MSE) were 1.38 and 1.59, suggesting that a linear model adequately captured the data's linear patterns.

Polynomial Kernel:. Designed to capture non-linear relationships, the polynomial kernel demonstrated improved predictive capabilities. With an R-squared of about 91.2% on the training set and 89.8% on the test set, along with MSE values of 1.05 and 1.27, respectively, the polynomial kernel showcased its ability to model more intricate patterns.

RBF Kernel:. The Radial Basis Function (RBF) kernel, known for its versatility in capturing complex relationships, outperformed the linear and polynomial kernels. Achieving an outstanding R-squared of 92.1% on the training data and 90.4% on the test data, with MSE values of 0.95 and 1.19, respectively, the RBF kernel adapted well to intricate dataset structures, emphasizing its flexibility in handling non-linear relationships.

7.0.2. Multi-layer Perceptron (MLP) Regressor

The exploration extended to MLP Regressor models with varying hidden layer sizes, revealing a noteworthy trend:

Increasing Hidden Layer Sizes:. As the hidden layer size increased, the MLPRegressor demonstrated progressively superior performance. Notably, a configuration with hidden layer sizes of (20, 20) produced exceptional results, achieving a training R-squared of 99.4% with an MSE of 0.07 and a test R-squared of 99.3% with an MSE of 0.09. This underscores the MLPRegressor's ability to learn intricate patterns, showcasing its potential for capturing complex relationships in the data.

Major Learnings and Overall Knowledge

Algorithm Performance Analysis

The document meticulously evaluates the performance of various regression algorithms, such as Linear Regression, Support Vector Regression (SVR), RandomForest, XGBoost, KNN, and Neural Network. It sheds light on their strengths and weaknesses across different polynomial degrees, offering insights into their adaptability to complex relationships within the data.

Polynomial Regression and Feature Engineering

A crucial aspect covered in the document is the use of PolynomialFeatures for generating polynomial features from the original dataset. The analysis demonstrates how increasing polynomial degrees can impact model performance, highlighting the importance of understanding the trade-offs associated with feature engineering.

Statistical Metrics for Model Evaluation

The document introduces and utilizes various statistical metrics for evaluating regression models, including R-squared, Mean Squared Error (MSE), Durbin-Watson statistic, Jarque-Bera statistic, and Jarque-Bera p-value. These metrics provide a comprehensive assessment of model accuracy, residuals behavior, and potential violations of assumptions.

Exploration of Non-parametric Methods

The document explores the effectiveness of non-parametric methods like KNN and tree-based methods (RandomForest, XGBoost) without extensive feature engineering. It emphasizes their ability to capture complex relationships and patterns in the data, showcasing their versatility and robustness.

Insights from SVR and MLPRegressor Analysis

In-depth analysis of Support Vector Regression (SVR) with different kernels and Multi-layer Perceptron (MLP) Regressor with varying hidden layer sizes provides deeper insights into non-linear relationships. It demonstrates the impact of different configurations on model performance, showcasing the adaptability of these algorithms to various dataset structures.

Consideration of Model Limitations

Acknowledging the limitations of non-parametric methods, including computational intensity, potential overfitting, and challenges in interpretability, the document emphasizes the importance of choosing models based on specific requirements and understanding their inherent limitations.

Overall Model Comparison and Selection

The document compares the performance of various regression algorithms across different polynomial degrees. It identifies trends, strengths, and weaknesses, assisting in making informed decisions about model selection based on the nature of the data and desired outcomes.

In summary, the document provides a holistic understanding of regression algorithms, feature engineering, model evaluation metrics, and the practical considerations involved in choosing and assessing models for different degrees of complexity.