Intro to AI

Models Used:
1)SVR
2)Polynomial regression
3)Random forest regression
4)XGB

1)SVR:

Support vector regression is a machine learning model based on the support vector machine model , SVR seeks to find the line of best fit between data points , but unlike normal regression it allows some tolerance of errors , the amount it allows is called epsilon(e).

Data preprocessing:

In our SVR we first used “data.describe()” to take a look at the statistics of our data , we found that columns X9 and X2 had many missing values , so we filled X9 with the mode since its categorical and X2 with the mean since its numerical. We then checked for outliers in the data but did not find any.
X3 also had some issues where “Low fat” was written in different ways and “regular” as well so we made them all the same.
We considered X1 to be the ID of the product so we dropped it , since we think it had no importance.

After filling X2 with the mean we plotted the correlations of the numerical data and found that X2 and X8 have very weak correlations with our target Y so we dropped them.

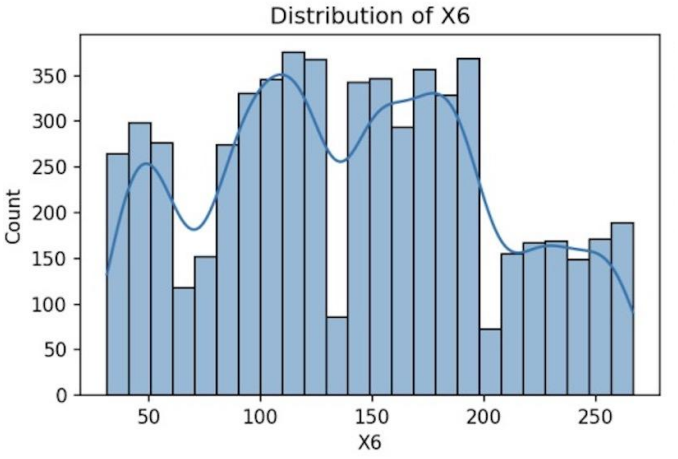
One hot encoding:

The columns X5,X7 and X11 were all non-ordinal categorical columns to we used one hot encoding on them to avoid any misinterpretation in the model if we used label encoding.
After one hot encoding X5 we found that all the categories of X5 don’t correlate much with the target Y so we dropped all the values of X5.

In X7 after the one hot encoding it was clear that X7_4 and X7_0 had some correlation with Y so we added those columns to our data frame and removed the rest.
After one hot encoding X11 we also found that X11_0 , X11_1 , and X11_3 had good correlations with Y so they were used them in our model and dropped the rest.

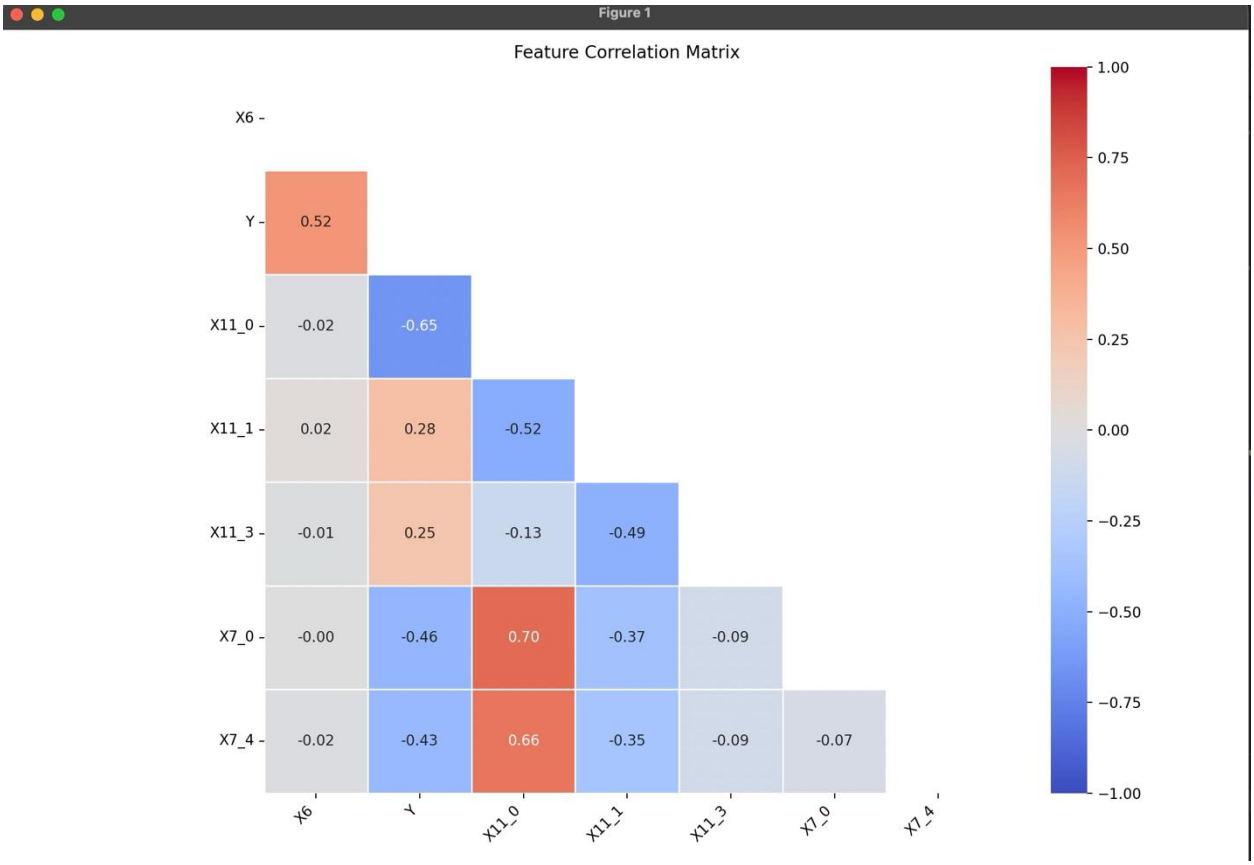
Label encoding:
The columns X3,X10,X9 were all ordinal category columns since X9 had low , medium high and X10 had tiers that we think were ordered from 1 to 3.
We used label encoding for all these columns but at the end found their correlations with Y to be very low , therefore they were dropped from our model.

Standardization and Normalization:
The only numerical column left we had was X6 so we plotted X6.



After looking at the graph of X6 and doing some research we found that the best for X6 would be to use standardization , so we used the Standard Scaler on X6.

Final Preprocesing:
After one hot encoding and label encoding we plotted the correlations on a heat map to take a look at the correlations between features.



We found that X11_0 had very strong correlations with X7_0 and X7_4 which could cause issues in our model , so we decided on removing X7_4 since it had the weaker correlation with the target Y and we kept X7_0.

We performed the same preprocessing on the test Data then began with the model training.

Model training:
When training our model we used a grid search to find the best parameters for the SVR model , we made the grid search choose the best epsilon , kernel , C , and gamma.
The epsilon is the tolerance for error , kernel was choosing between linear ,poly or rbf.

C is the regularization parameter , that helps us avoid overfitting on the training data and generalizing and gamma is a parameter for the rbf kernel.

We performed the grid search using 5 k-fold cross validations , meaning the data was split into 4 sets that were used for training and 1 to test and then evaluated based on mean absolute error, after each evaluation it switched the test set to another set.
The grid search gave us {'C': 50, 'epsilon': 0.005, 'gamma': 0.1, 'kernel': 'rbf'} as our parameters.
After that the model was finally trained and we passed in the test data to get our predictions.

Score: 0.369(Highest)

This score being our highest score , we then using this same model tried to remove X11_0 and keep X7_4 and X7_0 since X11_0 had a strong correlation with both of them and some correlation with the other X11's.
Using the same grid search as the previous SVR model we got these results for our parameters {'C': 50, 'epsilon': 0.01, 'gamma': 0.1} , we used rbf since it gave us the best results.
After that the model was finally trained and we passed in the test data to get our predictions.

Score: 0.370(Second highest)

2)Polynomial regression:

Polynomial regression extends linear regression by modeling the relationship between the independent variable x and the dependent variable y as a polynomial of degree nnn. Instead of fitting a straight line, it uses higher-degree terms (x2,x3,...,xnx^2, x^3, , x^nx2,x3,...,xn) to capture non-linear patterns in the data. The model learns coefficients for these polynomial terms using techniques like gradient descent to minimize the error between predictions and actual values. By balancing the complexity of the polynomial, it avoids underfitting or overfitting, making it ideal for data with curvilinear relationships.

Data preprocssing:
Columns X1, X3, X5, X8, X10, were dropped since they showed weak correlations with the Y, then the missing values in X2 and X9 were filled by the mean in X2 since it was numerical while X9 was filled with the mode since it was categorical. We used label encoding for X9 since it was categorical, and it was ordinal while one-hot encoding was applied on X7 and X11 since they were also categorical but did not show ordinality.

Model training:
After preprocessing the data, the model was trained on the data using a degree equal to 3 since it showed the most sensible predictions.

Standardization and Normalization:
We chose after looking at the graph of X6 to use Normalization.

Score: 0.400

3)Random forest:
Random Forest is an ensemble machine learning model that operates by constructing multiple decision trees during training and then averaging their predictions to improve accuracy and control overfitting. Unlike individual decision trees, Random Forest reduces variance and is more robust to noise in the data.

Data preprocssing:
We applied the same preprocessing as our SVR model removing , X3 , X5,X8,X9,X10 and X2. For this model we noticed that X1 consisted of 3 letters and two numbers “FDA15” the first two letters in many cases where either FD , DR or NC so we thought that they may represent some category, we tried splitting them into 3 columns but the correlations of the 3 columns were very weak with Y so we dropped X1 again.

Model training:

After pre-processing the data we used grid search again to find the best parameters for the random forest model , the parameters for the random forest model were max depth for the trees , n_estimators , min_samples_split, and min_sample_leaf.
n_estimators determines the number of decision trees in the random forest, min_samples_split determined the min number of data a node can carry to be able to split.
min_sample_leaf min number of nodes in a leaf for it to be considered a leaf.
The parameters the grid search gave us were {'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 200}
After trainng the model we predicted the test data.

Score: 0.385

4)XGBoost:

The XGBoost (Extreme Gradient Boosting) model is a powerful machine learning algorithm based on gradient boosting. It works by building a sequence of decision trees, where each tree learns to correct the errors of the previous ones. Instead of training all trees at once, XGBoost trains them iteratively, optimizing an objective function using gradient descent. It includes advanced features like regularization to prevent overfitting, efficient handling of missing data, and parallel computation for faster training. XGBoost is widely used because it is fast, scalable, and delivers high accuracy on structured/tabular data.

Data preprocessing:
This model used the same preprocessing that was used on the poly regression

Model training:
After preprocessing the data, the xgboost model was fitted and trained on the data that used parameters that were (n_estimators=200, learning_rate=0.3, max_depth=6, random_state=42)

Score: 0.481