Feature Engineering : Random Forest implementation

Overview

This project implemented feature engineering for retail demand forecasting using Random Forest regression. The goal was to improve prediction accuracy of product units sold by creating and evaluating new engineered features beyond the raw dataset.

New Features:

Time based features	Store related features	Product related features
Month, Year, Week of Year	4-week rolling average of units sold	Store-product average units
Holiday season indicator (Nov-Dec)	Store size categories (quartile-based)	
Summer season indicator (Jun-Aug)	Monthly store average units	

```
[16]: # time based features

merged_data['MONTH'] = merged_data['WEEK_END_DATE'].dt.month
merged_data['YEAR'] = merged_data['WEEK_END_DATE'].dt.year
merged_data['WEEK_OF_YEAR'] = merged_data['WEEK_END_DATE'].dt.isocalendar().week
merged_data['Is_Holiday_Season'] = merged_data['MONTH'].apply(lambda x: 1 if x in [11, 12] else 0) #  christmas
merged_data['Is_Summer'] = merged_data['MONTH'].apply(lambda x: 1 if x in [6,7,8] else 0) #summer months , when people take vacations
```

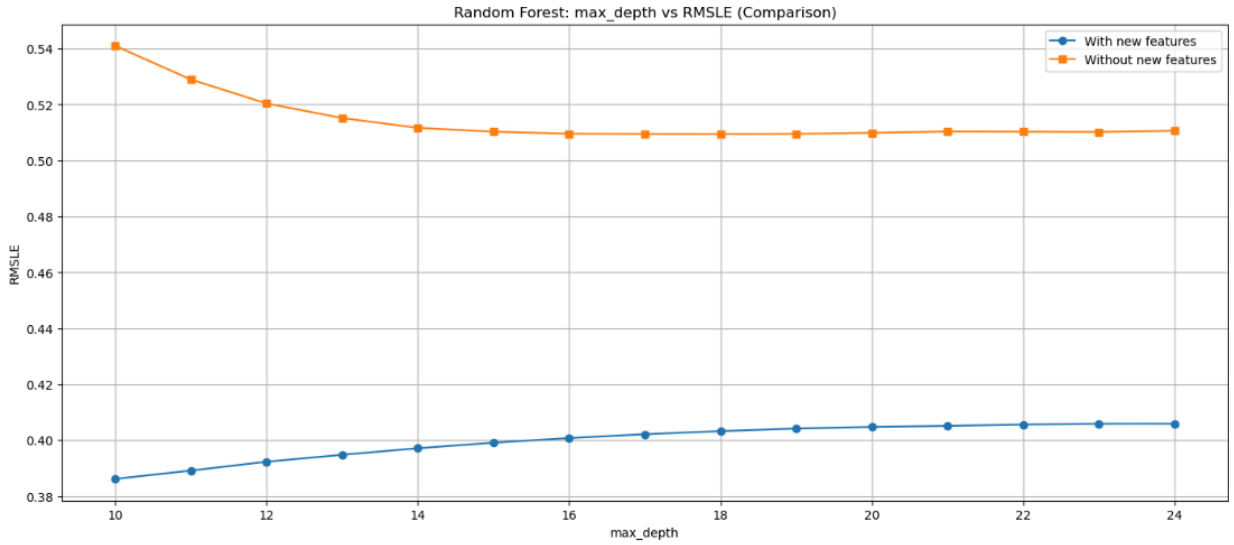
```
[17]: # store related features
merged_data['ROLLING_MEAN_4W'] = merged_data.groupby(['STORE_NUM', 'UPC'])['UNITS'].transform(lambda x: x.rolling(4, 1).mean()) # rolling mean of last 4
merged_data['ROLLING_MEAN_4W'] = merged_data['ROLLING_MEAN_4W'].fillna(0)

merged_data['STORE_SIZE_CAT'] = pd.qcut(merged_data['SALES_AREA_SIZE_NUM'], q=4, labels=[1,2,3,4])
merged_data['MONTH_STORE_AVG'] = merged_data.groupby(['STORE_NUM', 'MONTH'])['UNITS'].transform('mean')
```

```
[18]: #product relatd engineered features
merged_data['STORE_PROD_AVG'] = merged_data.groupby(['STORE_NUM', 'UPC'])['UNITS'].transform('mean')
```

Optimal Model Configuration:

The best performing model during the tuning used max_depth=10 with 400 estimators. The new features significantly outperformed baseline across all max depth configurations (10-24), however, the model showed diminishing returns beyond max depth=15, implying optimal complexity was achieved.



Key Results:

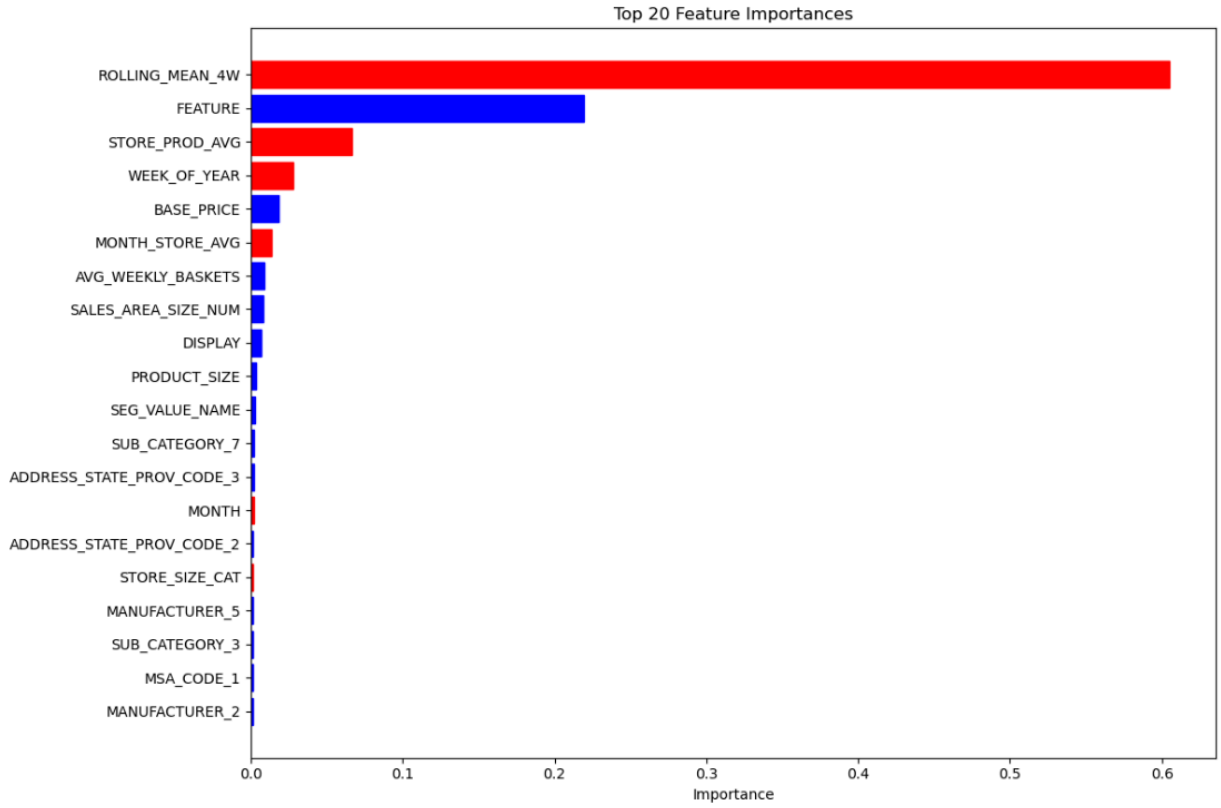
Performance Improvement with New Features:

Iteration	**validation RMSLE**	**Improvement**
Baseline Model (no new features)	0.5094	
Enhanced Model (with new features)	0.3861	0.1233 (24.19%)

Feature Importance Analysis:

Feature	**Importance**
ROLLING_MEAN_4W	60.52%
FEATURE	21.91%
STORE_PROD_AVG	6.65%
WEEK_OF_YEAR	2.78%
BASE_PRICE	1.85%
MONTH_STORE_AVG	1.33%
AVG_WEEKLY_BASKETS	0.85%
SALES_AREA_SIZE_NUM	0.83%
DISPLAY	0.65%
PRODUCT_SIZE	0.30%

SEG_VALUE_NAME	0.25%
SUB_CATEGORY_7	0.22%
ADDRESS_STATE_PROV_CODE_3	0.20%
MONTH	0.18%
ADDRESS_STATE_PROV_CODE_2	0.15%
STORE_SIZE_CAT	0.15%
MANUFACTURER_5	0.14%
SUB_CATEGORY_3	0.13%
MSA_CODE_1	0.11%
MANUFACTURER_2	0.10%



The red bars shown above are the new features, while the blue ones are the existing features.

Conclusion

This project demonstrates the value ofl feature engineering in retail demand forecasting. By incorporating time related patterns, store characteristics, and product-related characteristics, the model achieved substantial performance gains. The rolling 4-week average was the most powerful predictor, highlighting the importance of recent historical data in retail forecasting.