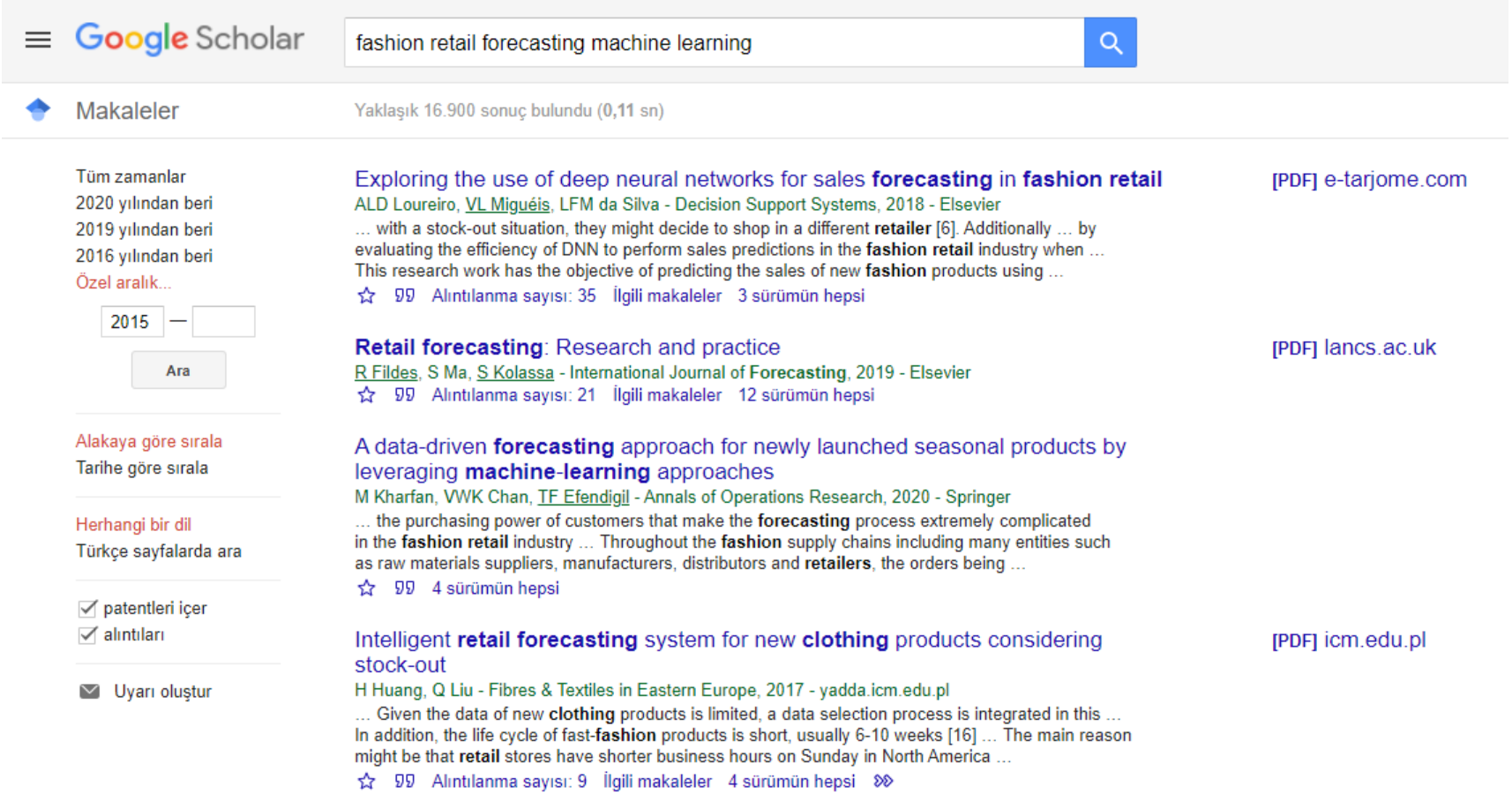
Forecasting with Ensemble Methods: An Application Using Fashion Retail Sales Data

Orkun Berk Yüzbaşıoğlu

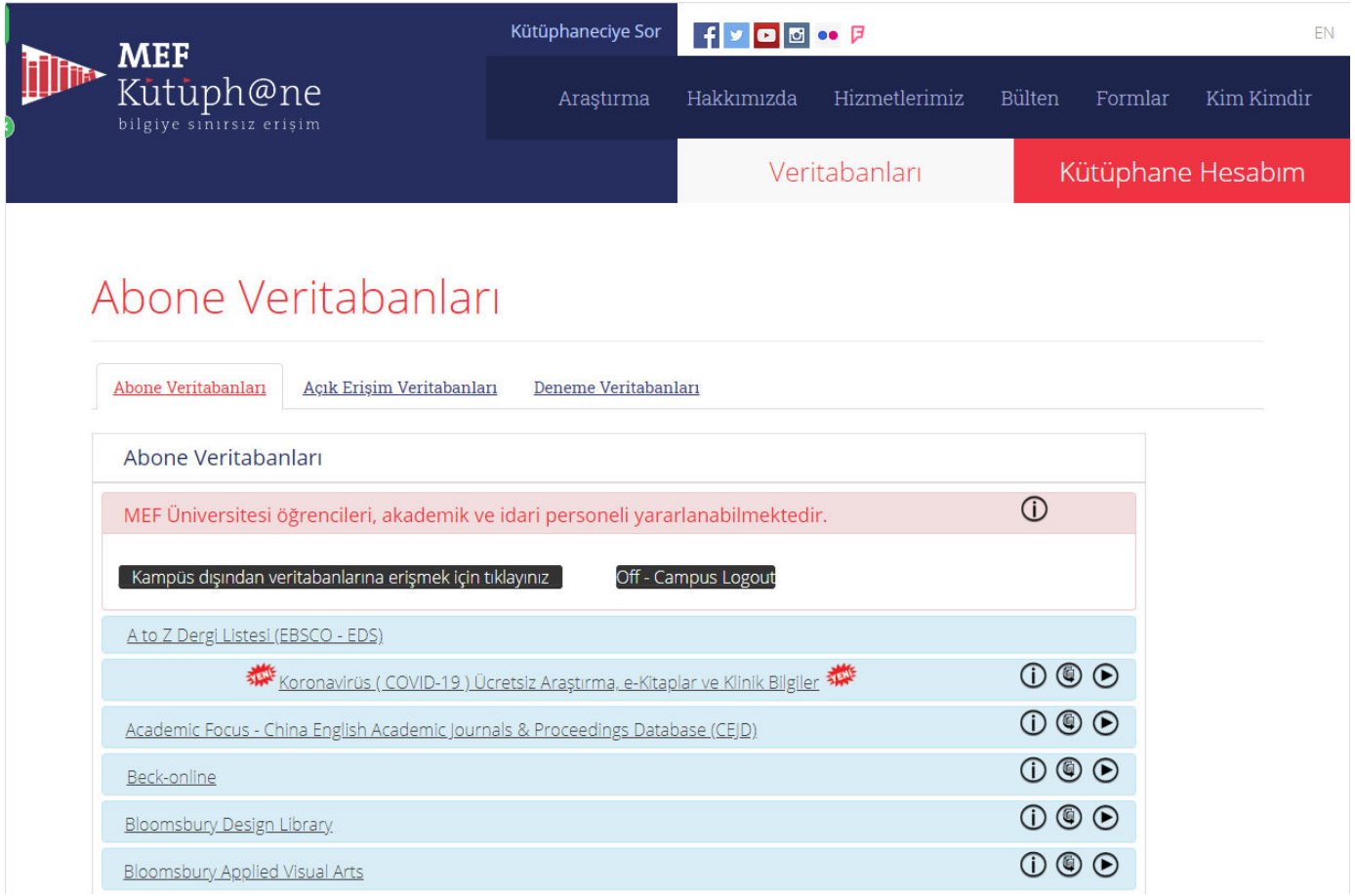
Summary

- In this project, ensemble machine learning models are used to predict short term store sales of a fashion retailer.
- Sales forecasts of various products at different stores are generated for a span of three months with bagging tree regressor, random forest regressor, and gradient boosting regressor algorithm.
- Algorithms are trained and evaluated with real past sales data of a Turkish fashion retailer.

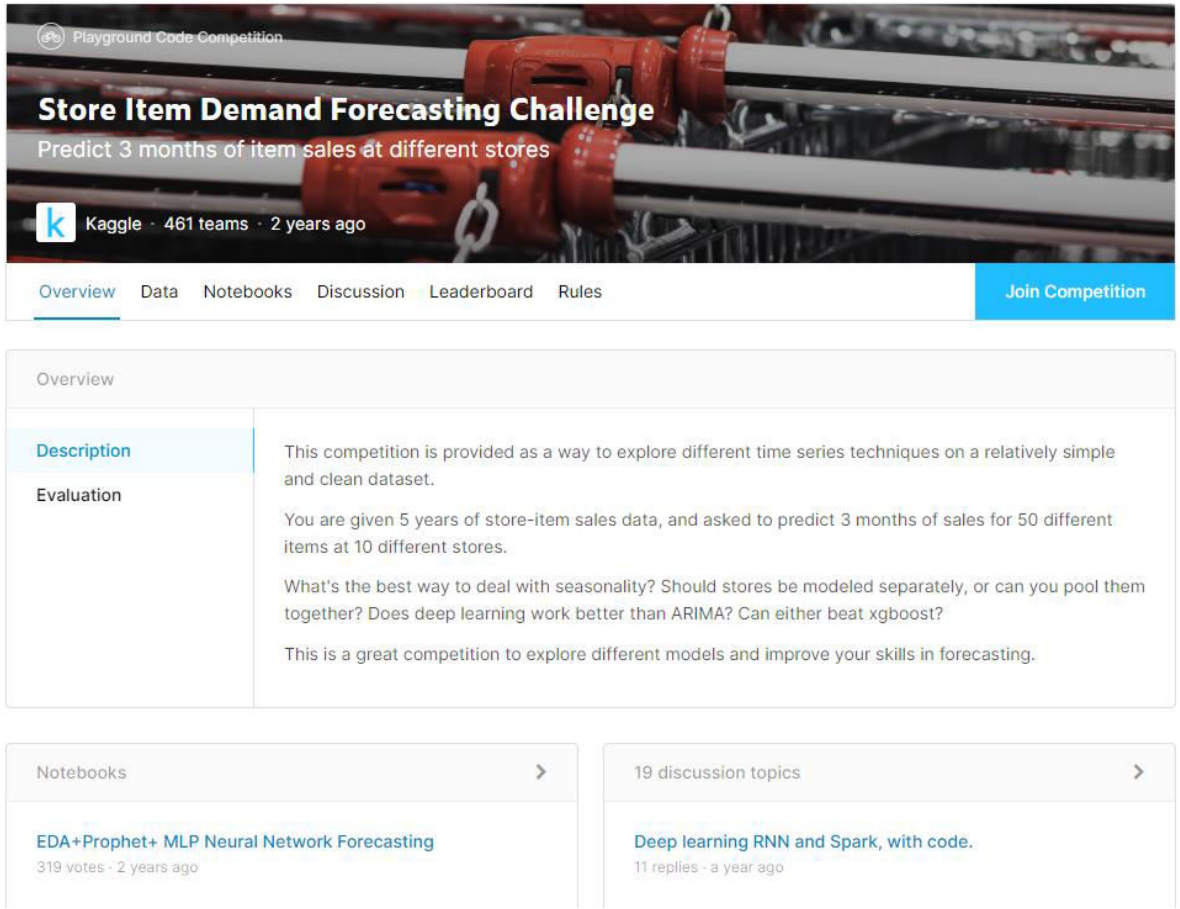
1. Literature Review on Forecasting Applied for Fashion Industry and Fashion Retail Sales (Theoretical & Academic)



1. Literature Review on Forecasting Applied for Fashion Industry and Fashion Retail Sales (Theoretical & Academic)



1. Literature Review on Forecasting Applied for Fashion Industry and Fashion Retail Sales (Practical)



2. Data Retrieval

- Sales Data (From RBDMS)
- Past Weather (NOAA)
- Weather Forecast (scrapy or from API)
- Special Dates (calendar package)
- Google Trends (pytrends package)

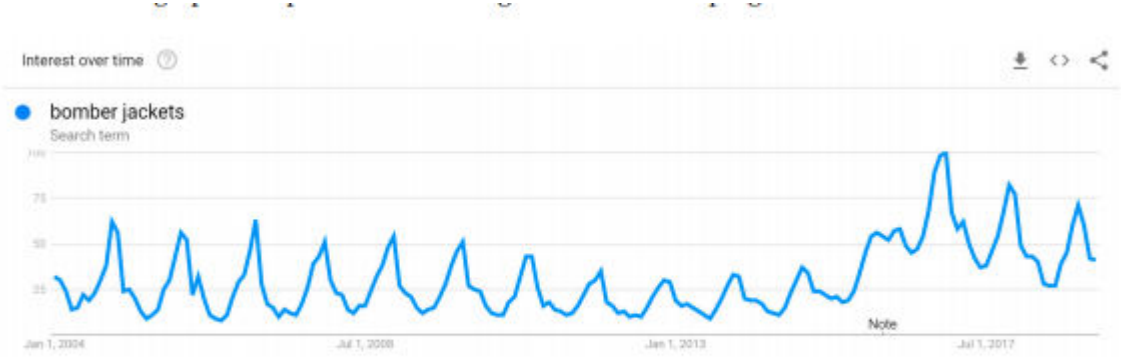


Figure 3. Google Trends for bomber jackets (January 2004–February 2019) (Data Source: Google Trends, 1 February 2019).

3. EDA

- To look for patterns in data
- For feature engineering
- For time series data: ACF, Line Charts

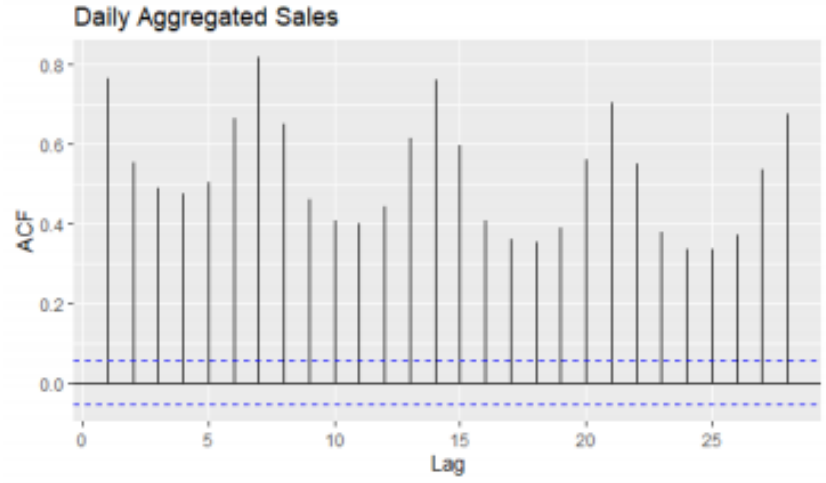


Figure 2. ACF plot of Daily Sales

Figure. ACF Plot

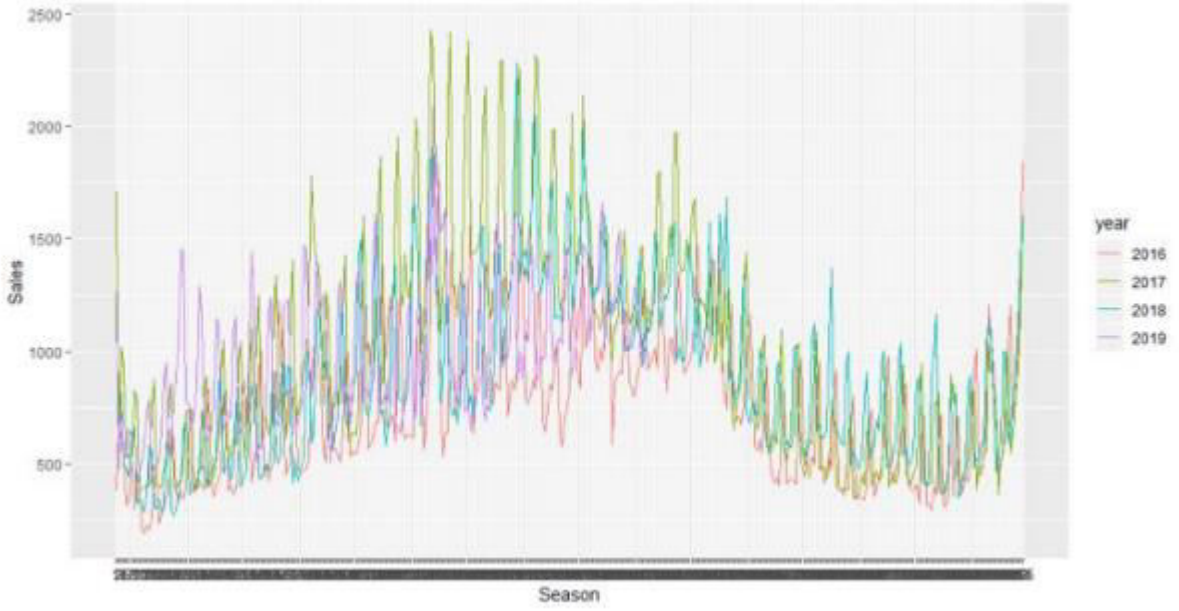


Figure. Line Chart

4. Feature Engineering

- Lagged Sales Features (from ACF plot sales of the same day from the week before last)
- Special Dates (such as Valentine's day, Republic Day (29 October))
- Geographic Features (city, country)
- Date Time Features:
  - Day of Week (1-7)
  - Day of Month (1-31)
  - Week of Year (1-52)*
  - Month (1-12),
  - Weekend (0, 1)

5. Feature Selection

- 1. Feature Importance from Random Forest

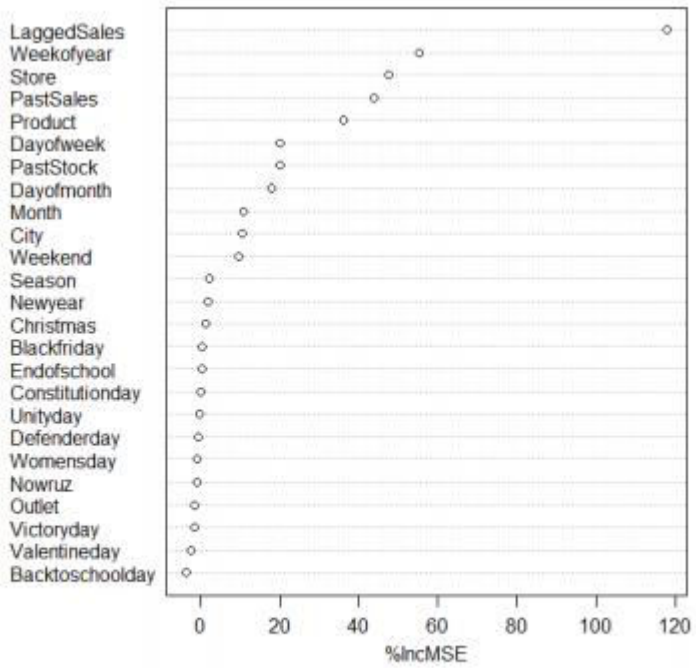


Figure 8. Variable Importance from Bagging Regression Tree

5. Feature Selection

- Scikit-Learn's **Feature Selection** Class (https://scikit-learn.org/stable/modules/feature_selection.html)
- **Removing Features with low variance:** *VarianceThreshold is a simple baseline approach to feature selection. It removes all features whose variance doesn't meet some threshold. By default, it removes all zero-variance features*
- **Recursive Feature Elimination with an estimator:** *Given an external estimator that assigns weights to features (e.g., the coefficients of a linear model), recursive feature elimination (RFE) is to select features by recursively considering smaller and smaller sets of features.*

6. Model Training

- Train-Validation-Test Split: No hyper-parameter tuning based on test Set

- Hyper-parameter's of model: From package documentation (scikit-learn, caret, lightgbm, xgboost, catboost, ngboost, tensorflow)

- Hyper-parameter tuning:
  - RandomizedSearch
  - GridSearch

7. Evaluating Results

- Metrics to compare different Models & Settings:

  RMSE, MAPE, MSE

- Visualizing Results:
  - Scatter Plots (y-test vs y-pred)
  - Residual/Error Plots (error vs y-test)

	Test Set R^2	RMSE
Bagging Regression Tree	70.8 %	8.05
Random Forest Regressor	72.88 %	7.71
Gradient Boosted Regressor	71.07%	8.01
Linear Regression	70.03%	8.06