AN INVESTIGATION ON VIETNAMESE CREDIT SCORING BASED ON BIG DATA PLATFORM AND ENSEMBLE LEARNING

Author:

Quang-Linh Tran - 18520997

Van-Binh Duong - 18520505

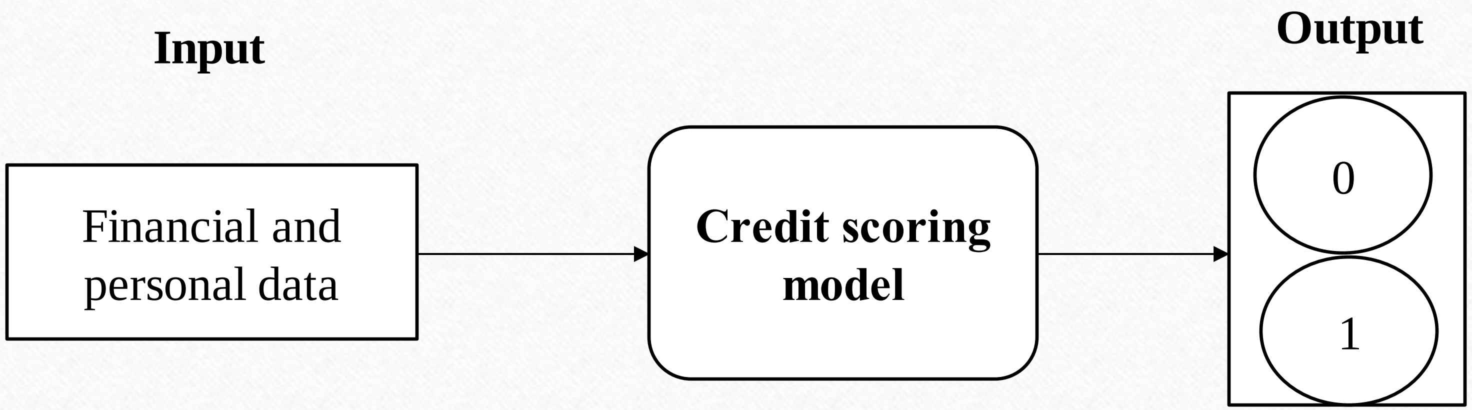
Gia-Huy Lam - 18520832

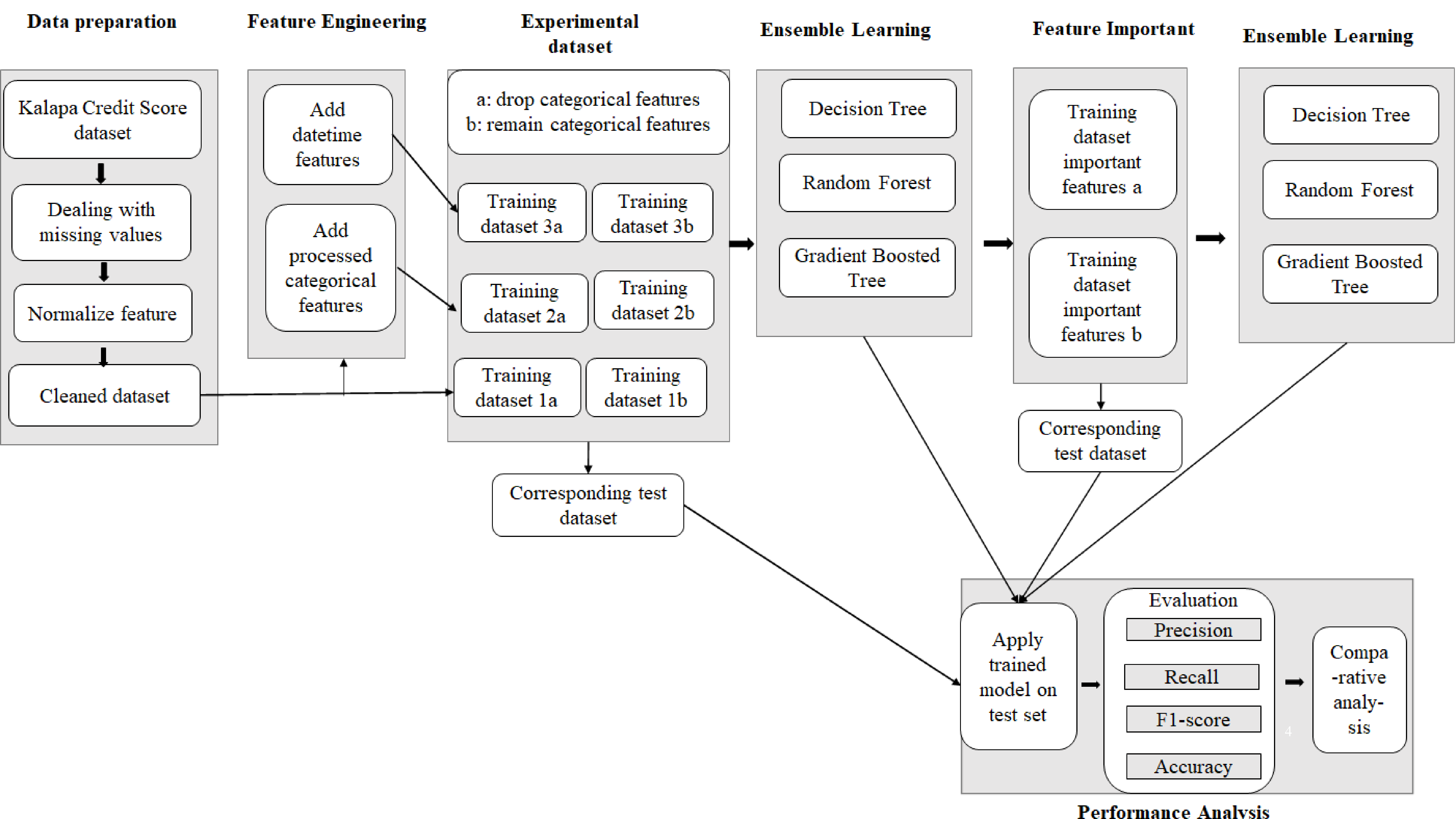
Advisor: PhD. *Trong-Hop Do*

Table of Content

1. INTRODUCTION

- Credit score: an important indicator for everyone.
- Evaluating manually: time-consuming and ineffective.
- Financial data increasing continuously requires a big data platform to handle.
- Feature engineering and ensemble learning are used to build predictive models.





Data preparation
Feature Engineering
Experimental dataset
Ensemble Learning
Feature Important
Ensemble Learning
Kalapa Credit Score dataset
Dealing with missing values
Normalize feature
Cleaned dataset
Add datetime features
Add processed categorical features
a: drop categorical features
b: remain categorical features
Training dataset 3a
Training dataset 3b
Training dataset 2a
Training dataset 2b
Training dataset 1a
Training dataset 1b
Decision Tree
Random Forest
Gradient Boosted Tree
Training dataset important features a
Training dataset important features b
Decision Tree
Random Forest
Gradient Boosted Tree
Corresponding test dataset
Corresponding test dataset
Apply trained model on test set
Evaluation
Precision
Recall
F1-score
Accuracy
Compa-rative analy-sis
Performance Analysis

2. RELATED WORK

Machine Learning-Based Empirical Investigation For Credit Scoring In Vietnam's Banking

(Khanh Quoc Tran et al)

- Kalapa Credit Score dataset
- Using machine learning models
- 83% F1-score with Random Forest

A comparative assessment of ensemble learning for credit scoring

(GangWang et al)

- Australia, Germany, China credit dataset.
- Using bagging, boosting, and stacking
- 80.76% accuracy

Credit scoring in the age of Big Data - A State-of-the-Art

(Youssef Tounsi et al.)

- Use social data instead of traditional financial data to evaluate credit score.
- Survey on proposed methods are given to address this problem
- Apache Hadoop and ApacheSpark are considered to use.

3. METHODOLOGIES

A. Big data platform

Big Data

5 V's

- Volume
- Variety
- Value
- Vecolity
- Veracity

How to utilize big data ?

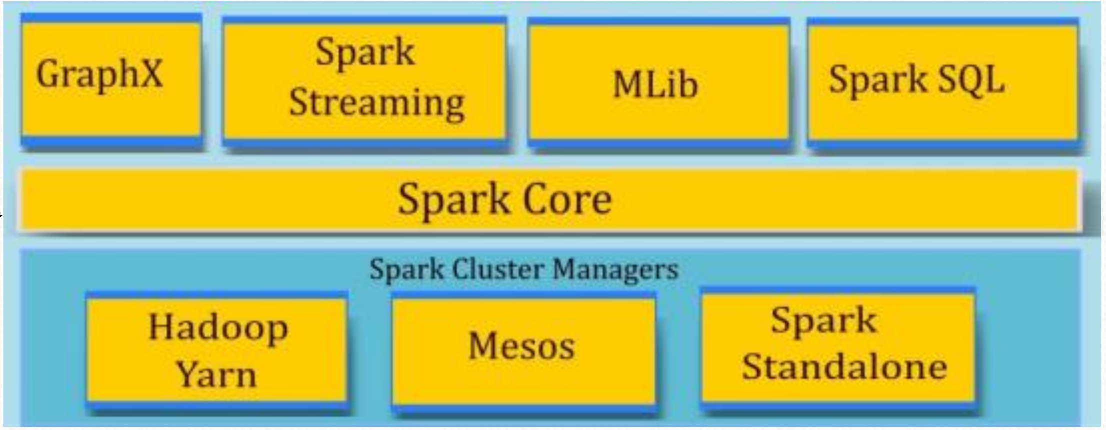


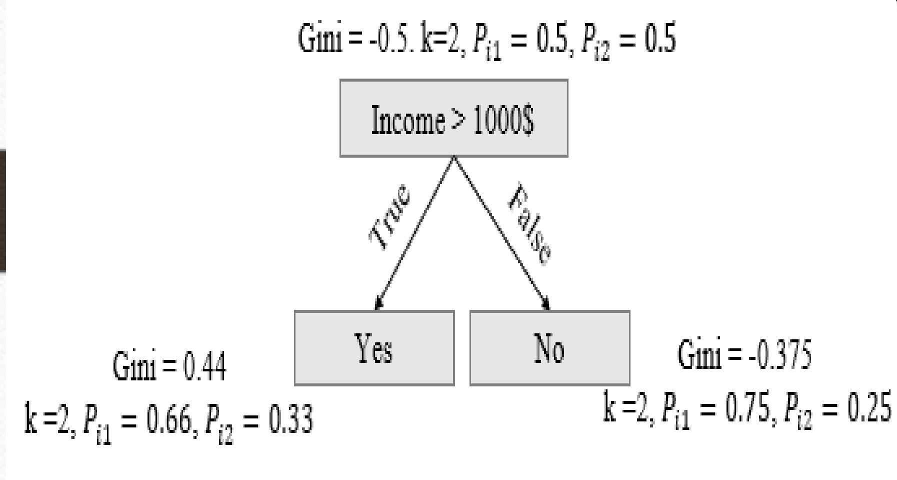
Fig1: Spark structure

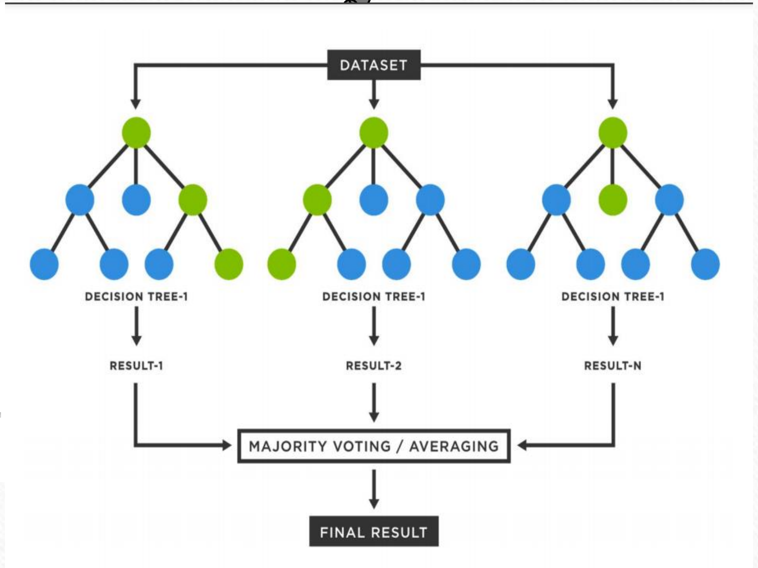
3. METHODOLOGIES

B. Ensemble Learning

Bagging

Boosting





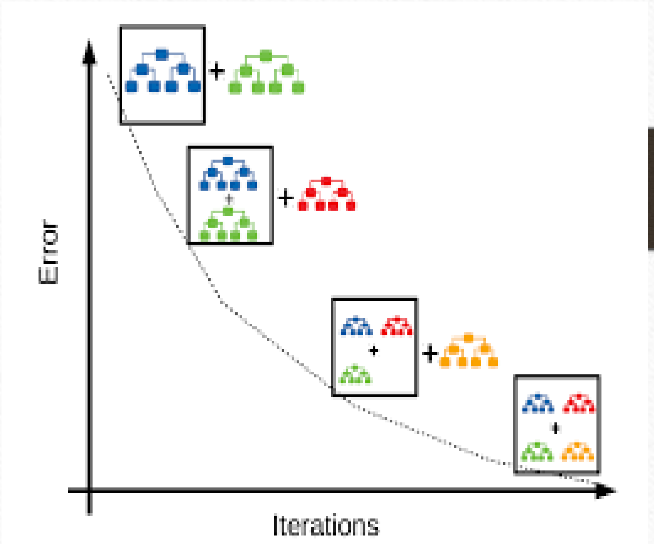


Fig2: Decision Tree

Fig3: Random Forest

Fig4: Gradient Boosted Trees

3. METHODOLOGIES

C. Feature Importance Extraction

- The higher the value of probability, the more important the feature
- Features extracted from best tree-based model
- Depend on the probability → Chooses important features

$$f_i = \sum_{j}^{k} s_j C_j,$$

– f_i: the probability of important feature i
– s_j: number of samples reaching node j
– C_j: the impurity value of node j
– k: nodes j splits on features i

Fig5: Formula of probability of important features

4. EXPERIMENT

A. Dataset and processing

Kalapa Credit Score:

- 193 attributes (117 attributes, missing rate >= 50%)
- Remove columns with >= 90% missing rate
- 53030 rows (68% label 0)

Attribute group	# of attributes	Processing
Date and datetime	28	Correct datatype and format
Unicode ones	30	Normalize the values
The rest	135	----------------------------

['Field_34', 'ngaySinh'] -> “%Y%m”
["Field_{}".format(i) for i in [1, 2, 43, 44]] -> "%Y-%m-%dT%H:%M:%S"
["Field_{}".format(i) for i in [5, 6, 7, 8, 9, 11, 15, 25, 32, 33, 35, 40]] -> "%Y-%m-%d"

"Zero" → 0, "One" → 1, "Two" → 2, "Three" -> 3, "Four" → 4
'thành phố Hà Nội' or “Ha nội city” → “hà nội”

4. EXPERIMENT

B. Feature Engineering

Generating new features:

- Datetime: **DT_A_B** = Field_A - Field_B (seconds)
- Date:
  + **DT_C_D** = Field_C - Field_D (days)
  + **days_from_now** = current processing date – Field_X
  + **age**: 2021 – year('ngaySinh')
  + **x_start_end** = x_srartDate – x_endDate
  + **x_y_startDate** = x_srartDate – y_startDate
  + **x_y_endDate** = x_endDate – y_endDate
  + weekend, weekday or not.
- Categoricals:

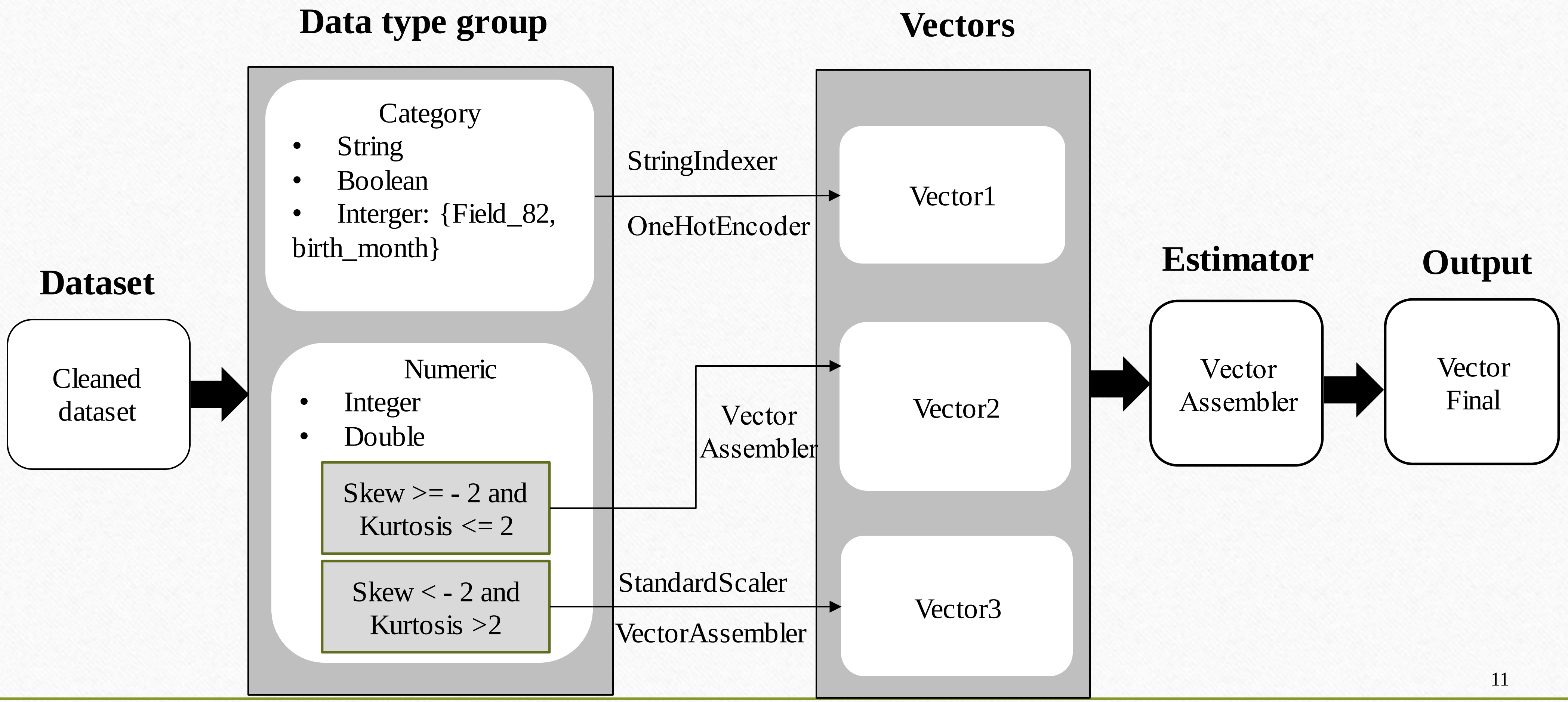
  gender = gioiTinh & info_social_sex

48 columns removed

81 columns added

4. EXPERIMENT

C. Input Preparation



4. EXPERIMENT

D. Most important features

GBT — **Data set 2**

Model → Feature importance → Most important features

Sparse Vector
Index: [1,3,..,180,182]
Weight: [0.02,0.03,…,0.07,0.05]
Sum_{weight} = 1

f_i >threshold
Threshold $= \frac{X}{Y}$
X in [0,5]
Y: length of index

X	**0**	**1**	**2**	**3**	**4**	**5**
Scenario A (# features)	82	23	13	9	6	4
Scenario B (# features)	62	20	10	5	2	1

4. EXPERIMENT

E. Experimental dataset statistics

Data set	# features in Scenario A (retaining raw categorical features)	# features in Scenario B (removing raw categorical features)
Data set1: Original	117	91
Data set2: feature engineering for datetime features	184	158
Data set3: data set 2 + feature engineering for categorical	196	168
Data set important features	23	20

5. RESULT

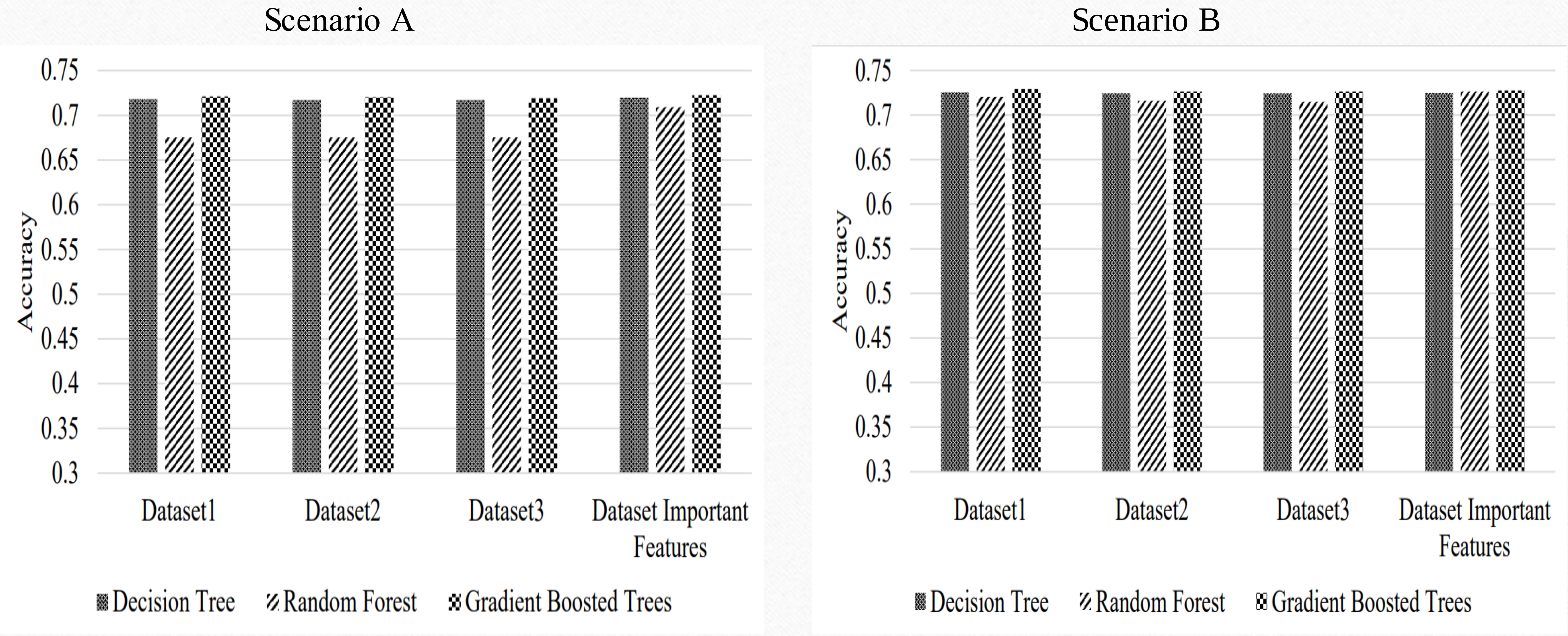
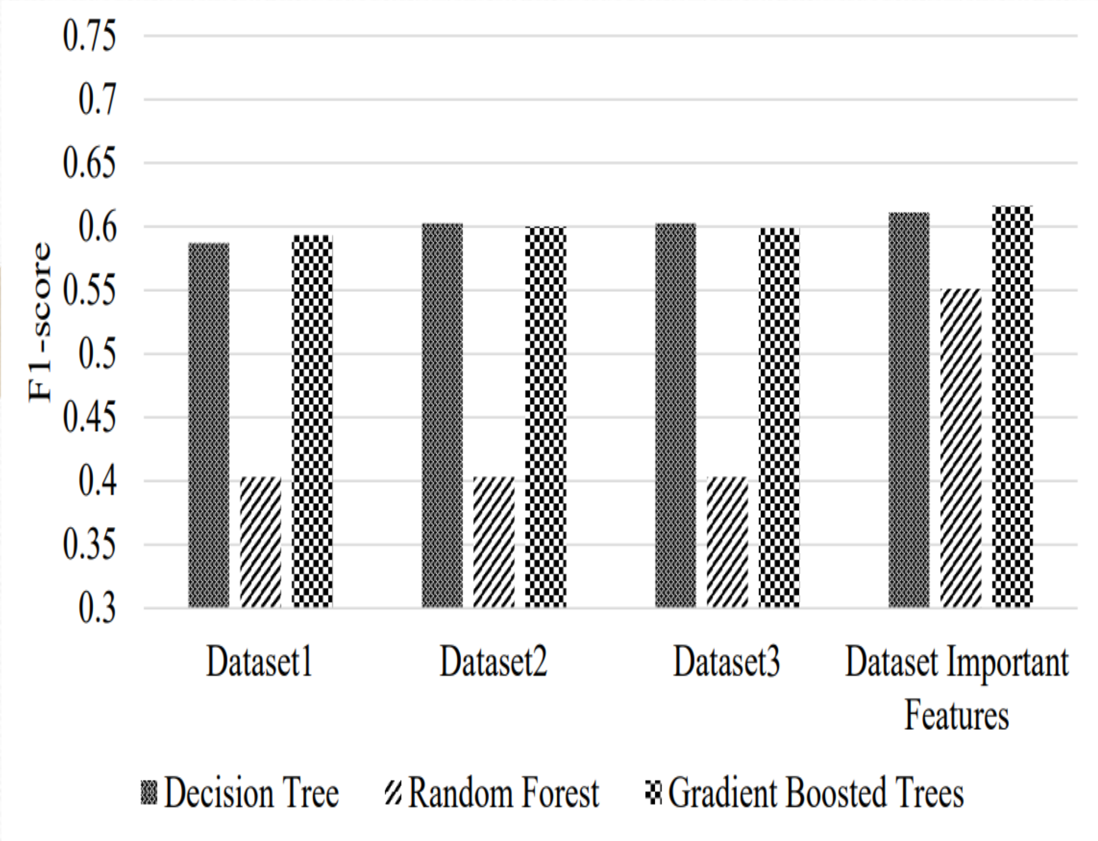


Fig6: Accuracy of different models on 4 data set

5. RESULT

Scenario A



Scenario B

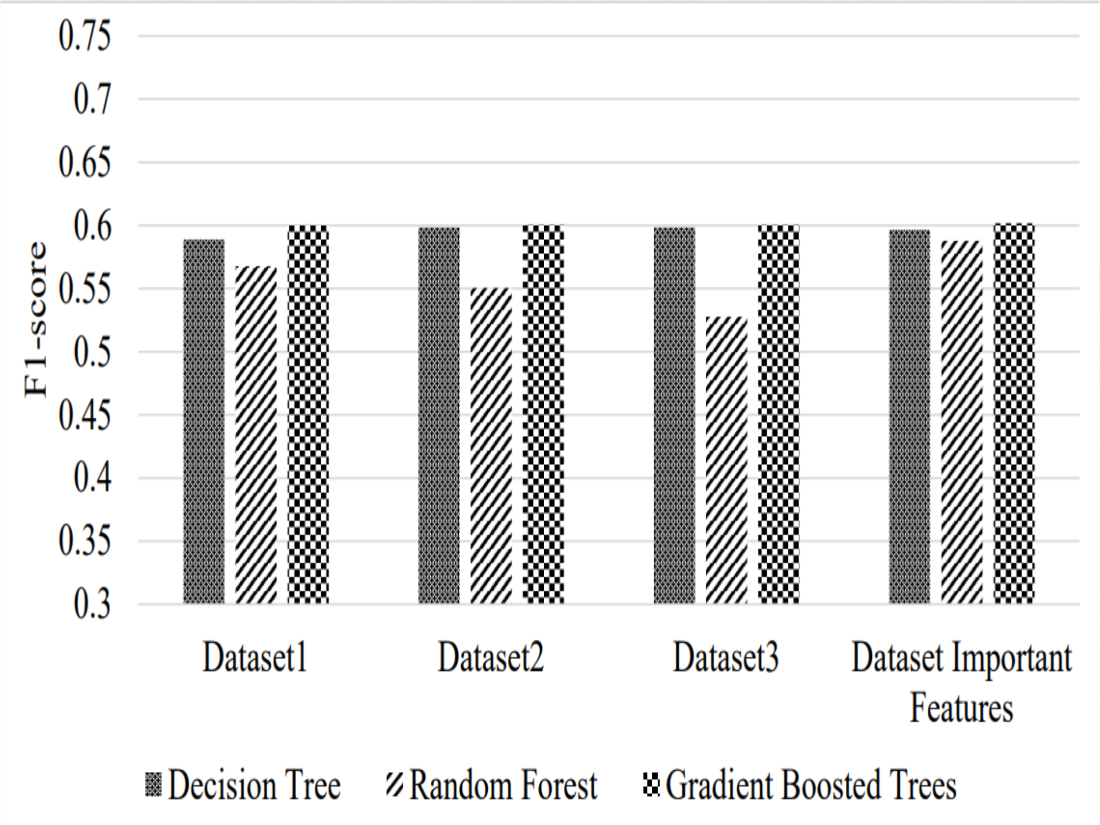
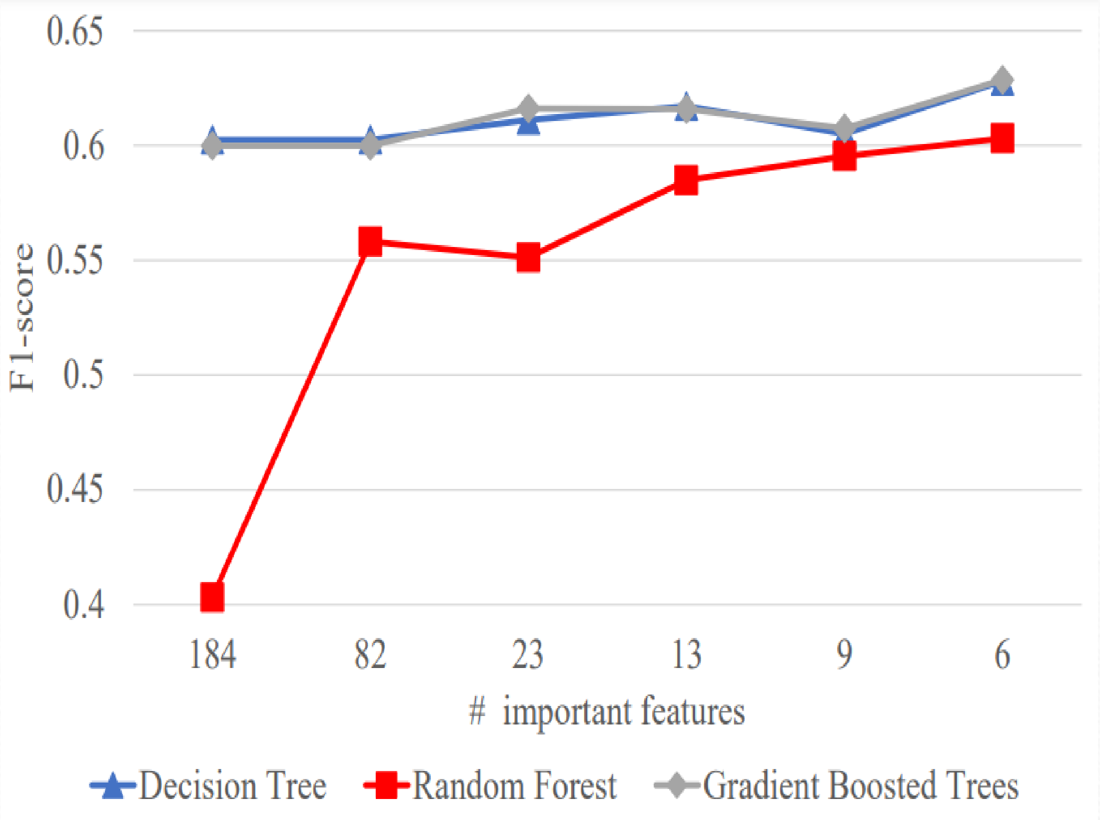


Fig7: F1-score of different models on 4 data set

5. RESULT

Scenario A



Scenario B

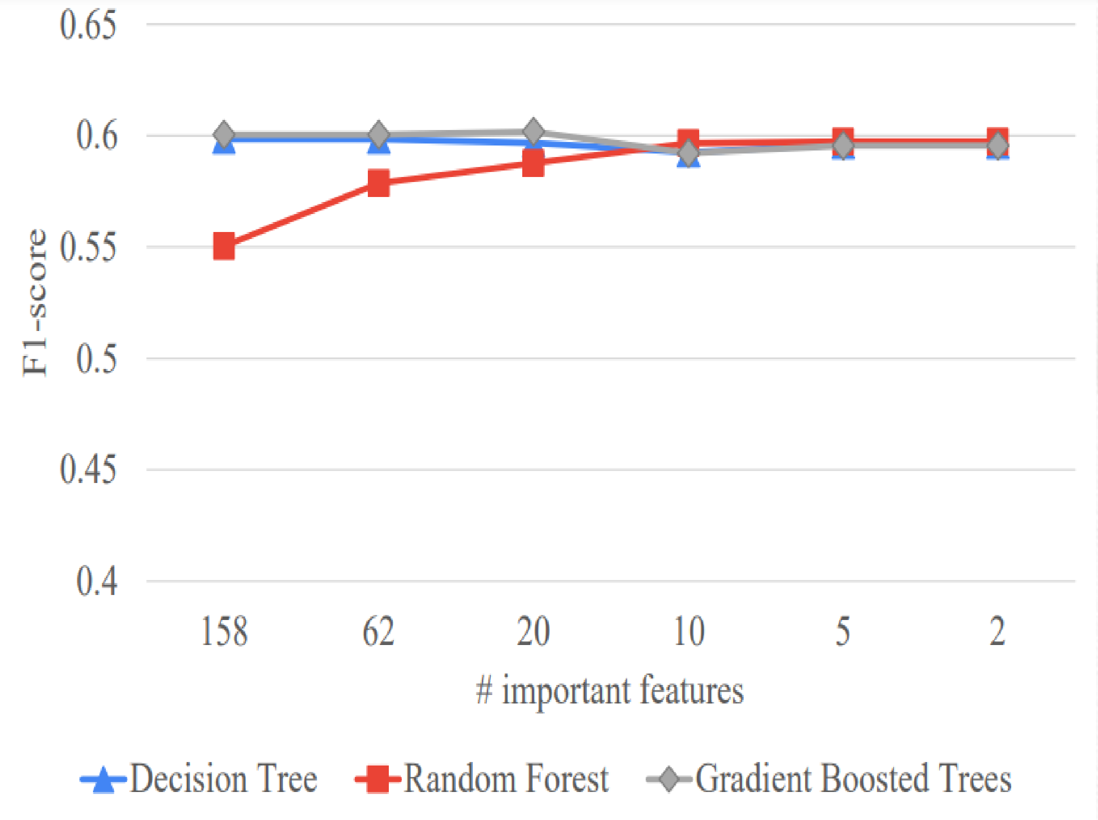


Fig8: F1-score on different numbers of important features in data set 2

5. CONCLUSION AND FUTURE WORK

Summary:

- Best performance: **60% F1_score and 72.92% Accuracy**
  - Scenario B of data set 1
  - GBT model
- Removing raw categorical features better than retaining them.
- Using important features improves efficiency
- Number of most important features in [5,10] gives the best f1-score.

Future work:

- Use streaming data (real-time financial activities or social data)
- Apply Deep Learning models

THANK YOU FOR WATCHING

ANN (512 -> 256 -> Dropout(0.2)->Output(2))
LR = 1e-06, Epochs = 100

Scenario A	Accuracy	Recall	Precision	F1 score
Other case (only dataset 2)	0.675411	0.337706	0.5	0.403132
6 features	0.716608	0.675549	0.621995	0.628072
4 features	0.699797	0.662109	0.570527	0.556345

Scenario B	Accuracy	Recall	Precision	F1 score
Other case (include 3 datasets)	0.675411	0.337706	0.5	0.403132
2 features	0.687973	0.628702	0.565617	0.554611
1 features	0.687973	0.628702	0.565617	0.554611

4. EXPERIMENT

E. Experiment scenarios

Dataset:

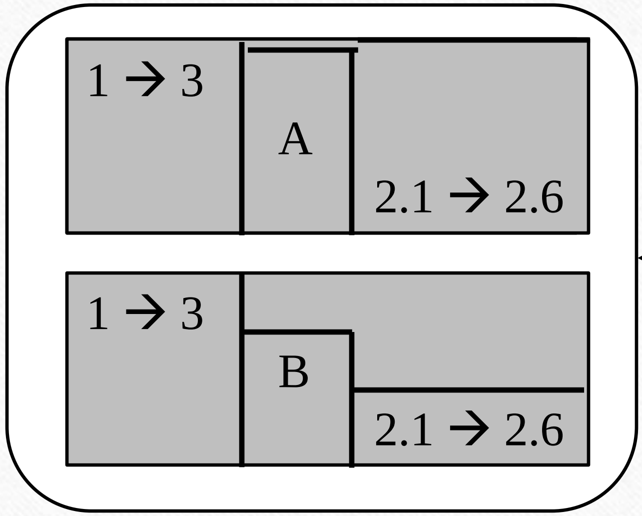
- Original (1)
- (1) + feature engineering for datetime features(2)
- (2) + feature engineering for categorical features(2)

Scenarios

- Remain raw categorical features (A)
- Drop raw categorical features (B)

Important features:

- f_i > threshold (x in [0,5])
- Total: 6 case (.1)->(.6)



Scenario A

Dataset	Model	Accuracy	Recall	Precision	F1 score
Data set1	Decision Tree	0.7183	0.5935	0.7068	0.587
	Random Forest	**0.6754**	**0.5**	**0.3377**	**0.4031**
	Gradient Boosted Tree	**0.721**	**0.5979**	**0.7114**	**0.5931**
Data set2	Decision Tree	0.717	0.6029	0.6888	0.6025
	Random Forest	0.6754	0.5	0.3377	0.4031
	Gradient Boosted Tree	0.7203	0.602	0.7017	0.5999
Data set3	Decision Tree	0.717	0.6029	0.6888	0.6025
	Random Forest	0.6754	0.5	0.3377	0.4031
	Gradient Boosted Tree	0.7192	0.601	0.6988	0.5988

Scenario B

Data set	Model	Accuracy	Precision	Recall	F1 score
data set1	Decision Tree	0.7254	0.593	0.7028	0.5891
	Random Forest	**0.7202**	**0.5791**	**0.7021**	**0.5677**
	Gradient Boosted Tree	**0.7292**	**0.6008**	**0.7075**	**0.6001**
data set2	Decision Tree	0.7244	0.5987	0.6921	0.5984
	Random Forest	0.7158	0.5685	0.6997	0.5504
	Gradient Boosted Tree	0.7263	0.6004	0.6968	0.6004
data set3	Decision Tree	0.7244	0.5987	0.6921	0.5984
	Random Forest	0.7147	0.5575	0.7271	0.5278
	Gradient Boosted Tree	0.7263	0.6004	0.6968	0.6004