MAJOR PROJECT

END TERM

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

ON

BLACK FRIDAY ANALYSIS USING MACHINE LEARNING

Submitted by:

Aashish Kumar(BFSI)

Aryan Tayal(ECRA)

Vishwatej Ajay(ECRA)

Under the guidance of

Dr. Piyush Chauhan

Associate Professor

Department of Informatics

School of Computer Science and Engineering

Department of Informatics

UPES, Dehradun-248007

2016-2020

School of Computer Science

and Engineering

UPES, Dehradun

Project Proposal Approval (2020)

Project Title: Black Friday Analysis using machine learning

Abstract:

Customer behavior refers to an individual's buying habits, including social trends, frequency patterns, and background factors influencing their decision to buy something. Businesses study customer behavior to understand their target audience and create more-enticing products and service offers. [1]

Customer behavior doesn't describe who is shopping in your stores, but, instead, how they're shopping in your stores. It assesses factors like how frequently customers shop, which products they prefer, and how they perceive your marketing, sales, and customer service offers. Understanding these details helps businesses communicate with customers in a productive and delightful way.

Introduction:

A customer behavior analysis is a qualitative and quantitative observation of how customers interact with your company. Customers are first segmented into buyer personas based on their common characteristics. Then, each group is observed at the stages on your customer journey map to analyze how the personas interact with your company.

A customer behavior analysis provides insight into the different variables that influence an audience. It gives you an idea of the motives, priorities, and decision-making methods being considered during the customer's journey. This analysis helps you understand how customers feel about of your company, as well as if that perception aligns with their core values.

Problem statement

A retail company “ABC Private Limited” wants to understand the customer purchase behaviour (specifically, purchase amount) against various products of different categories. They have shared purchase summary of various customers for selected high volume products from last month.

The data set also contains customer demographics (age, gender, marital status, city_type, stay_in_current_city), product details (product_id and product category) and Total purchase_amount from last month.

Now, they want to build a model to predict the purchase amount of customer against various products which will help them to create personalized offer for customers against different products.

Literature review

Customer Behavior Modeling is defined as the creation of a mathematical construct to represent the common behaviors observed among particular groups of customers in order to predict how similar customers will behave under similar circumstances.
Customer behavior models are typically based on data mining of customer data, and each model is designed to answer one question at one point in time. For example, a customer model can be used to predict what a particular group of customers will do in response to a particular marketing action. If the model is sound and the marketer follows the recommendations it generated, then the marketer will observe that a majority of the customers in the group responded as predicted by the model.[2]

Objective- Objectives of this project are

1. To understand the literature about customer behavior analysis.
2. To learn about Feature Engineering.
3. To learn and implement machine learning algorithms namely Linear regression, Decision tree etc.
4. To understand predictive modelling using Rapidminer.

Methodology:

Phase 1:Exploratory Data Analysis

- Calculation of Measures of Central tendency and measures of dispersion
- Univariate Data Analysis
- Bivariate Data Analysis

Description:

The train and test dataset have been merged and all basic statistical measures have been calculated. The attributes have been categorized into numerical and categorical. The correlation between the numerical predictors has been calculated and plotted in the form of a heatmap. The distribution of categorical predictors has also been plotted. Next , the impact of numerical and categorical predictors on the target variable has been studied by plotting them against the target variable.

Phase 2: Data Preprocessing

1. Imputation of missing values
2. Outlier Detection
3. Feature Engineering

Description:

In this phase, firstly , the missing values have been detected and imputed with appropriate values based on the attributes category. Duplicate values among the columns are detected. Next, outliers have been detected and dealt with if there are any. Unnecessary columns have been dropped . Columns have been modified and converted to numerical columns. Other categorical columns have been converted using Label Encoder and One Hot encoder. After that , the data has been split into train and test set and exported in csv format.

Phase 3: Modeling

1. Linear Regression
2. Decision Tree model

Description:

In this phase, a linear regression model has been built and fit to the train data. A decision tree model has also been built. The models have been trained on the trained data and then applied on the test set. The predicted values have been compared against the actual values and RMSE has been calculated.

Phase 4: Model Validation and evaluation

- Cross Validation
- Tuning

1.1 DEFINITION OF PREDICTIVE MODELING:

Predictive modeling is a process that uses data and statistics to predict outcomes with data models. These models can be used to predict anything from sports outcomes and TV ratings to technological advances and corporate earnings.

These synonyms are often used interchangeably. However, predictive analytics most often refers to commercial applications of predictive modeling, while predictive modeling is used more generally or academically. Of the terms, predictive modeling is used more frequently, which is illustrated in the Google Trends chart below. Machine learning is also distinct from predictive modeling and is defined as the use of statistical techniques to allow a computer to construct predictive models. In practice, machine learning and predictive modeling are often used interchangeably. However, machine learning is a branch of artificial intelligence, which refers to intelligence displayed by machines.

1.2 OVERVIEW:

Predictive modeling is useful because it gives accurate insight into any question and allows users to create forecasts. To maintain a competitive advantage, it is critical to have insight into future events and outcomes that challenge key assumptions.

Analytics leaders must align predictive modeling initiatives with an organization's strategic goals. For example, a computer chip manufacturer might set a strategic priority to produce chips with the greatest number of transistors in the industry by 2025. Analytics professionals could construct a predictive model to forecast the number of transistors per chip to become a leader if they feed the model product, geography, sales, and other related trend data. Additional sources could include data about the most transistor-dense chips, commercial demand for computing power, and strategic partnerships between chip manufacturers and hardware manufacturers. Once initiatives are in motion, analytics professionals can perform backward-looking analyses to assess the accuracy of predictive models and the success of the initiatives.

1.3 BENEFITS OF PREDICTIVE MODELING:

In its multiple forms—predictive modeling, decision analysis and optimization, transaction profiling, and predictive search—predictive analytics can be applied to a range of business strategies and has been a key player in search advertising and recommendation engines. These techniques can provide managers and executives with decision-making tools to influence upselling, sales and revenue forecasting, manufacturing optimization, and even new product development. Though useful and beneficial, predictive analytics isn't for everyone.

1.4 Code and Demo:

Importing libraries:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Importing Training and Test Dataset:

```
train=pd.read_csv("train_modified.csv")
test=pd.read_csv("test_modified.csv")
```

Model Building:

```
#Define target and ID columns:
target = 'Item_Outlet_Sales'
IDcol = ['Item_Identifier','Outlet_Identifier']
```

```
#Define target and ID columns:
target = 'Purchase'
IDcol = ['User_ID','Product_ID']

import sklearn
from sklearn.model_selection import cross_validate
from sklearn import metrics
from sklearn.model_selection import cross_val_score


def modelfit(alg, dtrain, dtest, predictors, target, IDcol, filename):

    #Fit the algorithm on the data
    alg.fit(dtrain[predictors], dtrain[target])

    #Predict training set:
    dtrain_predictions = alg.predict(dtrain[predictors])


cv_score=sklearn.model_selection.cross_val_score(alg,
dtrain[predictors],(dtrain[target]),cv=20, scoring='neg_mean_squared_error')

np.sqrt(np.abs(cv_score))

#Print model report:
    print("\nModel Report")

print("RMSE:%.4g"%np.sqrt(metrics.mean_squared_error((dtrain[target]).v
alues, dtrain_predictions)))


print("CV Score : Mean - %.4g | Std - %.4g | Min - %.4g | Max - %.4g" %
(np.mean(cv_score),np.std(cv_score),np.min(cv_score),np.max(cv_score)))

#Predict on testing data:
    dtest[target] = alg.predict(dtest[predictors])

    #Export submission file:
    IDcol.append(target)
    submission = pd.DataFrame({ x: dtest[x] for x in IDcol})
```

```
submission.to_csv(filename, index=False)
```

Linear Regression Model:

```
from sklearn.linear_model import LinearRegression
LR = LinearRegression(normalize=True)
predictors = train_df.columns.drop(['Purchase','Product_ID','User_ID'])
modelfit(LR, train_df, test_df, predictors, target, IDcol, 'LR.csv')
coef1 = pd.Series(LR.coef_, predictors).sort_values()
coef1.plot(kind='bar', title='Model Coefficients')
```

Ridge Regression Model:

```
from sklearn.linear_model import Ridge
RR = Ridge(alpha=0.05,normalize=True)
modelfit(RR, train_df, test_df, predictors, target, IDcol, 'RR.csv')
```

Decision Tree Model:

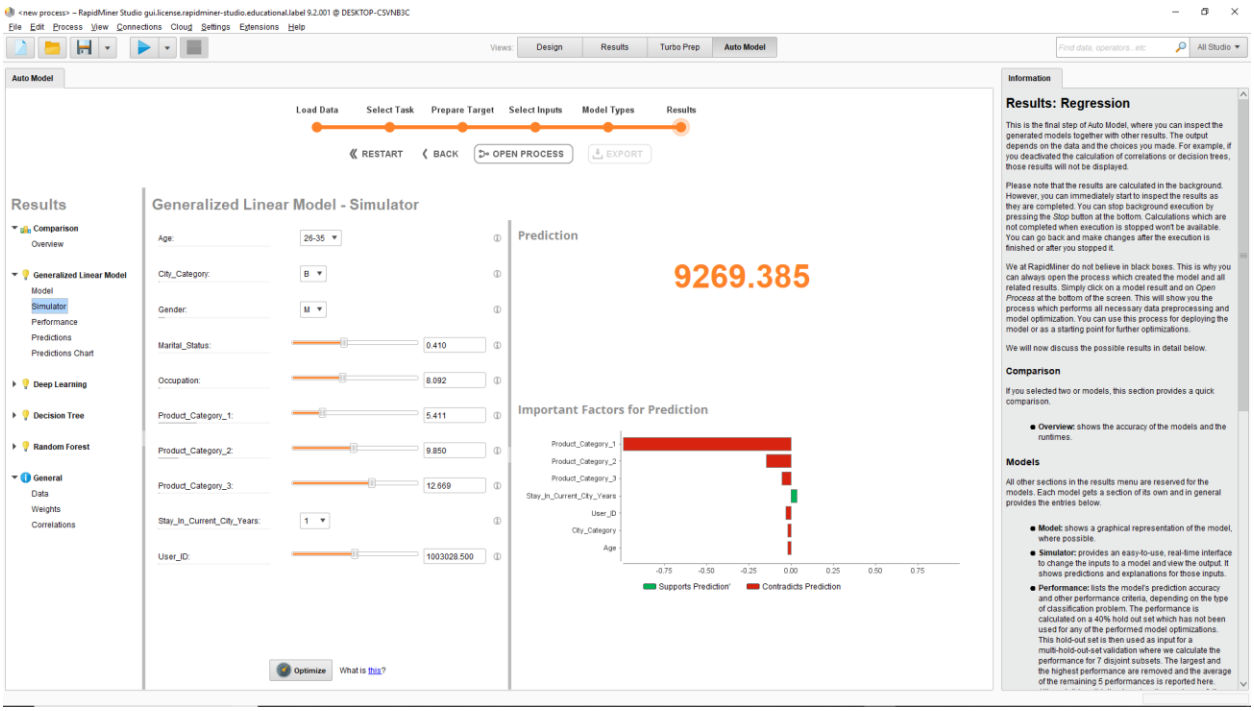
```
from sklearn.tree import DecisionTreeRegressor
DT = DecisionTreeRegressor(max_depth=15, min_samples_leaf=100)
modelfit(DT, train_df, test_df, predictors, target, IDcol, 'DT.csv')
```

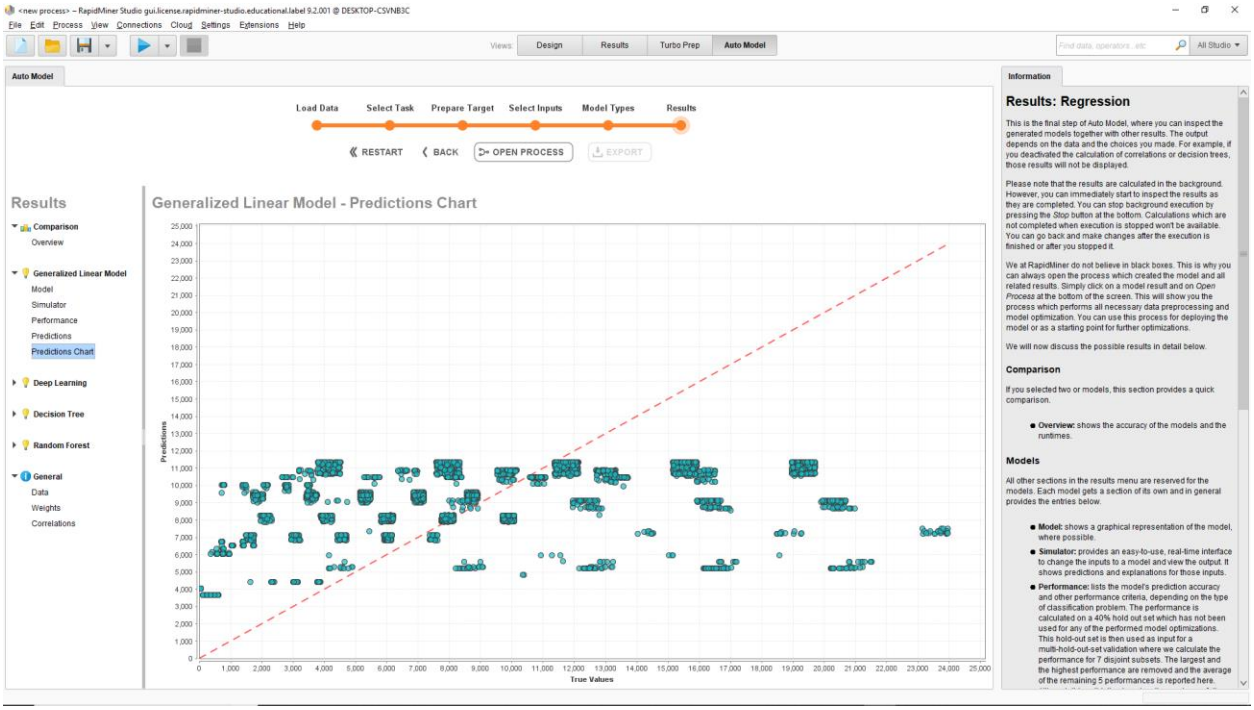
Random Forest Model:

```
RF = DecisionTreeRegressor(max_depth=8, min_samples_leaf=150)
modelfit(RF, train_df, test_df, predictors, target, IDcol, 'RF.csv')
```

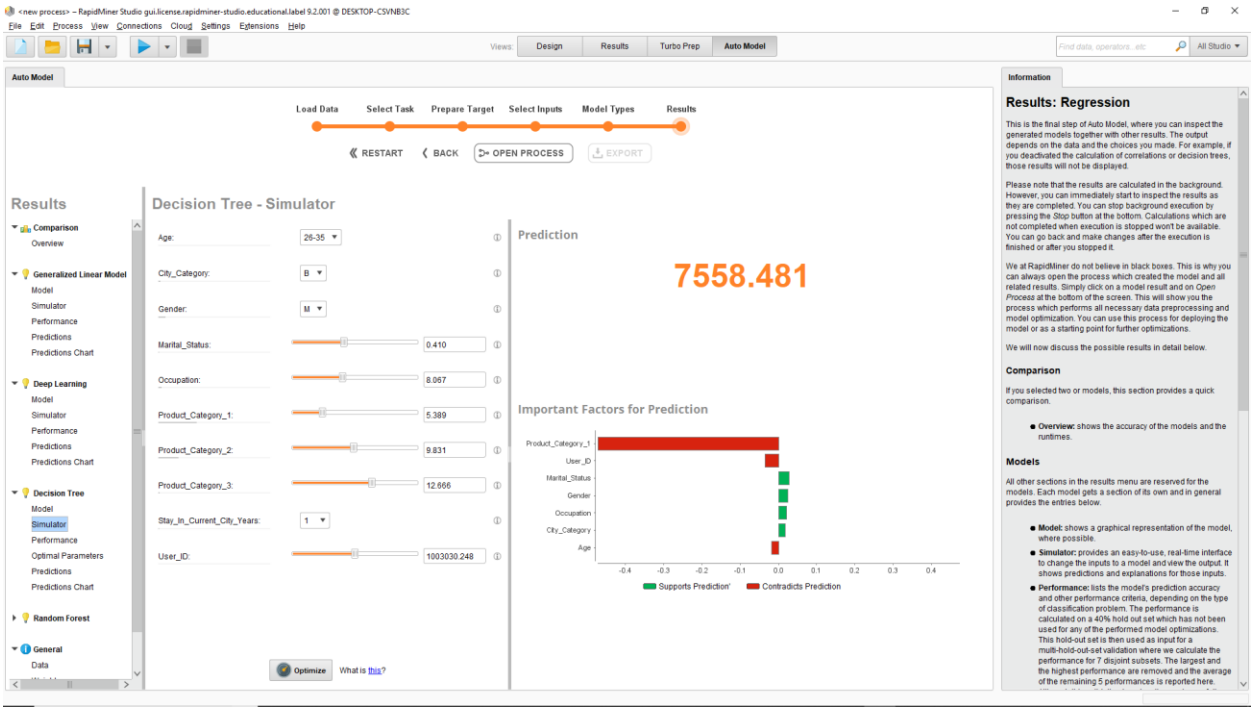
RESULT:

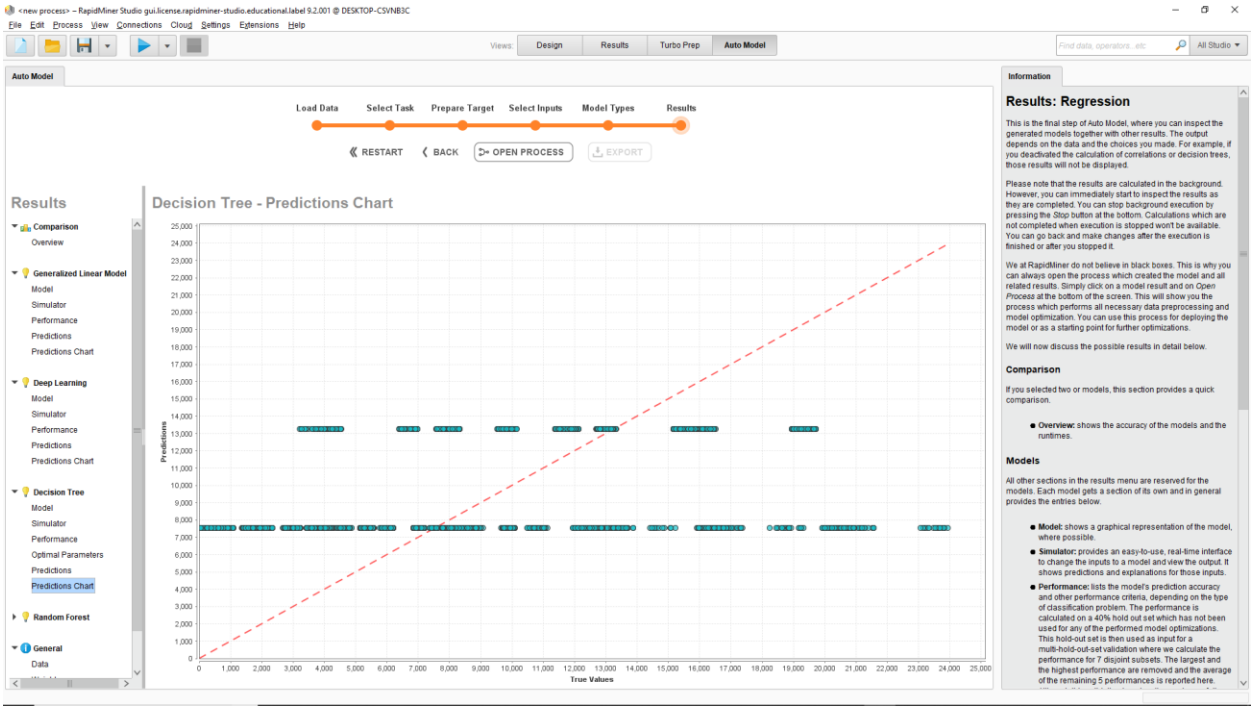
Generalized Linear Model:


<new process> – RapidMiner Studio gui.license.rapidminer-studio.educational.label 9.2.001 @ DESKTOP-CSVNB3C
File Edit Process View Connections Cloud Settings Extensions Help
Views: Design Results Turbo Prep Auto Model
Auto Model
Load Data Select Task Prepare Target Select Inputs Model Types Results
RESTART BACK OPEN PROCESS EXPORT
Results
Comparison
Overview
Generalized Linear Model
Model
Simulator
Performance
Predictions
Predictions Chart
Deep Learning
Decision Tree
Random Forest
General
Data
Weights
Correlations
Generalized Linear Model - Simulator
Age: 26-35
City_Category: B
Gender: M
Marital_Status: 0.410
Occupation: 8.092
Product_Category_1: 5.411
Product_Category_2: 9.850
Product_Category_3: 12.669
Stay_In_Current_City_Years: 1
User_ID: 1003028.500
Optimize What is this?
Prediction
9269.385
Important Factors for Prediction
Product_Category_1
Product_Category_2
Product_Category_3
Stay_In_Current_City_Years
User_ID
City_Category
Age
-0.75 -0.50 -0.25 0.00 0.25 0.50 0.75
Supports Prediction
Contradicts Prediction
Information
Results: Regression
This is the final step of Auto Model, where you can inspect the generated models together with other results. The output depends on the data and the choices you made. For example, if you deactivated the calculation of correlations or decision trees, those results will not be displayed.
Please note that the results are calculated in the background. However, you can immediately start to inspect the results as they are completed. You can stop background execution by pressing the Stop button at the bottom. Calculations which are not completed when execution is stopped won't be available. You can go back and make changes after the execution is finished or after you stopped it.
We at RapidMiner do not believe in black boxes. This is why you can always open the process which created the model and all related results. Simply click on a model result and on Open Process at the bottom of the screen. This will show you the process which performs all necessary data preprocessing and model optimization. You can use this process for deploying the model or as a starting point for further optimizations.
We will now discuss the possible results in detail below.
Comparison
If you selected two or models, this section provides a quick comparison.
Overview: shows the accuracy of the models and the runtimes.
Models
All other sections in the results menu are reserved for the models. Each model gets a section of its own and in general provides the entries below.
Model: shows a graphical representation of the model, where possible.
Simulator: provides an easy-to-use, real-time interface to change the inputs to a model and view the output. It shows predictions and explanations for those inputs.
Performance: lists the model's prediction accuracy and other performance criteria, depending on the type of classification problem. The performance is calculated on a 40% hold out set which has not been used for any of the performed model optimizations. This hold-out set is then used as input for a multi-hold-out-set validation where we calculate the performance for 7 disjoint subsets. The largest and the highest performance are removed and the average of the remaining 5 performances is reported here.

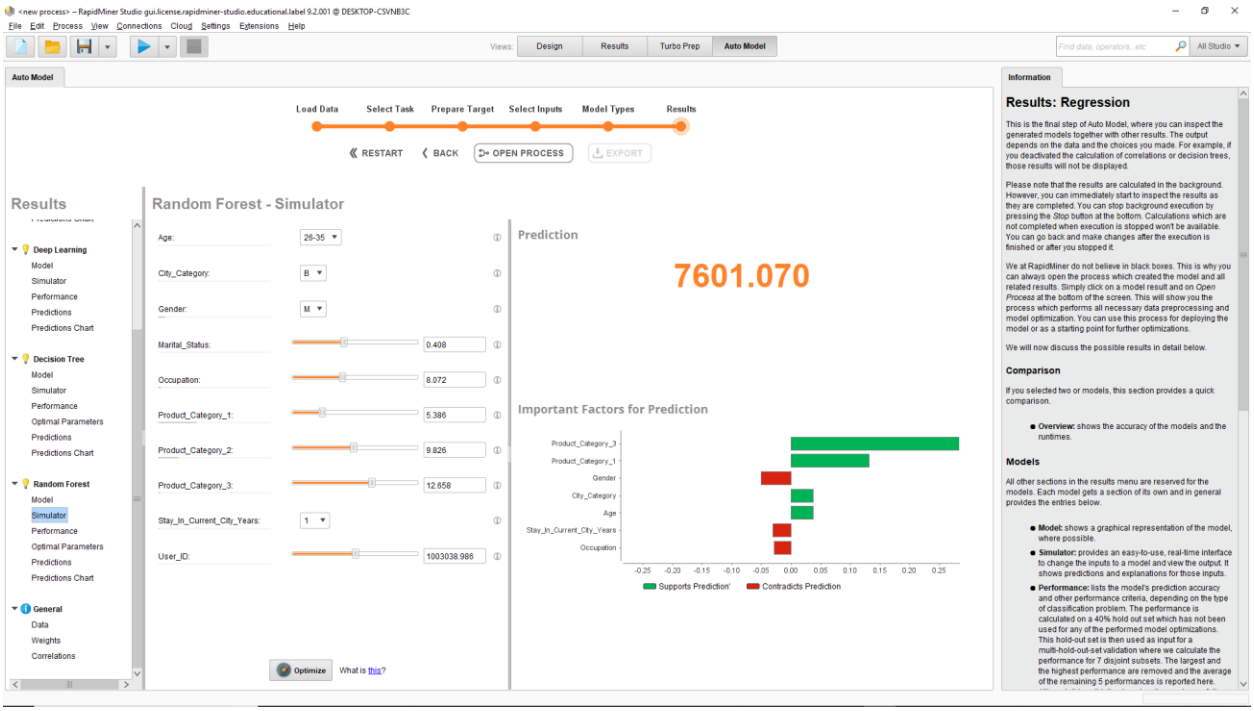

<new process> – RapidMiner Studio gui.license.rapidminer-studio.educational.label 9.2.001 @ DESKTOP-CSVNB3C
File Edit Process View Connections Cloud Settings Extensions Help
Views: Design Results Turbo Prep Auto Model
Auto Model
Load Data Select Task Prepare Target Select Inputs Model Types Results
RESTART BACK OPEN PROCESS EXPORT
Results
Comparison
Overview
Generalized Linear Model
Model
Simulator
Performance
Predictions
Predictions Chart
Deep Learning
Decision Tree
Random Forest
General
Data
Weights
Correlations
Generalized Linear Model - Predictions Chart
Predictions
True Values
Information
Results: Regression
This is the final step of Auto Model, where you can inspect the generated models together with other results. The output depends on the data and the choices you made. For example, if you deactivated the calculation of correlations or decision trees, those results will not be displayed.
Please note that the results are calculated in the background. However, you can immediately start to inspect the results as they are completed. You can stop background execution by pressing the Stop button at the bottom. Calculations which are not completed when execution is stopped won't be available. You can go back and make changes after the execution is finished or after you stopped it.
We at RapidMiner do not believe in black boxes. This is why you can always open the process which created the model and all related results. Simply click on a model result and on Open Process at the bottom of the screen. This will show you the process which performs all necessary data preprocessing and model optimization. You can use this process for deploying the model or as a starting point for further optimizations.
We will now discuss the possible results in detail below.
Comparison
If you selected two or models, this section provides a quick comparison.
Overview: shows the accuracy of the models and the runtimes.
Models
All other sections in the results menu are reserved for the models. Each model gets a section of its own and in general provides the entries below.
Model: shows a graphical representation of the model, where possible.
Simulator: provides an easy-to-use, real-time interface to change the inputs to a model and view the output. It shows predictions and explanations for those inputs.
Performance: lists the model's prediction accuracy and other performance criteria, depending on the type of classification problem. The performance is calculated on a 40% hold out set which has not been used for any of the performed model optimizations. This hold-out set is then used as input for a multi-hold-out-set validation where we calculate the performance for 7 disjoint subsets. The largest and the highest performance are removed and the average of the remaining 5 performances is reported here.

Decision Tree Model:

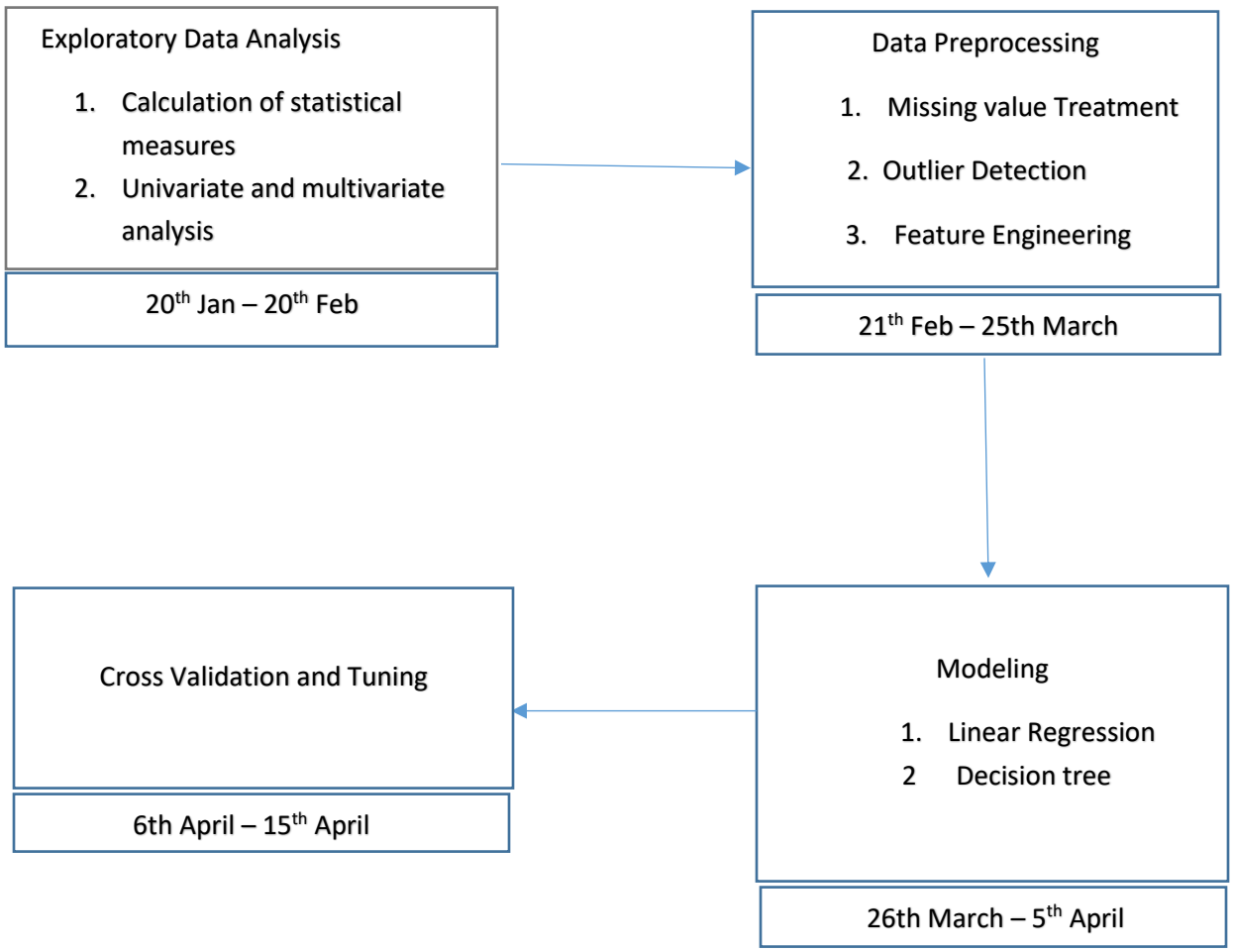




Random Forest Model:



PERT CHART:



System Requirements:-

Operating System: Windows, Linux and Unix

Hardware Configuration:

1. Minimum:
   Processor: Intel Core i3 (5th generation and above)

RAM: 2GB

2. Recommended:
   Processor: Intel Core i5 (5th generation and above)
   RAM: 8GB
   Graphics Processor(optional)

References:-

1. https://blog.hubspot.com/service/customer-behavior-analysis
2. https://www.optimove.com/resources/learning-center/customer-behavior-modeling
3. https://www.lsbf.org.uk/blog/news/analyse-consumer-behaviour/121862

APPROVED BY

(Name & Sign)	**(Name & Sign)**
Project Guide	**Program Head**