IE 7280 Statistical Methods in Engineering

Final Project

Northeastern University

Aneesha Subramanian
Ankita Ajit Doddihal
Ashok Thiruvengadam
Rishika Chhabrani

Executive Summary:

Regression analysis is a statistical method that helps us to understand how two or more variables are related to each other. In other words, it helps us to find the relationship between a dependent variable (the variable we want to predict) and one or more independent variables (the variables that we think might be affecting the dependent variable).

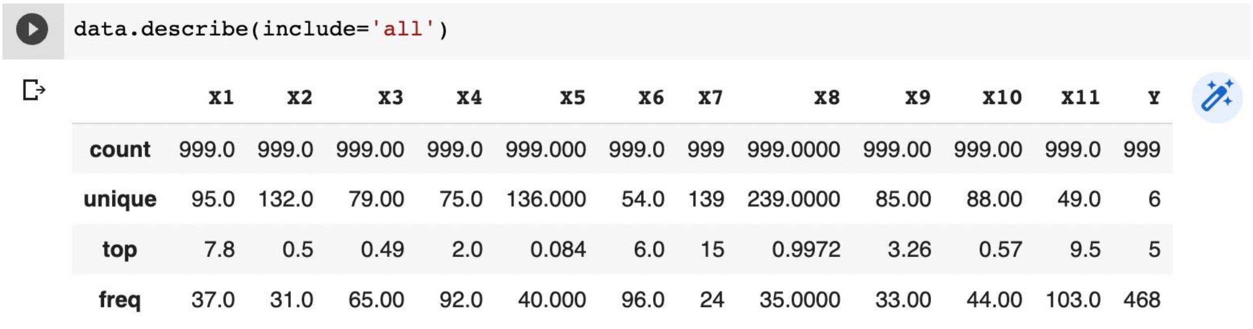
The basic idea behind regression analysis is to create a model that can predict the value of the dependent variable based on the values of the independent variables. This is done by analyzing the data to find a pattern or relationship between the variables. It is not a good idea to manually engineer the data here because we lack the knowledge necessary to understand the features' nature.
Our training dataset consists of 999 records, where Y is the response variable and the regressors are X1,...,X11.

As we develop this model, we will have the chance to use statistical analysis to examine the relative relevance of the various features, the amount of variance they can explain in the data, and, if applicable, any hidden relationships between the features. Additionally, statistical metrics like mean squared error, sum of squared errors, R squared, adjusted R squared will give us some reference to guide our model's testing.

Starting with exploratory data analysis and ending with the model evaluation on test data, each member of the group contributed equally to the project.

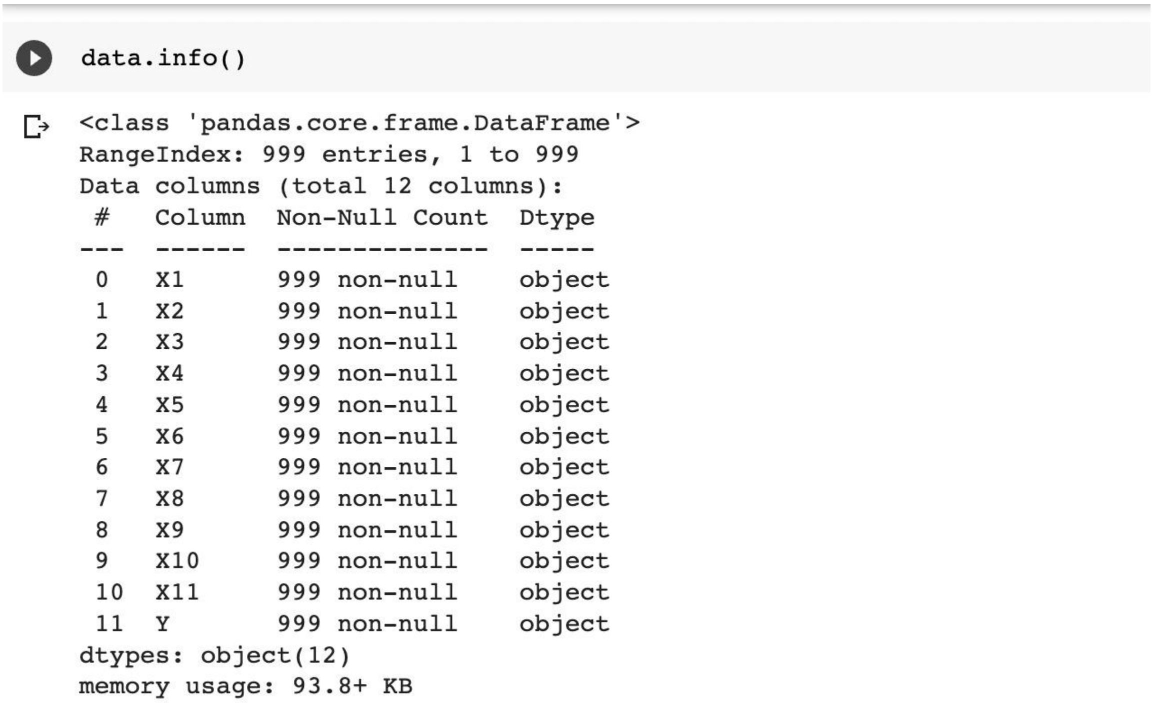
Data Inspection:

Preliminary analysis of the data shows that there are no missing values and that all of the independent features and the dependent variable are numerical.



```
data.describe(include='all')
```

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	Y
count	999.0	999.0	999.00	999.0	999.000	999.0	999	999.0000	999.00	999.00	999.0	999
unique	95.0	132.0	79.00	75.0	136.000	54.0	139	239.0000	85.00	88.00	49.0	6
top	7.8	0.5	0.49	2.0	0.084	6.0	15	0.9972	3.26	0.57	9.5	5
freq	37.0	31.0	65.00	92.0	40.000	96.0	24	35.0000	33.00	44.00	103.0	468

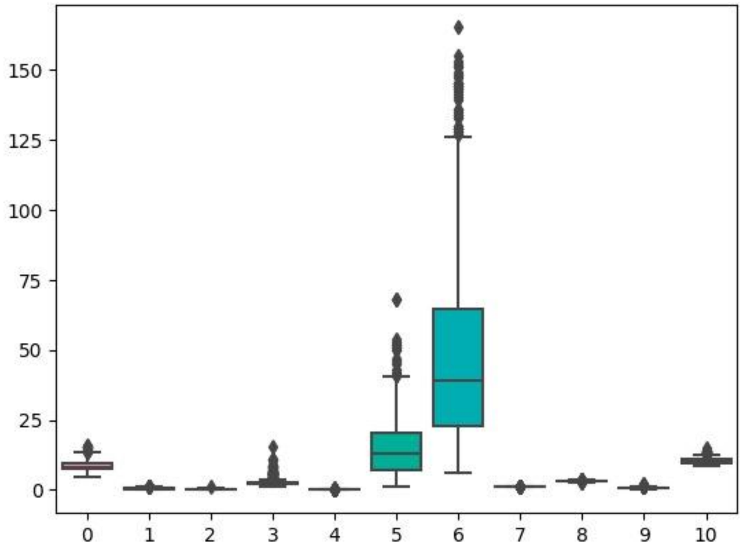


```
data.info()
```

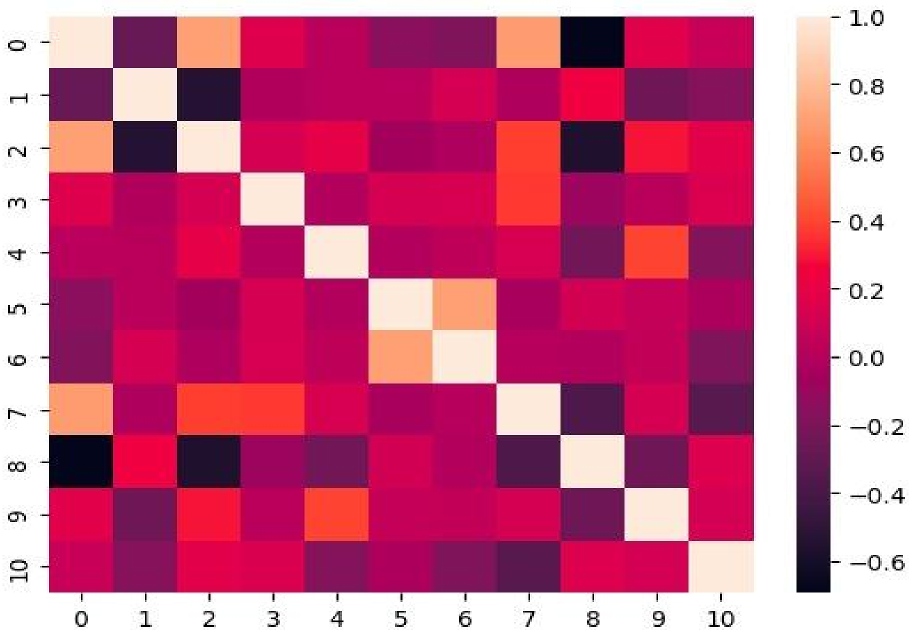
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 999 entries, 1 to 999
Data columns (total 12 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   X1      999 non-null    object
 1   X2      999 non-null    object
 2   X3      999 non-null    object
 3   X4      999 non-null    object
 4   X5      999 non-null    object
 5   X6      999 non-null    object
 6   X7      999 non-null    object
 7   X8      999 non-null    object
 8   X9      999 non-null    object
 9   X10     999 non-null    object
 10  X11     999 non-null    object
 11  Y       999 non-null    object
dtypes: object(12)
memory usage: 93.8+ KB
```

Outlier Detection and Correlation Check:

We used a boxplot to determine whether our data contains any outliers. The linear correlations between all the pairs of independent features can be discovered using heat maps.



Outliers can significantly affect the regression model by skewing the results and reducing the accuracy of the model. As we can see, the data set observes outliers for variables X5 and X6. It is also important to note that the removal of outliers should be done with caution, as removing too many or important observations can lead to an incorrect model fit and poor predictions. We assume that the outlier won't significantly impact the model's performance hence we are moving forward.



From the plot, we can infer the following

1) Highly correlated feature pairs: (X6, X7), (X1, X8), (X1, X3)
2) Weakly correlated feature pairs: (X1, X9), (X2, X3), (X3, X9)

Data Modelling:

In our models, we have analyzed the following metrics:

1. Mean Squared Error (MSE): The average squared difference between the expected and actual values are represented by the mean squared error (MSE) measure. Better model performance is indicated by lower MSE values.
2. Sum of Squared Errors (SSE): The total of the squared deviations between the predicted and actual values is this metric. SSE should be reduced because it indicates the entire inaccuracy in the model.
3. R-squared (R2): This statistic shows the percentage of the dependent variable's variance that can be accounted for by the independent variable (s). Better model fit and data explainability are indicated by higher R2 values.
4. Adjusted R-squared: This statistic is comparable to R2 but accounts for the number of predictors included in the model. The model is penalized for adding pointless predictors by the adjusted R2, which could lead to a higher R2. Better model performance is indicated by higher adjusted R2 values.

In our model selection - we have decided to evaluate using R2, Adjusted R2, and MSE.
The lower the MSE, the better the model's performance, and a value of 0 indicates a perfect fit to the data (although this is often not achievable in practice).
R-squared is the proportion of the variance and ranges from 0 to 1, where 0 means the model does not explain any variation in the dependent variable and 1 means the model perfectly explains all the variation in the dependent variable. However, R-squared alone can be misleading when the number of independent variables in the model is large. In this case, the adjusted R-squared is used. In summary, R-squared tells you how well the model fits the data, while adjusted R-squared tells you how well the model fits the data while taking into account the number of independent variables, and Mean square error (MSE) is the average of the square of the errors.

a) Polynomial regression

Polynomial Regression is a type of regression analysis used to model the relationship between a dependent variable and one or more independent variables by fitting a polynomial equation to the data. The degree of the polynomial equation is chosen based on the complexity of the data and the desired level of accuracy. For example, a quadratic equation (degree 2) would fit a parabolic curve to the data, while a cubic equation (degree 3) would fit a more complex curve.
It is often used when the relationship between the dependent and independent variables is not linear and cannot be adequately modeled by linear regression.

Evaluation Metrics for Model 1 - Polynomial Regression

Mean Squared Error: 0.28134867534589914
Sum of Squared Errors: 281.06732667055326
R-squared: 0.7186513246541009
Adjusted R-squared: 0.7155157264486247

b) Random forest regressor:
Random Forest Regressor is a machine learning algorithm that uses an ensemble of decision trees to make predictions on a continuous target variable. The algorithm is based on the concept of bagging, where multiple models are trained on different subsets of the data to reduce the variance of the predictions.

Evaluation Metrics for Model 2 - Random Forest Regressor

Mean Squared Error: 0.045092992992993
SSE: 45.047900000000006
R-squared: 0.9292101898947827
Adjusted R-squared: 0.9284212457092129

c) XGBoost:

XGBoost is a popular machine-learning algorithm that uses an ensemble of decision trees to make predictions on a target variable. The algorithm is based on the gradient boosting framework, which involves iterative fitting weak models to the residuals of the previous models in the ensemble. It is robust to outliers and can handle high-dimensional data, as well as missing and non-linear relationships between the input features and the target variable. XGBoost also provides insights into the importance of individual input features in making predictions, which can be useful for feature selection and interpretation.

Evaluation Metrics for Model 3 - Gradient Boost

Mean Squared Error: 0.01602389180335345
SSE: 16.007867911550097
R-squared: 0.9839600521928356
Adjusted R-squared: 0.9837812888434143

Model Comparison and Selection:

Polynomial Regression, as indicated by the relatively high mean squared error (MSE) of 0.281 and the lower R-squared and adjusted R-squared values compared to the other two models. These metrics suggest that the model may be fitting the training data too closely and may not generalize well to new data. In contrast, Models 2 and 3 - Random Forest Regressor and Gradient Boost, respectively, appear to have better generalization performance but Model 3 - Gradient Boost may show signs of overfitting on the testing dataset.

Based on the evaluation metrics, Model 2 - Random Forest regressor appears to be the best model to use for testing that will not overfit. This is because it has a low mean squared error (MSE) of 0.045 and a high R-squared value of 0.928, indicating a better fit for the data. The adjusted R-squared values for all three models are relatively close. Although Model 3 still has the highest value of 0.983, indicating that it is the best fit for the data while also being the most parsimonious, we feel it may overfit on the final dataset due to the R-square value being very close to 1. Therefore, We would recommend using Model 2 - Random Forest regressor

Model Evaluation on Test data

For the final model evaluation, we have trained our models with complete data. The models were then evaluated on the test set, consisting of 599 observations. After evaluating our candidate model (Gradient Boost) using the Test data set, we achieved the following R2, MSE values :

R2 - 0.99995   MSE - 3.33609

Based on the above results, although the Gradient Boost model is performing well, there are a few signs of overfitting as R2 is 0.99995 and MSE has increased to 3.33609 indicating that the model was performing well on the training data but a little poorly on the testing data which can indicate that the model is memorizing the training data instead of learning the underlying patterns.

Therefore, we suggest **Random Forest Regressor model as a better model based on the following values :**

R2 - 0.92945   MSE - 0.04707

Although Gradient Boost can be improved using feature engineering and other complex algorithms, we propose to use Random Forest Regressor as our final model.

Steps to improve our best model - Random Forest Regressor

1. **Feature engineering:** Feature engineering can help create better features from existing data, making the model more accurate. For example, creating interaction terms or adding polynomial features.
2. **Hyperparameter tuning:** Random forests have many hyperparameters, including the number of trees, maximum depth of each tree, and the number of features considered at each split. Tuning these hyperparameters can lead to improved performance.
3. **Ensemble learning:** Ensemble learning can help to improve the accuracy of the model by combining multiple random forest models. For example, you could train multiple random forests with different hyperparameters, and then average their predictions.
4. **Regularization:** Random forests are prone to overfitting, and regularization techniques such as limiting the depth of the trees or reducing the number of features considered at each split can prevent overfitting.
5. **Cross-validation:** Cross-validation can help to evaluate the performance of the model on unseen data and prevent overfitting. You can use techniques like k-fold cross-validation to assess the accuracy of the model.