# Executive Summary– Final Project for DDS 8555.

By Samuel Mbah Nde – DDS 8555 – Assignment 8

## Project Overview.

This document is an executive summary highlighting the work I did in my final assignment for Predictive Analysis class. This week, I participated in three Kaggle competitions, one time series analysis competition, [Store Sales Time Series Forecasting](https://www.kaggle.com/competitions/store-sales-time-series-forecasting/overview) (Alexis Cook & Holbrook, 2021), one regression analysis - [House Prices - Advanced Regression Technques](https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques)(Montoya & Canary, 2016)and one classification - [San Francisco Crime Classification](https://kaggle.com/competitions/sf-crime) (Kan, 2015). This document highlights the work done in each of the competition, from data exploration through model, feature engineering through model selection, training, validation, and submission of predictions. The code used for this analysis and all assets are in my GitHub repository at https://github.com/ndesamuelmbah/DDS-8555/tree/main/AssignmentFiles/Week8 (Nde, 2025). This document is divided into three sections, one for each of the analysis mentioned above.

**Section 1: Store Sales Time Series Forecasting.**

The objective of this competition was to forecast store sales of various products families of the Ecuadorian grocery retailer Corporación Favorita.

Data Exploration.

I started of by enriching the training and test dataset with the me provided metadata to obtain an enriched training dataset containing 54 stores each carrying more 33 product families. Observing that it would be difficult to explore each of those store-product pairs, I grouped the data by products and inspected the sales. In Figure 1 below, I plot the total sales by products for the product families Automotive, School and Office Supplies and Seafood. This means an effective model will have to be different for each product.

Figure 1  
Total Sales by product for the years 2016 and 2017.  
A graph of a graph

AI-generated content may be incorrect.

**Note:** *Observe the presence of a weekly seasonality in all the products. School and Office supplies have peaks in May and September corresponding to school year.*

In figure 2 below, I plot the total sales by store. The chart shows that each store has a clear seasonality of 7 meaning weekly sales follow a specific pattern for each store. However, the pattern is not the same for every store. For example, the peak sales in stores 1 and 2 tend to correspond to the lowest sales for store 3. This tells us that the optimal model will be one that treats sales of each store differently.

To assert the periodicity of the seasonality of total sales in the dataset. I plotted a periodogram shown in Figure 3 below.

**Figure 2:**Time series of Total sales by store number for year 2017 displaying a clear seasonality of 7.A graph of a graph

AI-generated content may be incorrect.

**Figure 3**  
Periodogram of Total Sales from training dataset.  
A graph showing a number of sales data

AI-generated content may be incorrect.   
**Note:** *The clear peak at 7 indicates strong weekly season. The minor peaks at 14 and 21 indicate harmonics. The minor peak at 30 suggests monthly seasonality.*

To identify the trend in the data, I plotted a decomposition for periods of 7 and 30 days.

**Figure 4:**

Decomposition plot of total sales from training dataset for seasonality values of 7 and 30.  
A screenshot of a graph

AI-generated content may be incorrect. A screenshot of a graph

AI-generated content may be incorrect.

**Note:** **Left:** *Decomposition plot with seasonality of 30 over the entire dataset. This plot makes it clear to see the trend in the data.* **Right:** *Decomposition plot with seasonality of 7 for 2017 data annotated with national holidays (red vertical lines). Note that sales on national holidays generally fall. I did not include holidays in left plot because the red lines will be too cluttered together.*

To educate the process of model parameters, I plotted an auto correlation plot and partial auto correlation plot of the dataset. As shown in Figure 5 below.

Figure 5:

Auto correlation and Partial Auto Correlation plot of total sales.  
A graph with blue dots and lines

AI-generated content may be incorrect. A graph with blue dots and numbers

AI-generated content may be incorrect.

**Observations from Auto correlation plots.**

* **Strong Positive Autocorrelation at Lag 1:** High correlation at lag 1 indicates that sales on one day are highly correlated with sales on the previous day. This suggests that the sales data is **highly persistent or exhibits strong temporal dependence**.
* **Seasonal Pattern at Lag 7:** Peaks at lag 7, 14, 21, and multiples of 7 indicate a **weekly seasonality.** This suggests that sales patterns repeat every 7 days, consistent with weekly shopping behavior.
* **Gradual Decay of Correlation:** A gradual decline in autocorrelation indicates that the influence of past observations diminishes over time but still retains some correlation over a 30–40-day period.
* **Seasonal Peaks Beyond 30 Lags:** There are secondary peaks after lag 30, suggesting the possibility of monthly or longer seasonal cycles in addition to the weekly pattern.

**Autocorrelation (ACF) Plot Interpretation.**

* **High Partial Autocorrelation at Lag 1:** A large spike at lag 1 suggests a **strong AR(1) process**, meaning that daily sales are strongly influenced by the previous day’s sales.
* **Significant Correlation at Lag 7:** The notable spike at lag 7 indicates that a **7-day lag (weekly lag)** has a significant direct impact on sales, supporting the hypothesis of **weekly seasonality.**
* **Rapid Decay After Lag 7:** After the initial lags (1 and 7), partial autocorrelation quickly drops, suggesting that additional lags beyond these points contribute less to direct correlations.

Implementing an ETS Model.

The learnings from the above guided me to implementing a loop that trains a model with parameters generated from decomposition plot for combinations of store numbers and product families. The loop is shown in Figure 6 below.

It is important to note a data normalization strategy which I used while training the model. Since multiplicative seasons and trend are not supported with time series that may contain zeroes in the target, and our training data did indeed contain zeroes, I normalized the dataset by adding 5 to the original sales before training the model. When I get my forecasts, I also subtract 5 from the forecast. This walk around allowed me to use exponential smoothing model on my data.

I followed a similar approach for getting forecasts using my **ARIMA** model with the parameters already discussed above. Figure 7 below shows the predictions from both my ETS model and my ARIMA model scoped to the training data in 2017 plus the forecasts.

**Figure 6:**

*Code snippet showing how I loop through each store number product pair, generating dynamic error, trend, and periodic parameters for making a time series.*  
**A screenshot of a computer screen

AI-generated content may be incorrect.**

**Note:** *For each pair of store number and product family, I infer the most appropriate trend, period, and error from the train data. Then use those to fit a model and save the results to a csv file. This approach ensures that a model is fit that captures the specifics of the data.*

**Figure 7**

Forecasted sales from ETS and ARIMA models.

A graph with blue and green lines

AI-generated content may be incorrect.

A graph showing a graph of sales

AI-generated content may be incorrect.

**Note:** The ARIMA model captures the trend much better than the ETS model but both models are conservative, rarely predicting values bigger than the observed ones, suggesting that there might be need to work on damping.

**Figure 8:**

**Proof of Submission  
A screenshot of a computer

AI-generated content may be incorrect.**

**Section 2: House Prices Advanced Regression Techniques.**

The goal of this competition is to use home attributes such as the square footage, the number of bedrooms and neighborhood location to determine the price of the home.

**Data Exploration.**

I started off with data exploration. I observed the presence of many missing values on some of the fields in the dataset, some of which are missing at random and other missing because they are not valid. So, I designed a process of filling those missing values in three steps.

1. Remove all fields with more than 80% missing values missing because including them in the model will only increase model complexity.
2. Fill the missing values of each field that is missing because the value does not make sense to be populated with an appropriate default value. For example, when categorical variable fireplace quality is missing because the house does not have a fireplace, I use a default value of `NotA`. Similarly, when a home is missing the Garage Year Built and the total garage area is 0 sqft, I fill the missing year with 1700. Using this approach, I was able to fill 99% of the missing values.
3. After completing the above approach, I noticed that only one variable had a missing value which was missing electrical circuit breaker value. I filled this with the modal circuit model of the houses in its neighborhood. I did this because homes in each neighborhood tend to have similar configurations.

**Feature engineering**

I created a correlation plot of the numerical features in the dataset and observed that there was high correlation between some features for example, high correlation between Garage Arae and the number of Cars that can fit into the Garage. This observation drove me to create new features by combining related features. The code snippet in Figure 9 below shows how I merged.

**Figure 9:**  
*Creating composite features by combining related features into 1 feature.*  
A screenshot of a computer code

AI-generated content may be incorrect.  
**Note:** *The feature `BsmtFinSF` representing the finished square footage is a combination of finished square footage of type 1 and 2. Similar logic was used to create features for `TotalBath` (total number of bathrooms in the home) by summing the number of full baths and half bathrooms with an assumption that 2 half bathrooms make a full bathroom. The final cell converts the qualitative variables representing the quality of some fields into ordinal variable.*

The choice of numbers to use in the conversion of the categorical variables to an ordinary scale was guided by the fact that an attribute which is in excellent condition (Ex) is much better than one that is in poor condition denoted `Po` (Gnat, 2021). The quality of 0 was used for missing values which were encoded as the category `NotA`. Figure 10 below shows a correlation heat map of the numeric features in the dataset.

**Figure10:**  
*Heatmap of numerical features in the training dataset after feature engineering.*  
A screen shot of a chart

AI-generated content may be incorrect.  
**Note:** *Note the presence of a few highly correlated fields for example `YearSinceGarageWasBuilt` has a -0.92 correlation with `Ordinal\_GarageQual` suggesting that older garages (higher `YearSinceGarageWasBuilt`) have a lower quality which is to be expected because of wear from use over time.*

To avoid violating the condition of no multi collinearity between features for linear regression model, I used a combination of Variance inflation factor (VIF) and correlation with sale price to determine which of the highly correlated features to remove from the dataset.

**Figure 11:**  
*Approach for removing highly correlated features using VIF and correlation with target feature.*A screenshot of a computer program

AI-generated content may be incorrect.  
**Note:** *Variables with a low correlation threshold of 0.075 (in absolute value) with sale price like ['3SsnPorch', 'LowQualFinSF', 'MiscVal', 'MoSold', 'YearSinceSold'] were dropped. Based on a combination of correlation with sale price and VIF, the features ['YearSinceGarageWasBuilt', 'Ordinal\_ExterQual', 'RemodelledAge', 'Ordinal\_BsmtQual', '1stFlrSF'] were removed.*

With feature engineering completed, divided the data into train and test sets with 80% for training, scaled the dataset and fitted linear regression model. I also attempted to train a polynomial regression model with features interacting but could not use polynomial directly because the feature set would be too large for my computer to process. This pushed me to use principal component analysis to extract features that could capture 80% of the variation in the training dataset (Josse & Husson, 2012). Figure 12 below shows a scree plot of the PCA analysis.

**Figure 12:**  
*Scree plot showing the proportion of variance explained by number of Components.*

A comparison of a graph

AI-generated content may be incorrect.  
**Note:** *About 55% of the variation in the dataset is captured by the first 5 principal components. Additional features contribute only a small amount of variation. We need 12 principal components to capture 80% of the variation in the dataset which originally consisted of 26 features.*

With these 12 principal components, I was able to train a model, but the model came with the challenge that it lacked explainability since one cannot attribute a principal component to a specific feature in the dataset. This motivated me to try Lasso regression with feature interaction.

**Figure 13:**  
Feature importance from Lasso regression.  
A graph with blue and white bars

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.  
**Note:** *From the cummulative importance percentage, you see that the first 11 featrues provide 80% of the importance. While this is not exactly the same as capturing variation in the dataset, we can use these features with their interactions to create a polynomial model with interactions.*

By using forward feature selection with lasso, I was able to perform cross validation and get the feature set with the best adjusted R-squared value.  
**Figure 13:**  
*Adjusted R squared by number of features – Forward Feature Selection.*

A graph with a line

AI-generated content may be incorrect.

Using this approach, I was able to identify 6 features for modelling. I added the interactions of these features to create polynomial regression model with feature interactions. The importance scores and the variance inflation factors of the resulting model are shown in figure 14 below.

**Figure 14:**  
*Feature importance and VIF for best performing polynomial model.*

A graph with blue and white text

AI-generated content may be incorrect. A graph of a graph with text

AI-generated content may be incorrect.  
**Note**: ***Left****:* *The most important feature for determining the price of the house is the total square footage. Other important factors are the neighborhood and the overall quality of the house.* ***Right****: All the features in the model have a VIF less than 10 which shows that there is very little multicollinearity between the features.*

**Investigation of Model Assumptions.**

* **Check for multicollinearity:** The Variable Inflation factor for the dataset shows a score of less than 10 for all the variables in the dataset. Any VIF more than 10 is generally considered problematic.
* **Test for Independence:** The Durbin-Watson Test for independence produced a test statistic of 1.93 which is very close to 2 - an indication that there is no auto correlation and the features are independent (King, 1981).
* **Normality of Residuals**
  + **Generally Normal:** The residuals mostly follow the diagonal line, suggesting reasonable normality.
  + **Light Left Tail:** The left tail deviates below the line, indicating a slightly lighter left tail than a perfect normal distribution.
  + **Heavy Right Tail:** The right tail deviates above the line, indicating a slightly heavier right tail.
  + **Outliers Present:** Some points deviate significantly, especially at the tails, suggesting presence of outliers.
  + **Central Normality:** The center of the distribution aligns well with the line, indicating good normality in that region.

**Looking forward and improvements.**

* **Address Tail Deviations:** The plot shows deviations from normality at both tails. Investigate if transformations of the target or features could improve normality.
* **Investigate Outliers:** The plot highlights potential outliers. Examine these data points to determine if they are errors or genuine data points that need special handling.
  + We might need to remove the outliers or
  + Transform the data so that the outliers become less influencing.
* **Consider Non-Linearity:** While the middle portion is good, the tail deviations might indicate uncaptured non-linear relationships. It would be interesting to see what some non-parametric models like Random Forest models can perform on the dataset.
* **Heteroscedasticity is Present:** The residual plot clearly shows heteroscedasticity with the prediction spreading towards the right of the dataset. This may be influenced by the outliers in the dataset. It needs to be investigated and addressed to improve the reliability and accuracy of your model. Some options for addressing this issue include:
  + **Transform the Target Variable:** Common transformations like taking the logarithm or square root of the target variable (SalePrice in your case) can often stabilize the variance.
  + **Weighted Least Squares (WLS):** WLS is a technique that assigns weights to data points based on their variance. This can give more weight to data points with lower variance and less weight to data points with higher variance.
* **Feature Engineering:** Refine or add features to capture the underlying patterns and reduce residual deviations potentially better. This may also help with the Heteroscedasticity issue.

**Figure 15:** Proof of participation in competition.  
A screenshot of a computer

AI-generated content may be incorrect.

**Section 3 - San Francisco Clime Classification.**

This competition aimed at identifying the category of a crime based on metadata such as the location where the crime happened, date and time.

I explored the dataset and found that there were no missing values in it. Then I visualized the distribution of crime be police district and crime category. The view shown in Figure 16 below informs us about the data imbalance present in the dataset.  
**Figure 16:**  
Crime distribution by category and police district.

A graph of a number of people

AI-generated content may be incorrect. A graph of blue bars with black text

AI-generated content may be incorrect.  
**Note**: ***Left****: Some categories of crime like theft occur more frequently others like Gambling. This presents a data imbalance issue for any algorithem we might build.* ***Right****: There is more crime in some police districts than others with the Southern district having about 4 times as much crime as the Richmond district. This may reflect the poverty levels in those districts or it may reflect the population size of those disctrics.*

To attach a sense of geography to the crime distribution, I plotted the crimes by location and the results are shown in Figure 17 below.

**Figure 17:**

Crime distribution by distribution visuallized in a map like NNA map of different colored squares

AI-generated content may be incorrect.   
**Note:** *There were a few outlier locations in the dataset where the latitude and longitudes were way off the value of San Francisco’s location, I had to remove those because I was confident that there were entered in error.*

Feature Engineering.

I created new features such as day, month, and hour from the date of the incident. I also encoded the addresses and district using `OrdinalEncoder` from sklearn. The choice of Ordinal encoding for this is because location data inherently has some order defined by the sequential nature of numbers (Breskuvienė & Dzemyda, 2023).

I then plotted a correlation matrix of the numerical variables in the dataset to see if any variables were highly correlated and needed to be removed before the analysis. Then I used grid search to identify the correct number of trees for fitting various tree models. Figure 18 below shows the results of grid search for the number of estimators to use for a random forest model.

**Figure 18:**  
ROC AUC for test/training datasets by number of estimators for Random Forest Classifier.

A graph with red and green lines

AI-generated content may be incorrect.   
**Note**: *Using More than 8 classifiers does not yield any improvement in either test or training dataset. The test performance is also consistently below the training performance suggesting that a random forest classifier may not be a good model for this dataset.*

I also trained a support vector machine classifier which did not do very well. I attempted to change the kernel for the support vector machine classifier from linear to other kernels but could not do so because my computer was short of computing power. The best classifiers I got for this dataset was Gradient boosting classifier. The feature importance for this classifier compared to the random forest classifier is shown in the figure 20 below.

**Figure 20**:  
*Feature importance of Gradient Boosting vs Random Forest Classifier.*

A comparison of a bar graph

AI-generated content may be incorrect.

The side-by-side feature importance plots reveal distinct patterns in how Random Forest and Gradient Boosting models prioritize features for prediction. **Left:** In the Random Forest model, “Day” and “Month” exhibit the highest importance, suggesting temporal patterns are crucial. **Right:** Conversely, Gradient Boosting emphasizes spatial features like “Y” and “X” coordinates, along with address-related information (“intersection,” “EncodedAddress,” “EncodedDistrict”). This indicates that while both models leverage similar features, they assign different weights, highlighting the varying ways these algorithms capture underlying data relationships. The discrepancies also suggest potential areas for feature engineering or model refinement, as understanding these differences can lead to improved predictive performance.

**Figure 21:**   
*Proof of Participation in competition.*  
A screenshot of a computer

AI-generated content may be incorrect.

# Bibliography

Alexis Cook, D., & Holbrook, R. (2021). *Store Sales - Time Series Forecasting*. Retrieved from Kaggle.com: https://www.kaggle.com/competitions/store-sales-time-series-forecasting/overview

Breskuvienė, D., & Dzemyda, G. (2023, 5). Categorical Feature Encoding Techniques for Improved Classifier Performance when Dealing with Imbalanced Data of Fraudulent Transactions. *INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL*. doi:10.15837/ijccc.2023.3.5433

Gnat, S. (2021). Impact of Categorical Variables Encoding on Property Mass Valuation. *Procedia Computer Science, 192*, 3542-3550. doi:10.1016/j.procs.2021.09.127

Josse, J., & Husson, F. (2012). Selecting the number of components in principal component analysis using cross-validation approximations. *Computational Statistics & Data Analysis, 56*(6), 1869-1879. doi:10.1016/j.csda.2011.11.012

Kan, W. (2015). *San Francisco Crime Classification*. Retrieved from Kaggle.com: https://kaggle.com/competitions/sf-crime

King, M. (1981). The alternative Durbin-Watson test: An assessment of Durbin and Watson's choice of test statistic. *Journal of Econometrics, 14*(1), 51-66. doi:10.1016/0304-4076(81)90058-0

Montoya, A., & Canary, D. (2016). *House Prices - Advanced Regression Techniques.* Retrieved from Kaggle.com: https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques

Nde, S. M. (2025, 03). *DDS-8555*. Retrieved from Github: https://github.com/ndesamuelmbah/DDS-8555.git