Salary Prediction System: A Comparative Analysis to Determine the Best ML Model for Predicting Job Salary

By Jennifer Grosz

Abstract: *A prediction is statement about what will happen or might happen in the future. Future events are unknown and, in many cases, confirmed, exact data about the future is impossible to obtain. This makes predictions useful tools for the determination of and preparation for probable future outcomes. In this paper, observations containing historical person-level data and salary information will be used as the basis for a comparative analysis between six machine learning models' salary prediction capabilities. Machine Learning models discussed in this paper include Multiple Linear Regression, Polynomial Regression, K-Nearest Regressor, Random Forest Regressor, Ridge Regression and Lasso Regression algorithms.*

Introduction

Problem Statement: The purpose of this analysis is to determine the best Machine Learning algorithm for producing the most accurate prediction of a person's prospective salary based on existing data. Use cases for these models include predicting the salary for a business' newly hired employee or predicting the salary for a business professional job posting online where the anticipated salary information is not provided; it can also be used as an algorithm to predict your own salary. This paper achieves the goal of identifying the best salary prediction algorithm by using person-level data between 2016 and 2020 extracted from the Integrated Public Use Microdata Series - Current Population Survey (IPUMS CPA). The prediction algorithms used for this analysis take certain key features as inputs and produce the most accurate salary predictions based on the previous data each model was trained with.

Motivations: The inspiration behind this paper stems from personal frustrations experienced stemming from the lack of information about expected salary figures provided to prospective applicants during the online job search and application process. Additionally, this analysis is being utilized as an opportunity to further explore and apply machine learning algorithms to develop a deeper personal understanding of learned algorithms and the process for evaluating their performance results.

Literature Review: The methods implemented in this analysis were drawn from several sources where similar works of such projection systems have been studied. U. Bansal et all published an article describing their process of implementing Simple Linear and Multiple Linear Regression models for housing and salary predictions. Das, Barik, Mukherjee published an article in January of 2020 discussing their techniques for the implementation of a salary prediction system using Polynomial Regression models. Achrekar, Pawade, Mathias, and Tekwani published an article in March of 2020 discussing a system that was implemented to help people to analyze current job trends and assist applicants with the identification of required job skills depending on prospective employment location and company in the job market. Similarly, Xin & Khalid published an article in 2018 detailing their implementation of methods using Lasso and Ridge regression techniques to predict housing prices. Each of these published works provided insights and inspiration for the implementation and analysis of algorithms discussed in this paper.

Additionally, the inspiration behind the idea of trying different categorical-coding methods used for handling categorical data in this analysis was drawn Brownlee's article about Ordinal and One-Hot Encoding. In this article, the author claims the best method for handling categorical data "...is unknowable." (Brownlee) and a recommendation was provided to test each technique to discover what works best. This suggestion was the foundation for the dummy vs ordinal coding analysis conducted alongside identifying the best ML model for salary prediction performance.

Implementation

Summary of work: The ML models used for this analysis include Multiple Linear Regression, Polynomial Regression, K-Neighbors Regressor, Random Forest Regressor, Ridge Regression, and Lasso Regression algorithms. The metrics used to measure and evaluate performance of the models include R-squared Score, Mean Squared Error, Mean Average Error, Root Mean Squared Error, and Explained

Variance Score. Two separate versions of each model have been implemented to determine the best method for handling categorical features (ordinal coding and dummy coding) for each model. This analysis identifies the best machine learning algorithm for producing the most accurate salary predictions by tuning the models with different parameter input settings to determine the optimal input parameters as well as evaluating each individual model's performance between two different encoding methods used for handling categorical features.

To implement this process, the entire data set will be split (80/20) into 80% training and 20% test data sets. The 80% training data from this initial split is used for tuning and training of the models. Once optimal model parameters and categorical data encoding methods have been identified, a final comparative test will be ran using the 20% test data and the best-performing version of each machine learning algorithm. The resulting test data from the initial split will not be used during the tuning or training process, it is separated and only used only in a final test.

Experiment Setup

Feature Selection: Based on the results generated from a correlation matrix, the decision was made to exclude the feature years_experience from the data set because of its high correlation coefficient (0.97) with age. Eliminating highly correlated features can help reduce overfitting because less redundant data means there is less opportunity for the models to make decisions based on noise. The features used from the data set are region_name, age, sex, education, and Position. The target variable for the models to predict is earnings. (*See Appendix A)*

Tuning: For each model, a performance evaluation has been conducted to determine the optimal input parameters for each algorithm. Tuning of the models included running the Polynomial regression algorithm with ordinal encoding of categorical features with the degree parameter equal to 2, 3, 4, 5, 6, and 7. K-Neighbor Regressor algorithms were ran with the n_neighbors parameter equal to 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 50, 100, 200, and 300. Finally, Random Forest Regressor algorithms were ran with the n_estimator parameter equal to 10, 50, 100, 150, 200, 250, and 300.

Training: Once the optimal input parameters for each model were identified the resulting optimal algorithms were trained using a 10-fold cross validation. After reviewing the results of the cross validation training the resulting 5 best performing versions of each model are selected for the final test.

Testing: The final test is conducted using the 20% test data that was originally separated from the complete data set. This test identifies which machine learning algorithm provides the most accurate salary prediction by using test data for which they have not been trained with.

Results Presentation

Tuning Results: Respective to ordinal and dummy encoding of categorical features, the Polynomial Regression models performed best with the parameter degrees = 7 and degrees = 2, the K-Neighbor Regressor models performed best when n_neighbors = 8 and n_neighbors = 10, and the Random Forest Regressor models performed best when n_estimators = 250 both methods of encoding categorical features. These input parameter settings resulted in the best performance from each machine learning model. (*See Appendix B)*

Training Results: The table below shows the training results for each model.

Trained Model	Data Used	Parameter	Average R-Squared	Average MSE	Average MAE	Average RMSE	Explained Variance
10-fold CV Multiple Linear Regression	Number	None	0.18	2247.14	33.13	47.39	0.18
10-fold CV Multiple Linear Regression	Dummy	None	0.45	1509.76	24.9	38.84	0.45
10-fold CV Polynomial Regression	Number	Degree = 7	0.42	1598.61	25.47	39.96	0.42
10-fold CV Polynomial Regression	Dummy	Degree = 2	0.47	1459.58	23.83	38.19	0.47
10-fold CV K-Neighbor Regressor	Number	N_neighbors = 8	0.41	1618.53	25.02	40.21	0.41
10-fold CV K-Neighbor Regressor	Dummy	N_neighbors = 10	0.45	1515.79	24.07	38.91	0.45
10-fold CV Random Forest Regressor	Number	N_estimators = 250	0.53	1304.29	18.58	36.1	0.53
10-fold CV Random Forest Regressor	Dummy	N_estimators = 250	0.52	1322.47	18.45	36.35	0.52
10-fold CV Ridge Regresson	Dummy	None	0.47	1463.1	25	38.25	0.47
10-fold CV Lasso Regresson	Dummy	None	0.47	1462.99	24.99	38.25	0.47

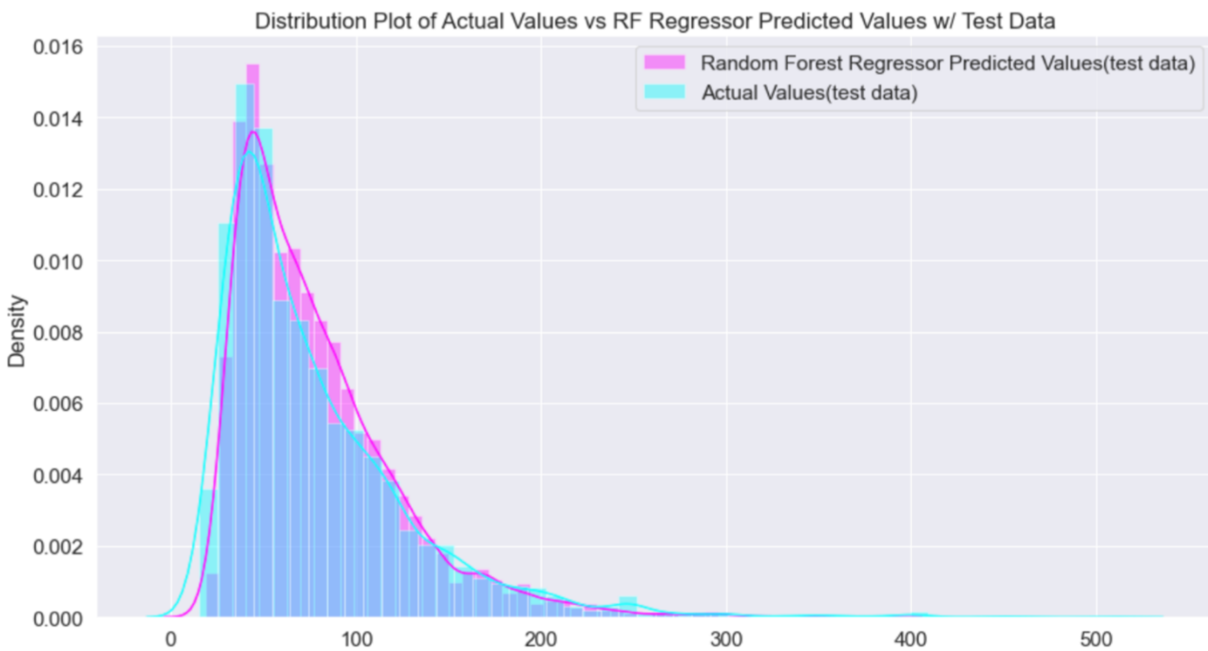
All models, except for the Random Forest Regressor, perform best when the dummy encoded categorical features data set is used. (*See Appendix C)*

Results Discussion

Final Test Results: The table below shows the results for each top-performing version of the models, using test data which they have not been trained on. *(See Appendix D)*

ML Model	R-Squared	MSE	MAE	RMSE	Explained Variance
Multiple Linear Regression	0.44	1551.43	24.98	39.39	0.44
Polynomial Regression	0.46	1489.02	23.95	38.59	0.46
K-Neighbor Regressor	0.42	1595.38	24.22	39.94	0.42
Random Forest Regressor	0.51	1357.11	18.61	36.84	0.51
Ridge Regression	0.44	1552.04	25.01	39.4	0.44
Lasso Regresson	0.44	1552.01	25.01	39.4	0.44

Plotted below is the density distribution of Actual (blue) and Random Forest predicted (pink) earnings values from the final test.



Conclusion: A Random Forest Regressor algorithm using the ordinal encoding of categorical features performed the best across all models and all performance metrics. Since the earnings variable is represented in thousands of USD, the performance results from the Random Forest Regression are to be interpreted as follows:

- The Mean Squared Error of this model is $1,357,110
- The Mean Absolute Error of this model is $18,610
- The Root Mean Squared Error of this model is $36,840
- 51% of the variation in outcome can be explained by the variation in the independent variables

Resources:

Brownlee, Jason. “Ordinal and One-Hot Encodings for Categorical Data”. June 12, 2020. https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/

U Bansal et al. “Empirical analysis of regression techniques by house price and salary prediction” 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1022 012110. https://iopscience.iop.org/article/10.1088/1757-899X/1022/1/012110/pdf

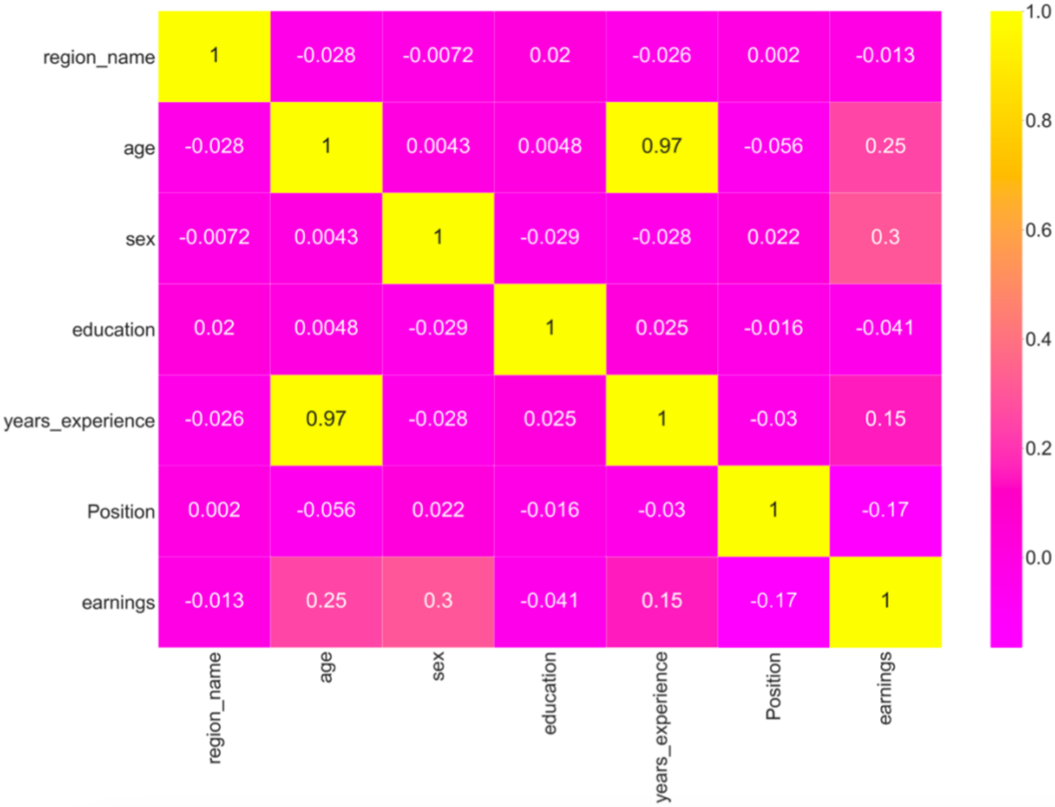
Das, Sayan & Barik, Rupashri & Mukherjee, Ayush. “Salary Prediction Using Regression Techniques”. January 2020. SSRN Electronic Journal. 10.2139/ssrn.3526707. https://www.researchgate.net/publication/339055809_Salary_Prediction_Using_Regression_Techniques

Achrekar D., Pawade M., Mathias G, Tekwani B. “Job Portal Analysis and Salary Prediction System”. March 2020. International Research Journal of Engineering and Technology (IRJET). Volume: 07 Issue: 03. https://www.irjet.net/archives/V7/i3/IRJET-V7I31054.pdf

XIN, Seng Jia & Khalid, Kamil. “Modelling House Price Using Ridge Regression and Lasso Regression” 2018. International Journal of Engineering & Technology, 7 (4.30) 498-501. https://www.sciencepubco.com/index.php/ijet/article/view/22378

Appendix A: Feature Selection

Figure A is a visualization of the Feature Selection's Correlation Matrix that was used for the determination of eliminating years_work from the features included for model analysis described in this paper.



(Figure A)

Appendix B: Tuning Results

Figure B.1 is a table of performance metrics resulting from the Polynomial Regression Model tuning using Ordinal Encoding of Categorical Features

Model	Data Used	Degree	MSE	R-squared
Polynomial Regression	number-coded	2	2077.54	0.244909
Polynomial Regression	number-coded	3	1988.12	0.27741
Polynomial Regression	number-coded	4	1874.63	0.318659
Polynomial Regression	number-coded	5	1716.24	0.376225
Polynomial Regression	number-coded	6	1629.29	0.407828
Polynomial Regression	number-coded	7	1627.43	0.408505

(Figure B.1)

Figure B.2 is a table of performance metrics resulting from the K-Neighbors Regressor Model tuning using Ordinal Encoding of Categorical Features

Model	Data Used	N_Neighbors	MSE	R-Squared
K-Neighbors Regressor	number-coded	1	2148.91	0.223209
K-Neighbors Regressor	number-coded	2	1844.51	0.333246
K-Neighbors Regressor	number-coded	3	1725.75	0.376174
K-Neighbors Regressor	number-coded	4	1699.73	0.385581
K-Neighbors Regressor	number-coded	5	1656.56	0.401186
K-Neighbors Regressor	number-coded	6	1648.66	0.404042
K-Neighbors Regressor	number-coded	7	1625.13	0.412547
K-Neighbors Regressor	number-coded	8	1620.72	0.41414
K-Neighbors Regressor	number-coded	9	1642.7	0.406194
K-Neighbors Regressor	number-coded	10	1644.03	0.405713
K-Neighbors Regressor	number-coded	20	1701.1	0.385086
K-Neighbors Regressor	number-coded	30	1731.87	0.373962
K-Neighbors Regressor	number-coded	50	1792.96	0.35188
K-Neighbors Regressor	number-coded	100	1853.43	0.33002
K-Neighbors Regressor	number-coded	200	1957.75	0.292309
K-Neighbors Regressor	number-coded	300	2023.21	0.268649

(Figure B.2)

Figure B.3 is a table of performance metrics resulting from the K-Neighbors Regressor Model tuning using Dummy Encoding of Categorical Features

Model	Data Used	N_Neighbors	MSE	R-Squared
K-Neighbors Regressor	dummy-coded	1	2032.78	0.265189
K-Neighbors Regressor	dummy-coded	2	1676.17	0.394096
K-Neighbors Regressor	dummy-coded	3	1603.08	0.420518
K-Neighbors Regressor	dummy-coded	4	1576.59	0.430091
K-Neighbors Regressor	dummy-coded	5	1544.86	0.441564
K-Neighbors Regressor	dummy-coded	6	1531.04	0.446559
K-Neighbors Regressor	dummy-coded	7	1512.32	0.453324
K-Neighbors Regressor	dummy-coded	8	1507.02	0.45524
K-Neighbors Regressor	dummy-coded	9	1518.15	0.451219
K-Neighbors Regressor	dummy-coded	10	1499.67	0.457898
K-Neighbors Regressor	dummy-coded	20	1501.02	0.457408
K-Neighbors Regressor	dummy-coded	30	1523.86	0.449154
K-Neighbors Regressor	dummy-coded	50	1536.67	0.444522
K-Neighbors Regressor	dummy-coded	100	1610.69	0.417765
K-Neighbors Regressor	dummy-coded	200	1711.21	0.381431
K-Neighbors Regressor	dummy-coded	300	1791.81	0.352293

(Figure B.3)

Figure B.4 is a table of performance metrics resulting from the Random Forest Regressor Model tuning using Ordinal Encoding of Categorical Features

Model	Data Used	N_Estimators	MSE	R-Squared
Random Forest Regression	number-coded	10	1536.16	0.444707
Random Forest Regression	number-coded	50	1427.48	0.483995
Random Forest Regression	number-coded	100	1412.69	0.48934
Random Forest Regression	number-coded	150	1405.25	0.492031
Random Forest Regression	number-coded	200	1401.86	0.493253
Random Forest Regression	number-coded	250	1399.29	0.494182
Random Forest Regression	number-coded	300	1400.24	0.493841

(Figure B.4)

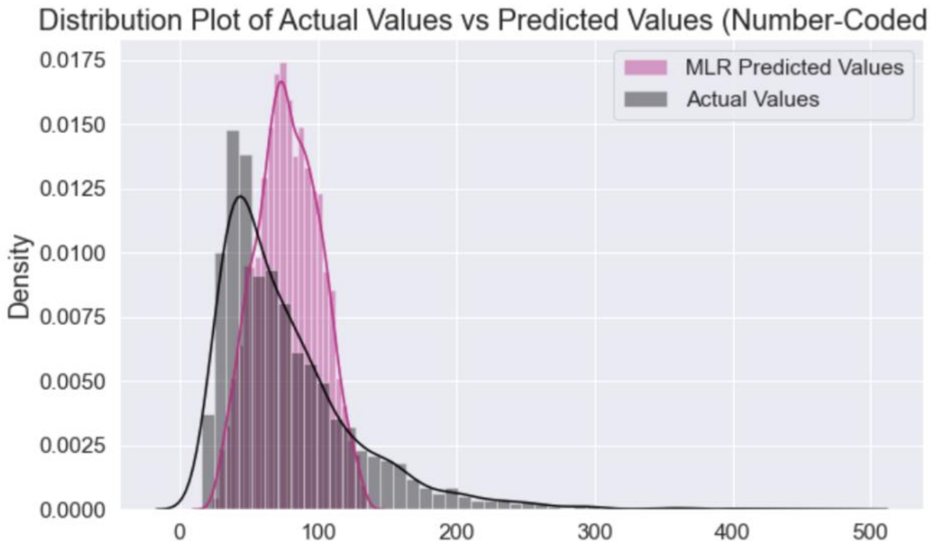
Figure B.5 is a table of performance metrics resulting from the Random Forest Regressor Model tuning using Dummy Encoding of Categorical Features

Model	Data Used	N_Estimators	MSE	R-Squared
Random Forest Regression	dummy-coded	10	1509.9	0.454201
Random Forest Regression	dummy-coded	50	1412.98	0.489234
Random Forest Regression	dummy-coded	100	1405.41	0.491972
Random Forest Regression	dummy-coded	150	1395.39	0.495592
Random Forest Regression	dummy-coded	200	1392.74	0.496552
Random Forest Regression	dummy-coded	250	1388.55	0.498067
Random Forest Regression	dummy-coded	300	1390.19	0.497473

(Figure B.5)

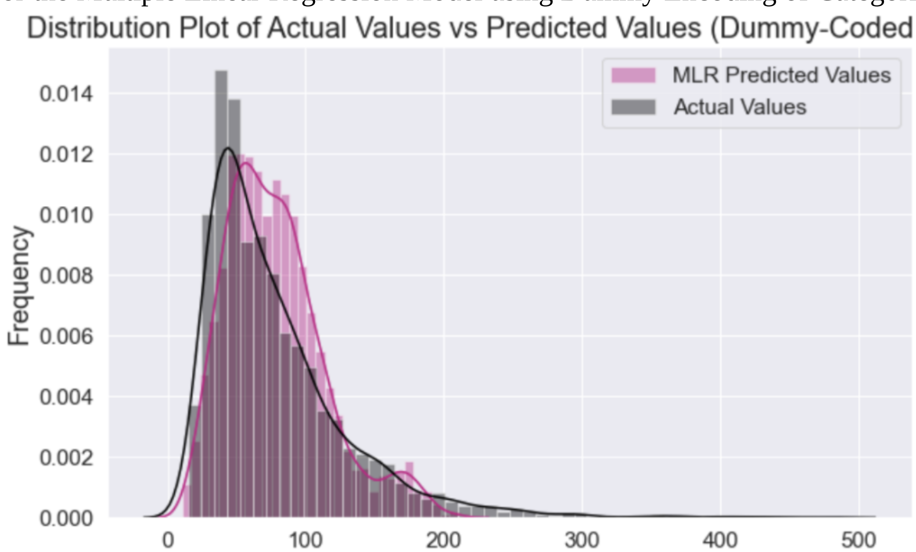
Appendix C: 10-Fold Cross-Validation Training Results

Figure C.1.A is the resulting distribution plot after a 10-fold cross validation training was implemented for the Multiple Linear Regression Model using Ordinal Encoding of Categorical Features



(Figure C.1.A)

Figure C.1.B is the resulting distribution plot after a 10-fold cross validation training was implemented for the Multiple Linear Regression Model using Dummy Encoding of Categorical Features



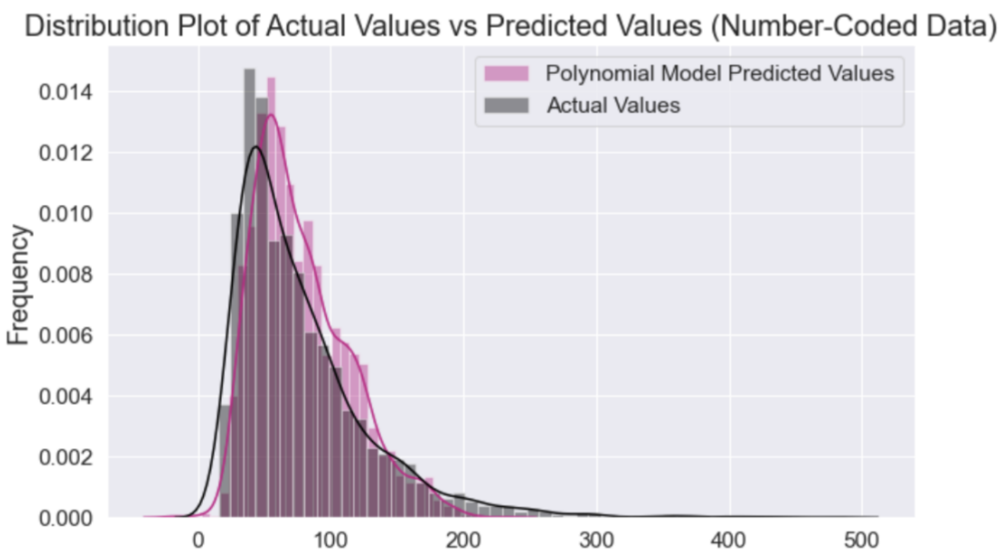
(Figure C.1.B)

Figure C.1.C is a complete table of performance metrics resulting from the Multiple Linear Regression Models' 10-fold Cross Validation Training

Trained Model	Data Used	Average R-Squared	Average MSE	Average MAE	Average RMSE	Explained Variance
10-fold CV Multiple Linear Regression	Number	0.18	2247.14	33.13	47.39	0.18
10-fold CV Multiple Linear Regression	Dummy	0.45	1509.76	24.9	38.84	0.45

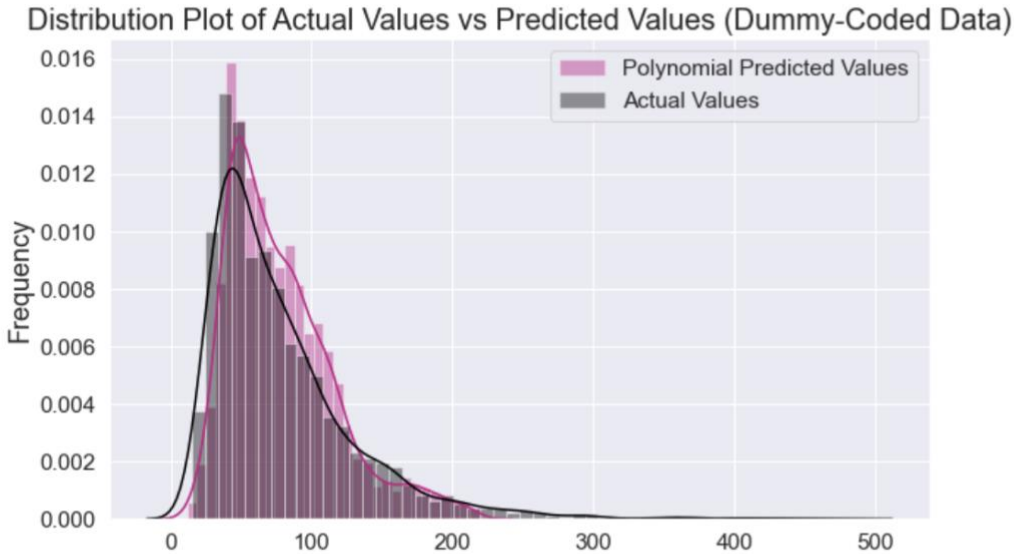
(Figure C.1.C)

Figure C.2.A is the resulting distribution plot after a 10-fold cross validation training was implemented for the Polynomial Regression Model using Ordinal Encoding of Categorical Features



(Figure C.2.A)

Figure C.2.B is the resulting distribution plot after a 10-fold cross validation training was implemented for the Polynomial Regression Model using Dummy Encoding of Categorical Features



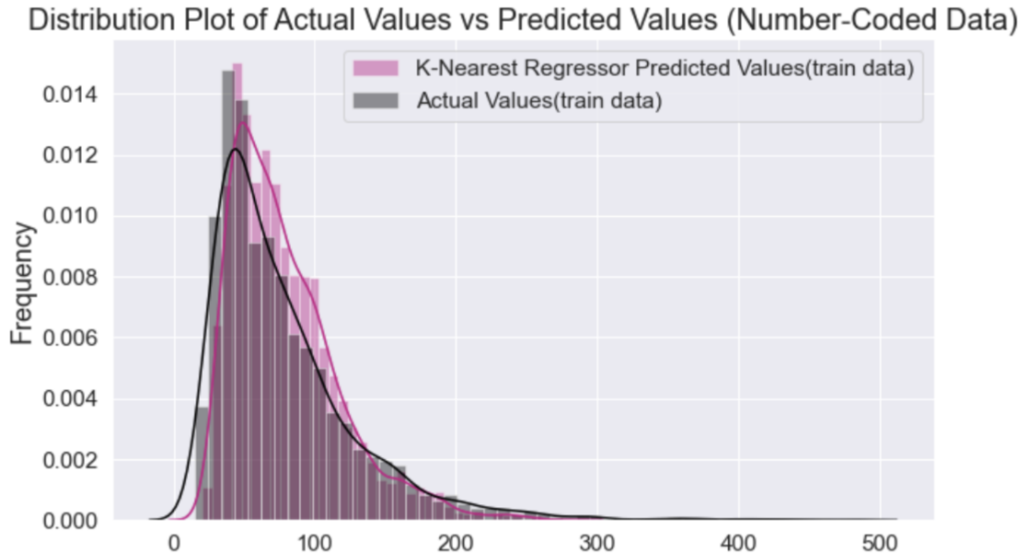
(Figure C.2.B)

Figure C.2.C is a complete table of performance metrics resulting from the Polynomial Regression Models' 10-fold Cross Validation Training

Trained Model	Data Used	Degrees	Average R-Squared	Average MSE	Average MAE	Average RMSE	Explained Variance
10-fold CV Polynomial Regression	Number	7	0.42	1598.61	25.47	39.96	0.42
10-fold CV Polynomial Regression	Dummy	2	0.47	1459.58	23.83	38.19	0.47

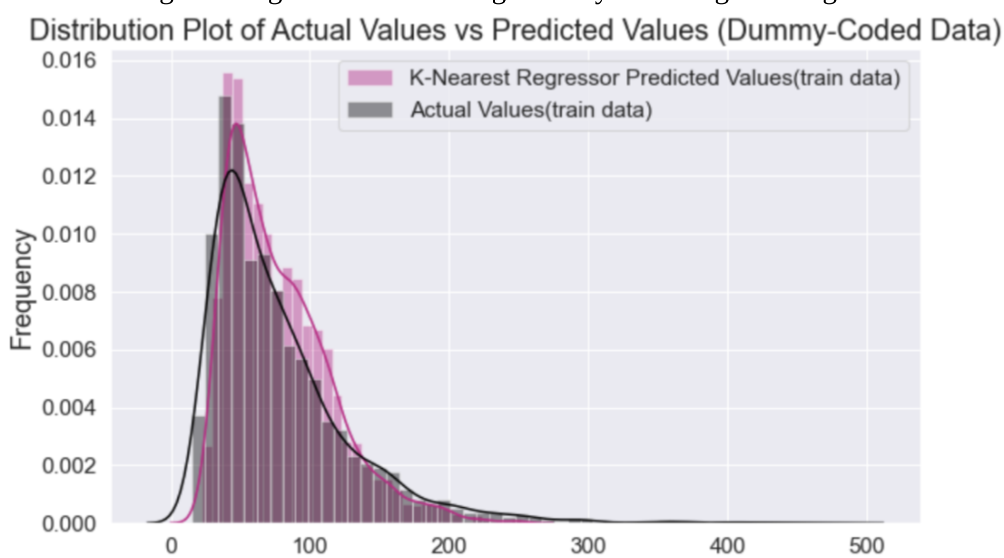
(Figure C.2.C)

Figure C.3.A is the resulting distribution plot after a 10-fold cross validation training was implemented for the K-Neighbor Regressor Model using Ordinal Encoding of Categorical Features



(Figure C.3.A)

Figure C.3.B is the resulting distribution plot after a 10-fold cross validation training was implemented for the K-Neighbor Regressor Model using Dummy Encoding of Categorical Features



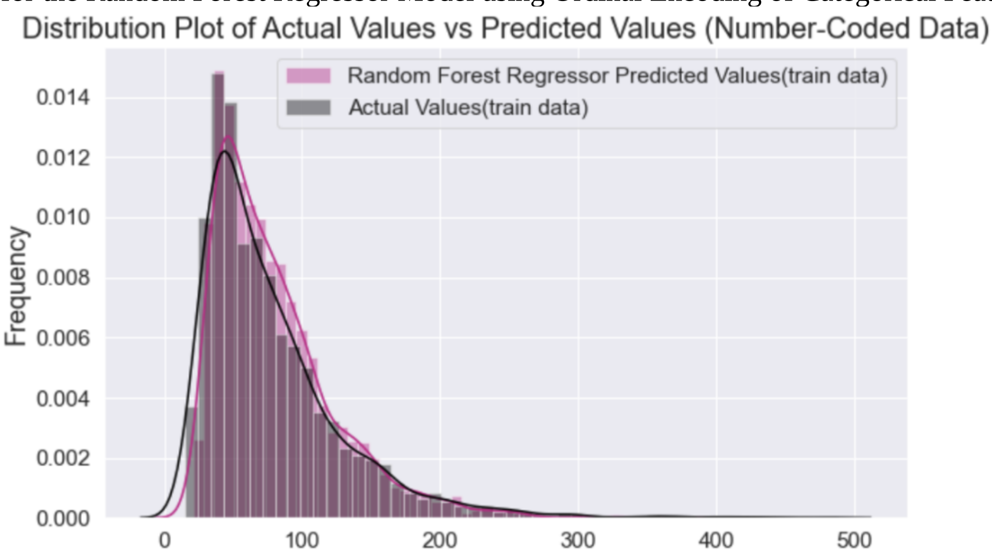
(Figure C.3.B)

Figure C.3.C is a complete table of performance metrics resulting from the K-Neighbor Regressor Models' 10-fold Cross Validation Training

Trained Model	Data Used	neighbors	Average R-Squared	Average MSE	Average MAE	Average RMSE	Explained Variance
10-fold CV K-Neighbor Regressor	Number	8	0.41	1618.53	25.02	40.21	0.41
10-fold CV K-Neighbor Regressor	Dummy	10	0.45	1515.79	24.07	38.91	0.45

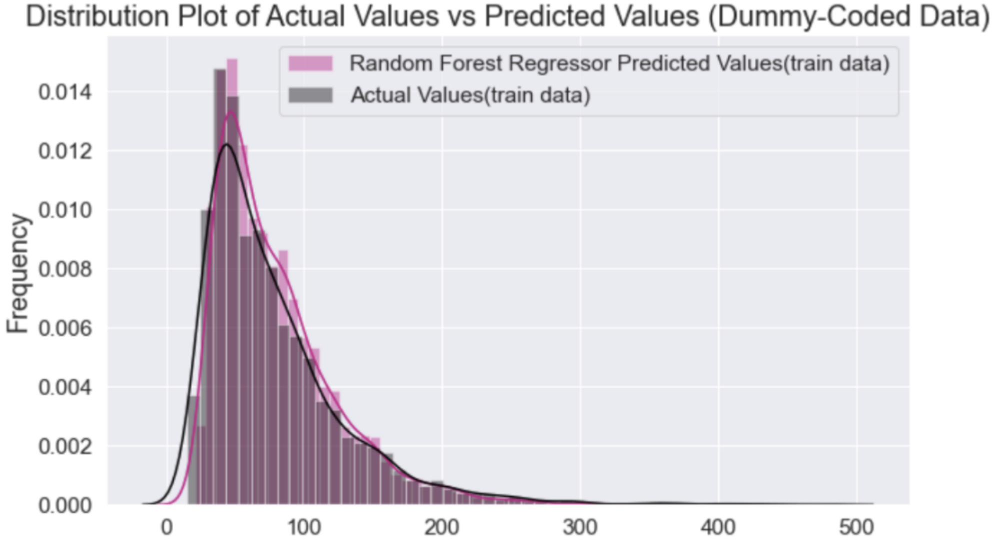
(Figure C.3.C)

Figure C.4.A is the resulting distribution plot after a 10-fold cross validation training was implemented for the Random Forest Regressor Model using Ordinal Encoding of Categorical Features



(Figure C.4.A)

Figure C.4.B is the resulting distribution plot after a 10-fold cross validation training was implemented for the Random Forest Regressor Model using Dummy Encoding of Categorical Features



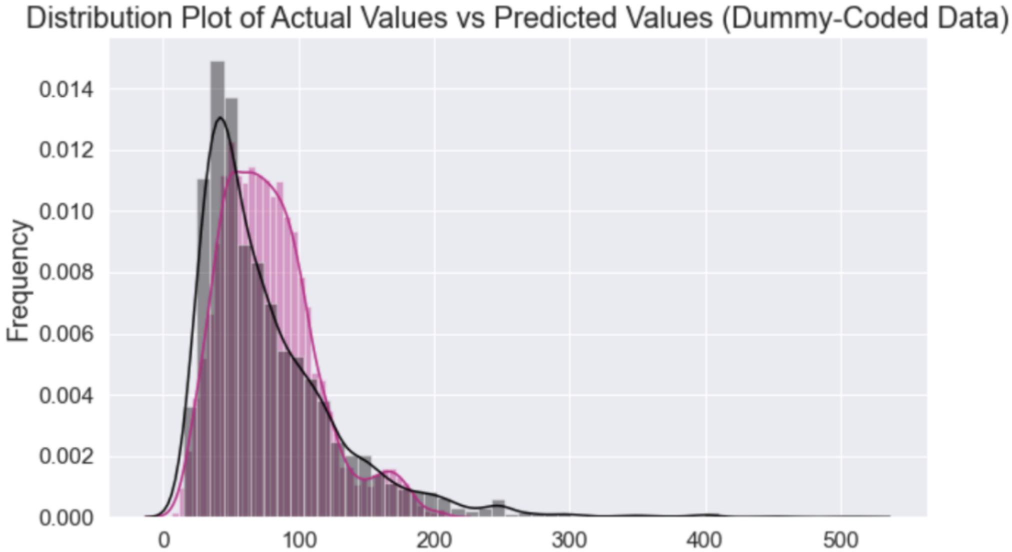
(Figure C.4.B)

Figure C.4.C is a complete table of performance metrics resulting from the Random Forest Regressor Models' 10-fold Cross Validation Training

Trained Model	Data Used	Estimators	Average R-Squared	Average MSE	Average MAE	Average RMSE	Explained Variance
10-fold CV Random Forest Regressor	Number	250	0.53	1304.29	18.58	36.1	0.53
10-fold CV Random Forest Regressor	Dummy	250	0.52	1322.47	18.45	36.35	0.52

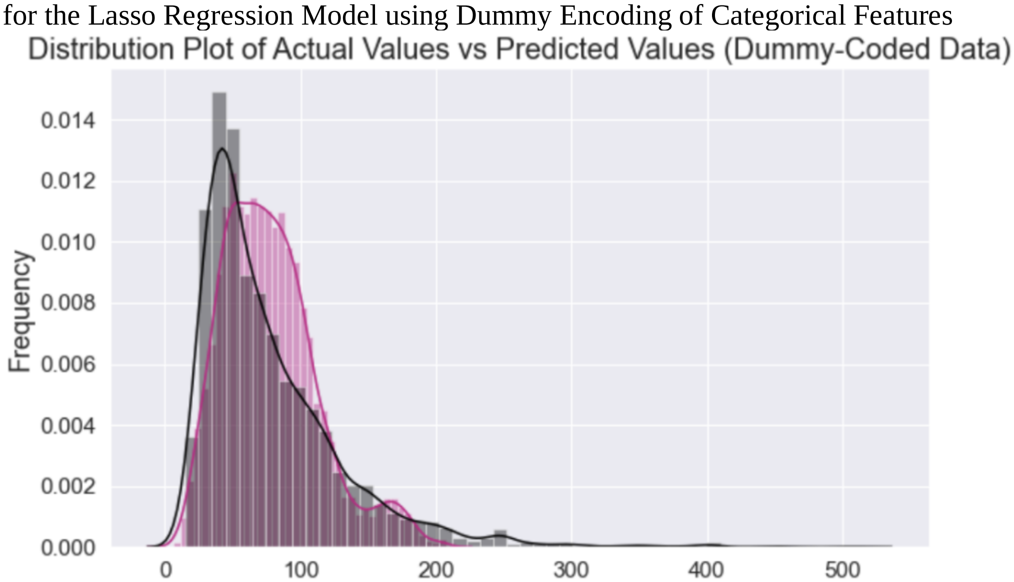
(Figure C.4.C)

Figure C.5.A is the resulting distribution plot after a 10-fold cross validation training was implemented for the Ridge Regression Model using Dummy Encoding of Categorical Features



(Figure C.5.A)

Figure C.5.B is the resulting distribution plot after a 10-fold cross validation training was implemented for the Lasso Regression Model using Dummy Encoding of Categorical Features



(Figure C.5.B)

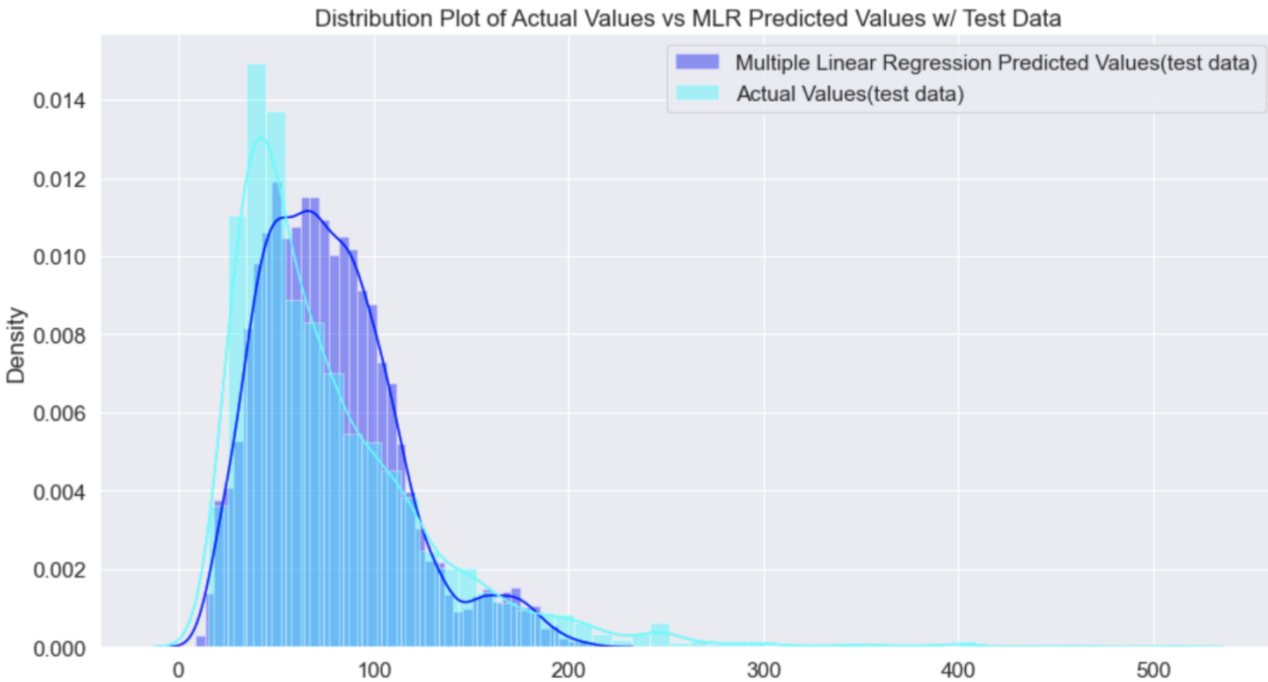
Figure C.4.C is a complete table of performance metrics resulting from the Ridge and Lasso Regression Models' 10-fold Cross Validation Training

Trained Model	Data Used	Average R-Squared	Average MSE	Average MAE	Average RMSE	Explained Variance
10-fold CV Ridge Regression	Number	0.47	1463.1	25	38.25	0.47
10-fold CV Lasso Regression	Dummy	0.47	1462.99	24.99	38.25	0.47

(Figure C.5.C)

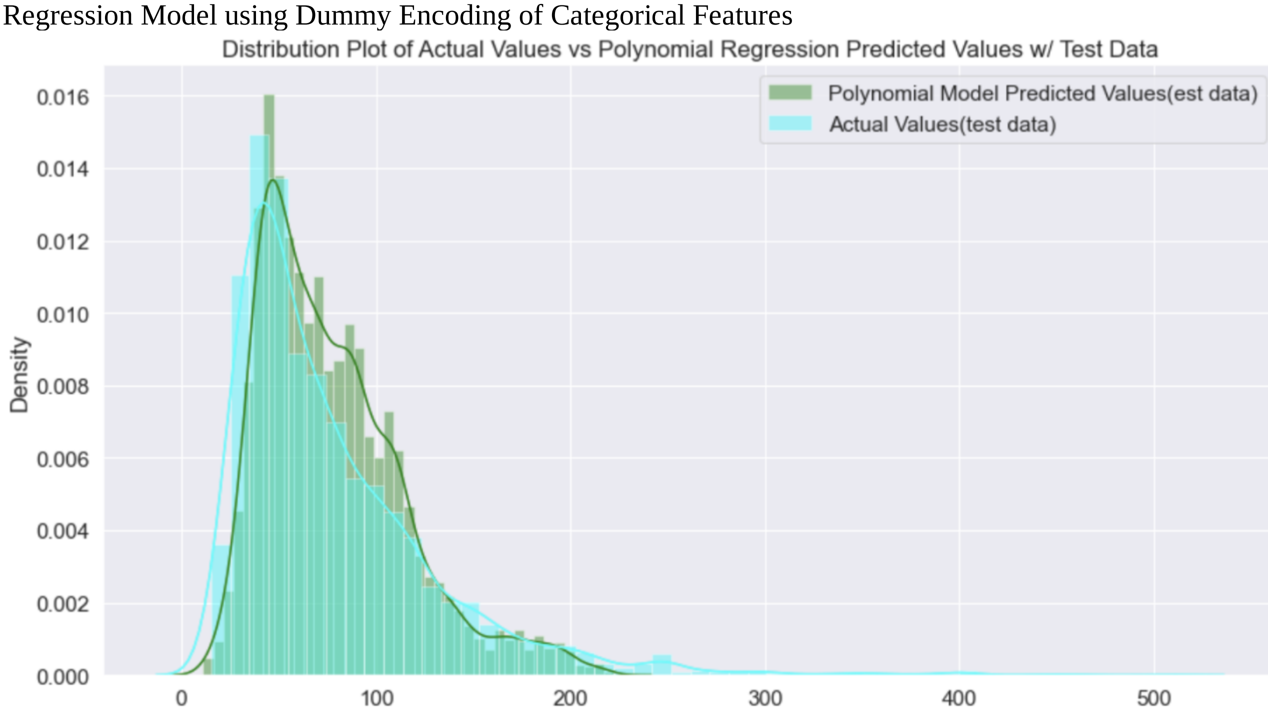
Appendix D: Final Test Results from each model

Figure D.1 is the resulting distribution plot after the final test was implemented for the Multiple Linear Regression Model using Dummy Encoding of Categorical Features



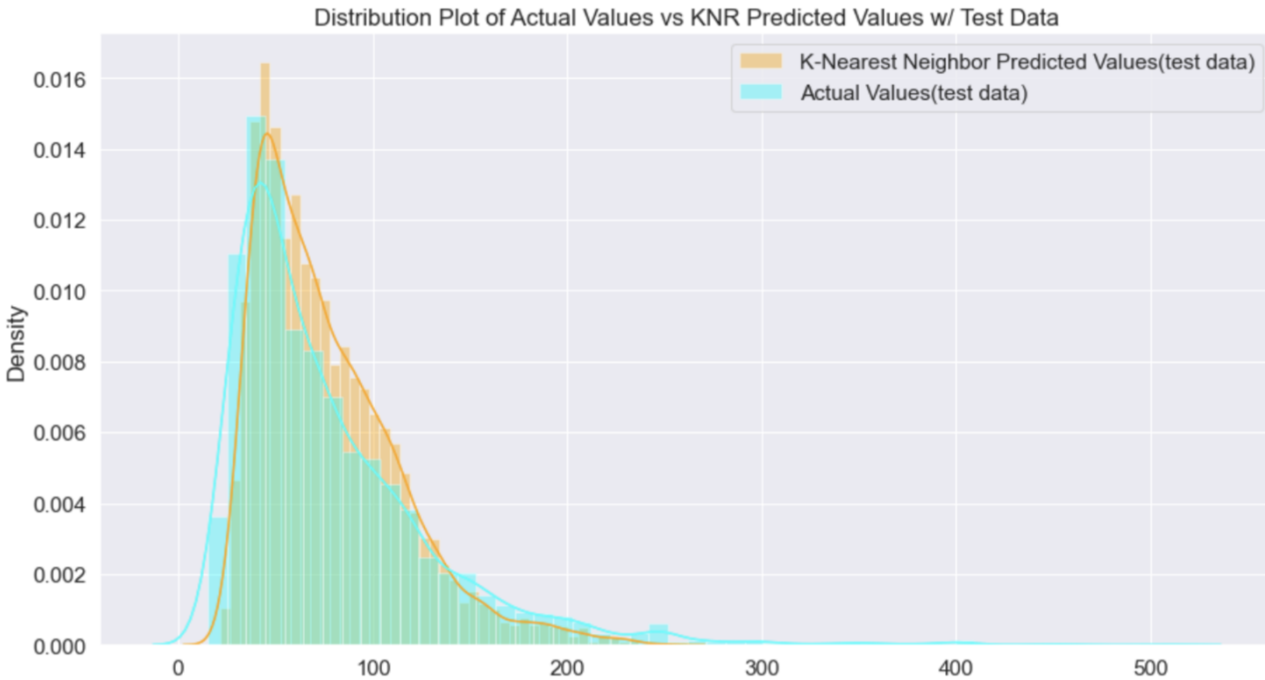
(Figure D.1)

Figure D.2 is the resulting distribution plot after the final test was implemented for the Polynomial Regression Model using Dummy Encoding of Categorical Features



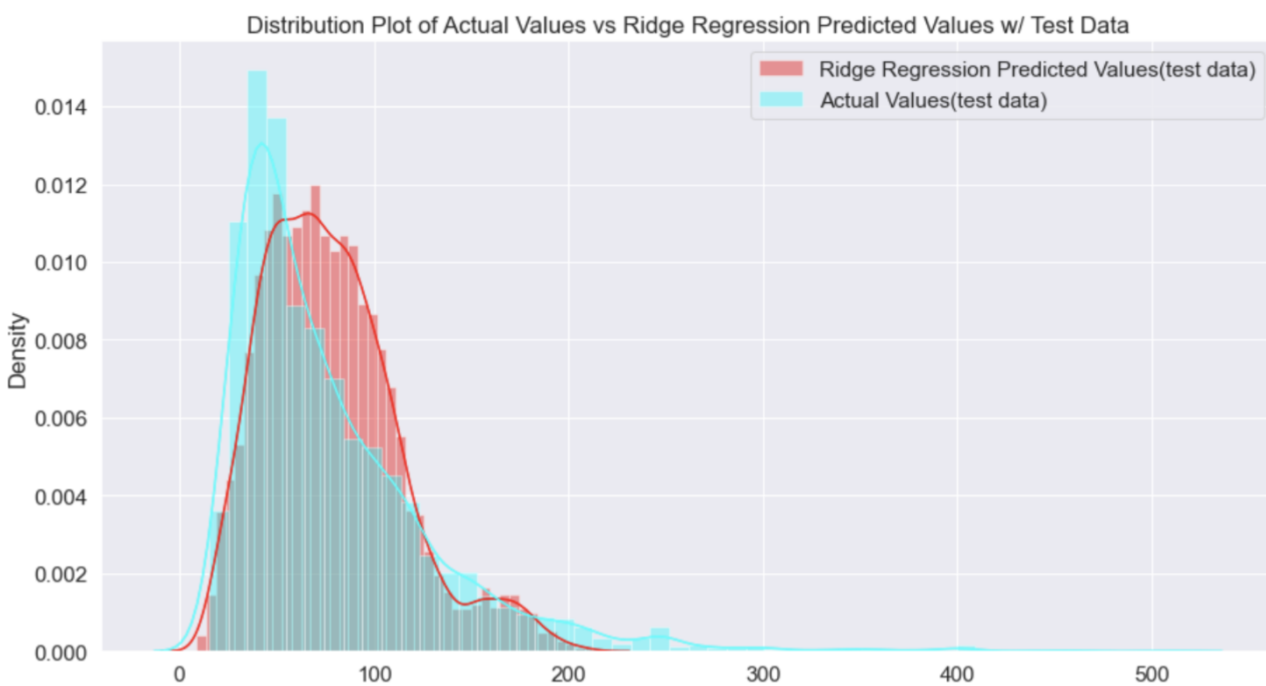
(*Figure D.2)*

Figure D.3 is the resulting distribution plot after the final test was implemented for the K-Neighbor Regressor Model using Dummy Encoding of Categorical Features



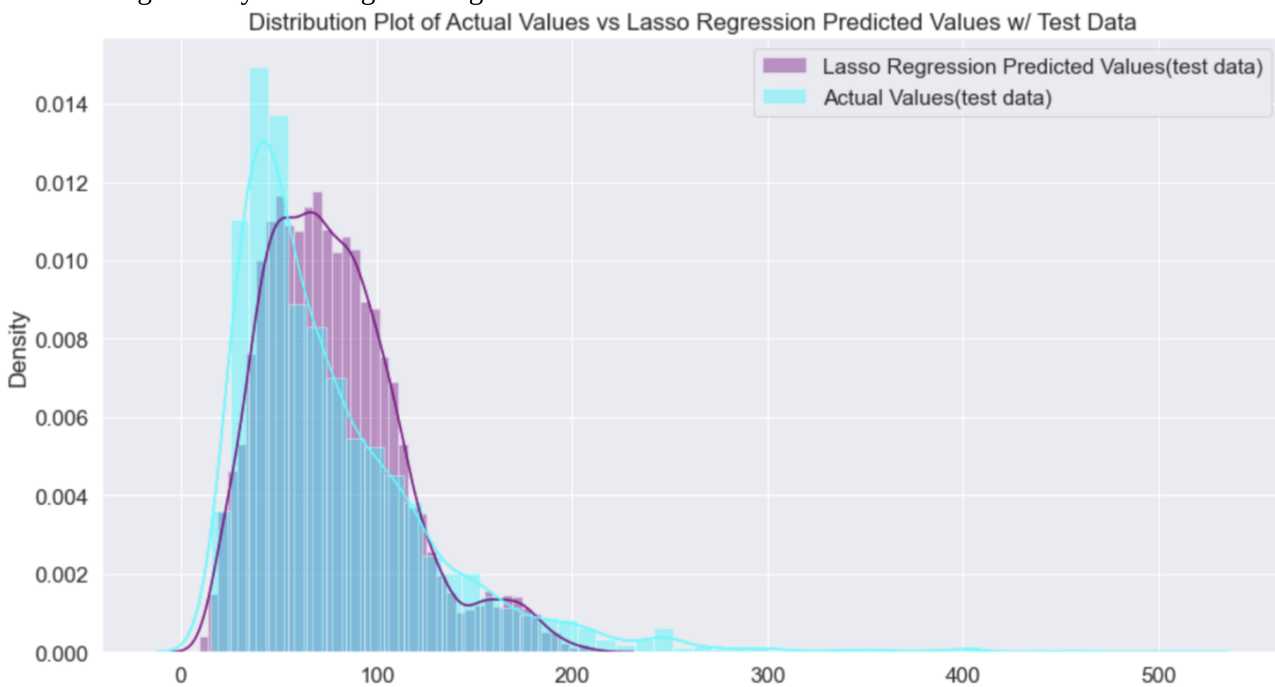
(Figure D.3)

Figure D.4 is the resulting distribution plot after the final test was implemented for the Ridge Regression Model using Dummy Encoding of Categorical Features



(Figure D.4)

Figure D.5 is the resulting distribution plot after the final test was implemented for the Lasso Regression Model using Dummy Encoding of Categorical Features



(Figure D.5)