T.C.

MARMARA UNIVERSITY

FACULTY of ENGINEERING

CSE4062 - Introduction to Data Science and Analytics

Spring 2024 - Delivery #3 - Predictive Analytics

# “Predicting Revenue Trends Using Machine Learning”

## Group 1

| Full Name | Student ID | Department | E-Mail |
|---|---|---|---|
| Duygu Yasinoğlu | 150122982 | Computer Engineering | duyguyasinoglu@gmail.com |
| Erdem Pehlivanlar | 150119639 | Computer Engineering | erdemphl@gmail.com |
| Mustafa Özgür Hocaoğlu | 150120058 | Computer Engineering | mustafaozgurh@gmail.com |
| Yunus Emre Dar | 150321823 | Industrial Engineering | yunusdar@marun.edu.tr |
| Muhammed Emin Bardakcı | 150319057 | Industrial Engineering | eminbrdk4@gmail.com |
| Deniz Yücetepe | 150620027 | Chemical Engineering | denizyucetepe@marun.edu.tr |

# Table Of Contents

# 1.Introduction

Our project revolves around the application of various classification algorithms, ranging from traditional methods such as k-Nearest Neighbors (k-NN) and Naive Bayes, to more advanced techniques including Support Vector Machines (SVM), Naive Bayes(NB) and Decision Trees (DT). We are also encouraged to explore ensemble methods such as Random Forest and Boosting Trees, leveraging the collective intelligence of multiple models to enhance predictive performance.

By evaluating different models based on metrics such as accuracy, F1 score, and AUC, we aim to identify the most effective approach for predicting outcomes in our dataset. Through this, we seek to gain valuable insights regarding the data we have.

# 2.Data Preprocessing

Our dataset contains some missing and noisy values. We applied some steps to clean the data:

- Some of the columns used to have negative or NaN values, replaced them with mean of the that specific column (only positive values are used to calculate the mean)
- By using Pandas' "***drop_duplicates***" function we removed duplicate rows from the dataset.
- Since we have a very large dataset which has 180K instances it was hard to process the next operation. Therefore, we used the first 100K instances (rows) of the dataset for other tasks.

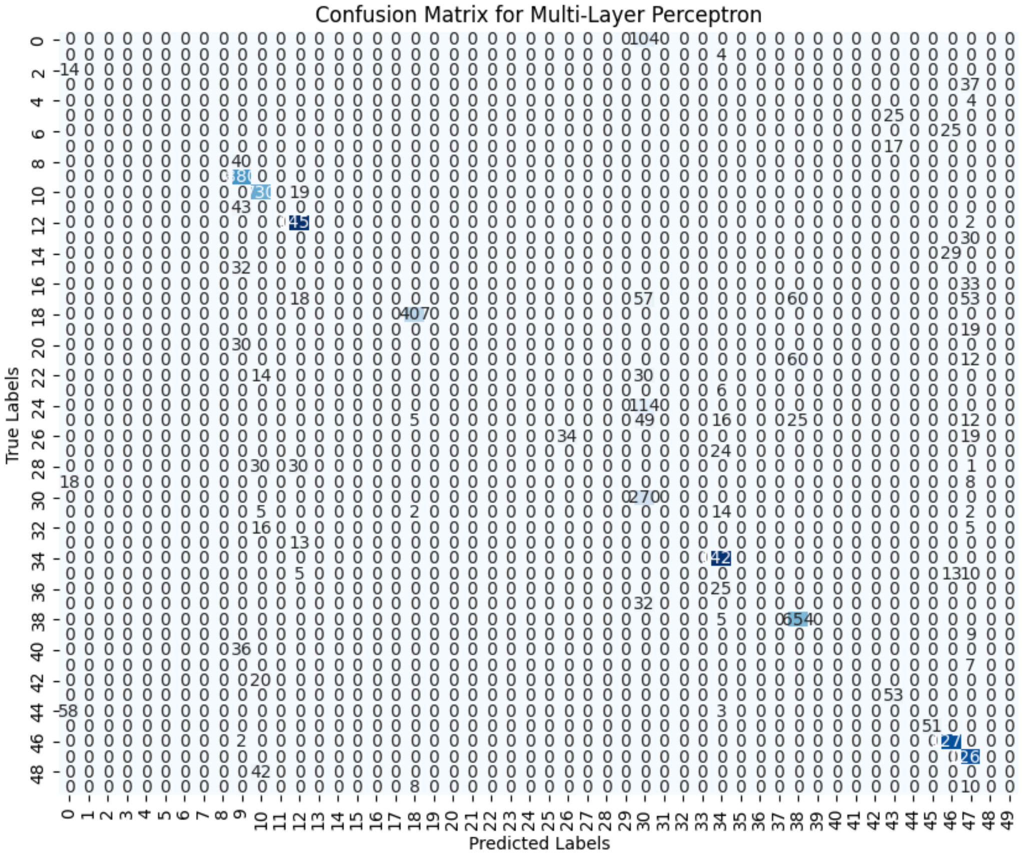
# 3. Feature Selection

We applied three different feature selection algorithm:

- **SET#1:**The first set of selected features extract from Random Forest Algorithm:

  **['Order Item Cardprod Id', 'Category Id', 'Product Price','Product Card Id', 'Department Id']**

  **A sample of Confusion Matrix of set #1:**


Confusion Matrix for Multi-Layer Perceptron
True Labels
Predicted Labels

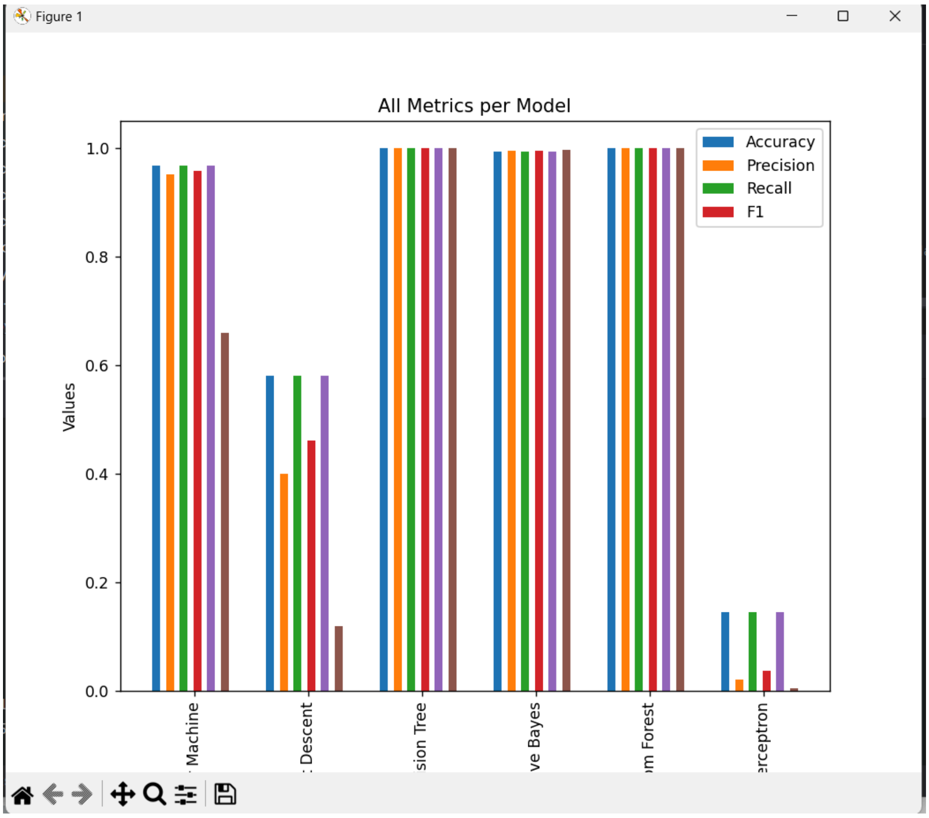
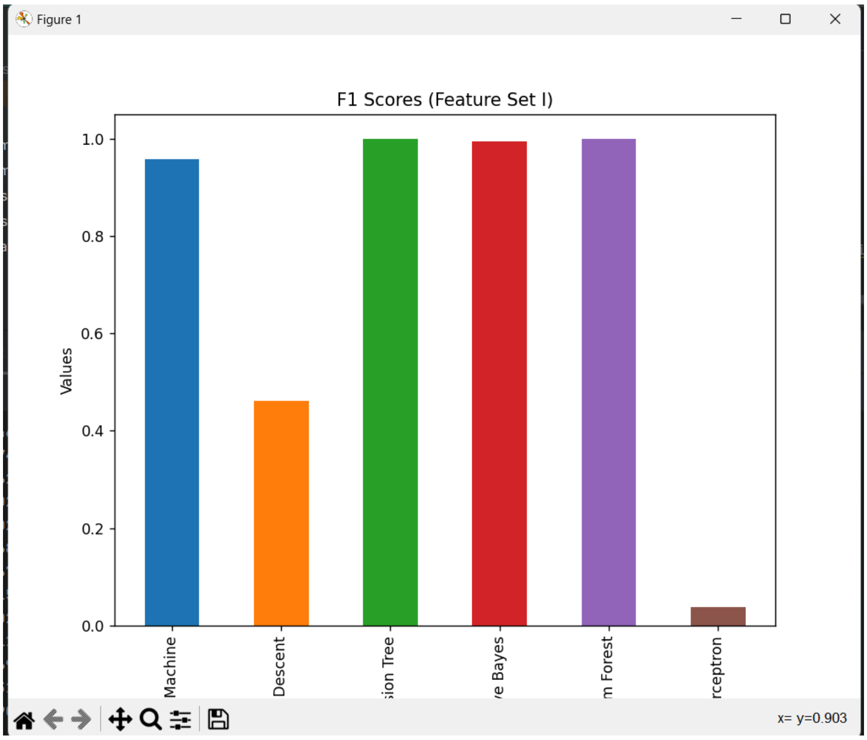


Figure 1
All Metrics per Model
Values
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
Precision
Recall
F1
Machine
Descent
sion Tree
ve Bayes
m Forest
erceptron



- **t-statistic: -5.694681125420523**
- **p-value: 9.212483171237696e-08**

- **SET#2:**The second set of selected features extract from Information Gain Algorithm:

  **['Order Item Id','Order Item Cardprod Id','Category Id','Product Card Id','Product Price']**

  **A sample of Confusion Matrix of set #2:**

  - The third set of selected features extract from Recursive Feature Elimination Algorithm:

  **['Sales','Order Profit Per Order','Category Id','Product Card Id','Product Price']**

  **A sample of Confusion Matrix of set #3:**

Table 1. Selected Features

| Column Name | Feature Selection Method 1:Random Forest Algorithm | Feature Selection Method 2: Information Gain | Feature Selection Method 3: Recursive Feature Elimination |
|---|---|---|---|
| **Order Item Cardprod Id** | + | + | |
| **Category Id** | + | | + |
| **Product Price** | + | + | + |
| **Product Card Id** | + | + | + |
| **Department Id** | + | | |
| **Sales** | | | + |
| **Order Item Id** | | + | |
| **Order Profit Per Order** | | | + |

# 3. Results

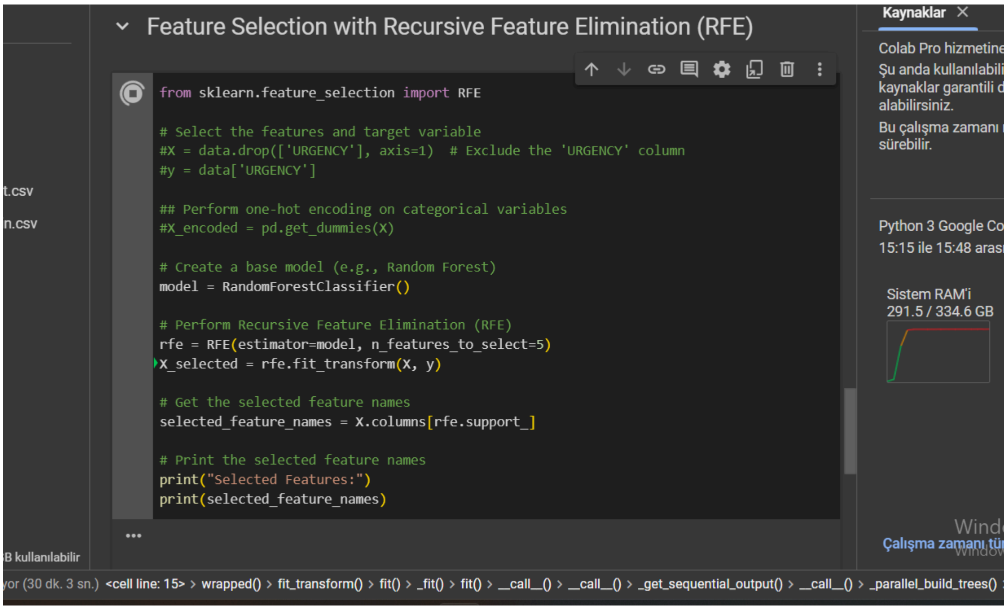
Table 2. Test Results Comparison

| Algorithm | F1 Score | Accuracy | Precision | Recall | Selected Feature Set1 | Selected Feature Set2 | Selected Feature Set3 |
|---|---|---|---|---|---|---|---|
| **SVM** | 0.95766 | 0.96783 | 0.95159 | 0.96783 | X | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **SVM** | | | | | | X | |
| **SVM** | | | | | | | X |
| **Stochastic Gradient Descent** | 0.46260 | 0.53031 | 0.42579 | 0.53031 | X | | |
| **Stochastic Gradient Descent** | | | | | | X | |
| **Stochastic Gradient Descent** | | | | | | | X |
| **Decision Tree** | 1.0 | 1.0 | 1.0 | 1.0 | X | | |
| **Decision Tree** | | | | | | X | |
| **Decision Tree** | | | | | | | X |
| **Naive Bayes** | 0.99605 | 0.99605 | 0.99675 | 0.99621 | X | | |
| **Naive Bayes** | | | | | | X | |
| **Naive Bayes** | | | | | | | X |
| **Random Forest** | 1.0 | 1.0 | 1.0 | 1.0 | X | | |
| **Random Forest** | | | | | | X | |
| **Random Forest** | | | | | | | X |
| **Multi-Layer Perceptron** | 0.47561 | 0.47561 | 0.35198 | 0.38170 | X | | |
| **Multi-Layer Perceptron** | | | | | | X | |
| **Multi-Layer Perceptron** | | | | | | | X |

# 4. Conclusion

The classification results for selected features showcase the effectiveness of the selected features - 'Order Item Cardprod Id', 'Category Id', 'Product Price', 'Product Card Id', - in predicting the target Category Name. Notably, all algorithms achieved high performance, with Decision Tree, Naive Bayes, and Random Forest attaining perfect scores across all metrics. This indicates that the chosen features are highly discriminative and contribute significantly to accurate classification. Among the algorithms, Decision Tree, Naive Bayes, and Random Forest stand out as particularly effective for this feature set, demonstrating flawless performance. Overall, Feature Set #0 exhibits strong predictive power, emphasizing the importance of feature selection in achieving optimal classification outcomes.



Feature Selection with Recursive Feature Elimination (RFE)

```
from sklearn.feature_selection import RFE

# Select the features and target variable
#X = data.drop(['URGENCY'], axis=1)  # Exclude the 'URGENCY' column
#y = data['URGENCY']

## Perform one-hot encoding on categorical variables
#X_encoded = pd.get_dummies(X)

# Create a base model (e.g., Random Forest)
model = RandomForestClassifier()

# Perform Recursive Feature Elimination (RFE)
rfe = RFE(estimator=model, n_features_to_select=5)
X_selected = rfe.fit_transform(X, y)

# Get the selected feature names
selected_feature_names = X.columns[rfe.support_]

# Print the selected feature names
print("Selected Features:")
print(selected_feature_names)
```

Colab Pro hizmetine
Şu anda kullanılabili
kaynaklar garantili d
alabilirsiniz.
Bu çalışma zamanı
sürebilir.

Python 3 Google Co
15:15 ile 15:48 arası

Sistem RAM'i
291.5 / 334.6 GB

Çalışma zamanı tü

yor (30 dk. 3 sn.) <cell line: 15> > wrapped() > fit_transform() > fit() > _fit() > fit() > __call__() > __call__() > _get_sequential_output() > __call__() > _parallel_build_trees()

Picture of RAM usage(291gb)