Identify Drinking **from body signals**

By: Quoc-Trung Nguyen

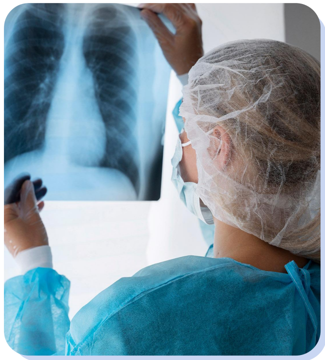
Introduction

Data source:
https://www.kaggle.com/datasets/sooyoungher/smoking-drinking-dataset

The data is general health checkup results from National Health Insurance Service in Korea.

Dimensions:
- 991346 rows
- 24 columns

Motivations:
- Learn more about medical terms
- Choose **the best classifier model** to predict drinking

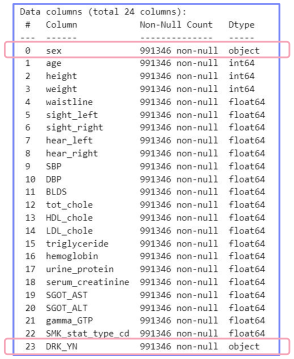


Patient medical profile (after cleaning)

Sex	**Male, Female**
Age (rounded)	**20 - 85** (years)
Height (rounded)	**130 - 190** (cm)
Weight (rounded)	**25 - 140** (kg)
Waistline	**27 - 149 .1** (cm)
Eye sight (L,R)	**0 - 2.5**
Hearing (L,R)	**normal, abnormal**
Smoking	**no, yes, used to**

SBP	67	273	[mmHg]
DBP	32	185	[mmHg]
BLDS	25	852	[mg/dL]
tot_chole	54	753	[mg/dL]
HDL_chole	1	20.9	[mg/dL]
LDL_chole	1	687	[mg/dL]
triglyceride	1	1273	[mg/dL]
hemoglobin	1	25	[g/dL]
urine_protein	1	6	category
serum_creatinine	0.1	24.4	[mg/dL]
SGOT_AST	1	1911	[IU/L]
SGOT_ALT	1	1940	[IU/L]
gamma_GTP	1	999	[IU/L]

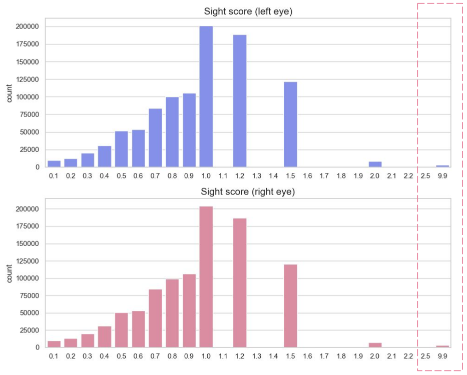
Data Quality Control - Type casting



```
Data columns (total 24 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   sex               991346 non-null  object
 1   age               991346 non-null  int64
 2   height            991346 non-null  int64
 3   weight            991346 non-null  int64
 4   waistline         991346 non-null  float64
 5   sight_left        991346 non-null  float64
 6   sight_right       991346 non-null  float64
 7   hear_left         991346 non-null  float64
 8   hear_right        991346 non-null  float64
 9   SBP               991346 non-null  float64
 10  DBP               991346 non-null  float64
 11  BLDS              991346 non-null  float64
 12  tot_chole         991346 non-null  float64
 13  HDL_chole         991346 non-null  float64
 14  LDL_chole         991346 non-null  float64
 15  triglyceride      991346 non-null  float64
 16  hemoglobin        991346 non-null  float64
 17  urine_protein     991346 non-null  float64
 18  serum_creatinine  991346 non-null  float64
 19  SGOT_AST          991346 non-null  float64
 20  SGOT_ALT          991346 non-null  float64
 21  gamma_GTP         991346 non-null  float64
 22  SMK_stat_type_cd  991346 non-null  float64
 23  DRK_YN            991346 non-null  object
```

Comments:
- No null value in the dataset
- Columns **sex** and **DRK_YN** need to be encoded:
  - → sex_dict = {'Male': 1, 'Female': 0}
  - → drink_dict = {'Y: 1, 'N': 0}

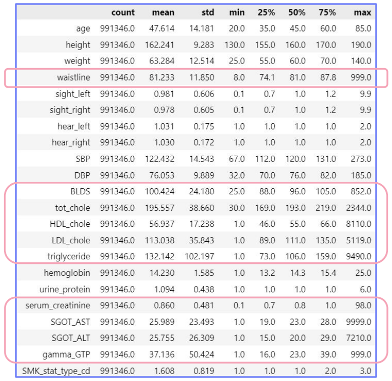
Data Quality Control - Check value



Comments:

- In sight_left and sight_right columns, value 9.9 means blind.
- This is not good for the model because the higher the score, the better vision
  ➔ change value 9.9 to 0

Data Quality Control - Outliers

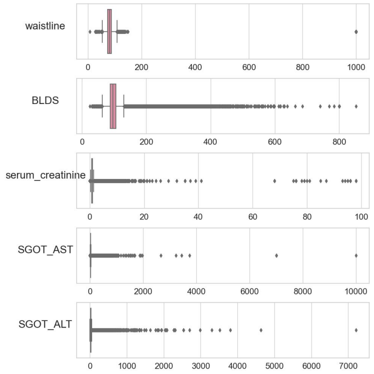


	count	mean	std	min	25%	50%	75%	max
age	991346.0	47.614	14.181	20.0	35.0	45.0	60.0	85.0
height	991346.0	162.241	9.283	130.0	155.0	160.0	170.0	190.0
weight	991346.0	63.284	12.514	25.0	55.0	60.0	70.0	140.0
waistline	991346.0	81.233	11.850	8.0	74.1	81.0	87.8	999.0
sight_left	991346.0	0.981	0.606	0.1	0.7	1.0	1.2	9.9
sight_right	991346.0	0.978	0.605	0.1	0.7	1.0	1.2	9.9
hear_left	991346.0	1.031	0.175	1.0	1.0	1.0	1.0	2.0
hear_right	991346.0	1.030	0.172	1.0	1.0	1.0	1.0	2.0
SBP	991346.0	122.432	14.543	67.0	112.0	120.0	131.0	273.0
DBP	991346.0	76.053	9.889	32.0	70.0	76.0	82.0	185.0
BLDS	991346.0	100.424	24.180	25.0	88.0	96.0	105.0	852.0
tot_chole	991346.0	195.557	38.660	30.0	169.0	193.0	219.0	2344.0
HDL_chole	991346.0	56.937	17.238	1.0	46.0	55.0	66.0	8110.0
LDL_chole	991346.0	113.038	35.843	1.0	89.0	111.0	135.0	5119.0
triglyceride	991346.0	132.142	102.197	1.0	73.0	106.0	159.0	9490.0
hemoglobin	991346.0	14.230	1.585	1.0	13.2	14.3	15.4	25.0
urine_protein	991346.0	1.094	0.438	1.0	1.0	1.0	1.0	6.0
serum_creatinine	991346.0	0.860	0.481	0.1	0.7	0.8	1.0	98.0
SGOT_AST	991346.0	25.989	23.493	1.0	19.0	23.0	28.0	9999.0
SGOT_ALT	991346.0	25.755	26.309	1.0	15.0	20.0	29.0	7210.0
gamma_GTP	991346.0	37.136	50.424	1.0	16.0	23.0	39.0	999.0
SMK_stat_type_cd	991346.0	1.608	0.819	1.0	1.0	1.0	2.0	3.0

Comments:

- There are 10 columns with high different between quantile 75% and max value

Data Quality Control - Outliers



waistline : There are 57 records of value= 999.0 and 1 record of value=8. This is unrealistic -> remove these records

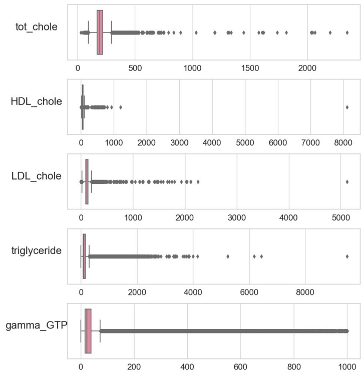
BLDS: There has been record on glucose in blood at > 1000, -> keep these values as rare cases

serum_creatinine: values that is higher than 5.0 are very critical and considered kidney damage. This can be the result of wrong input without fraction -> remove values > 30

SGOT_AST: The normal amount is between 20-40 IU/L. This can be a result of wrong inputs between U/L and IU/L -> remove records higher than 2000

SGOT_ALT: Same as SGOT_AST -> remove records higher than 2000

Data Quality Control - Outliers



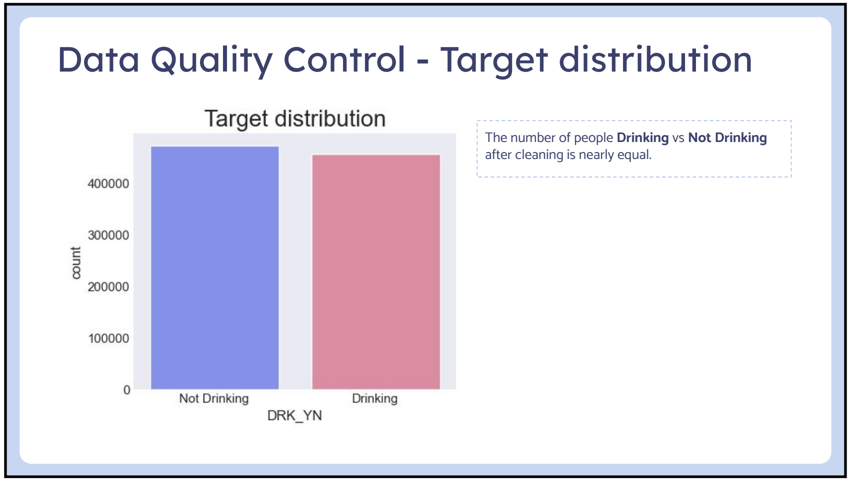
These 4 features are bounded by the formula:

HDL + LDL + 20% triglycerides = total_chole

Let sum_chole = HDL + LDL + 20% triglycerides, there are:
39702 rows with: sum_chole > total_chole + 1*
and 20905 rows with: sum_chole < total_chole - 1*
: ±1 is for rounding value control

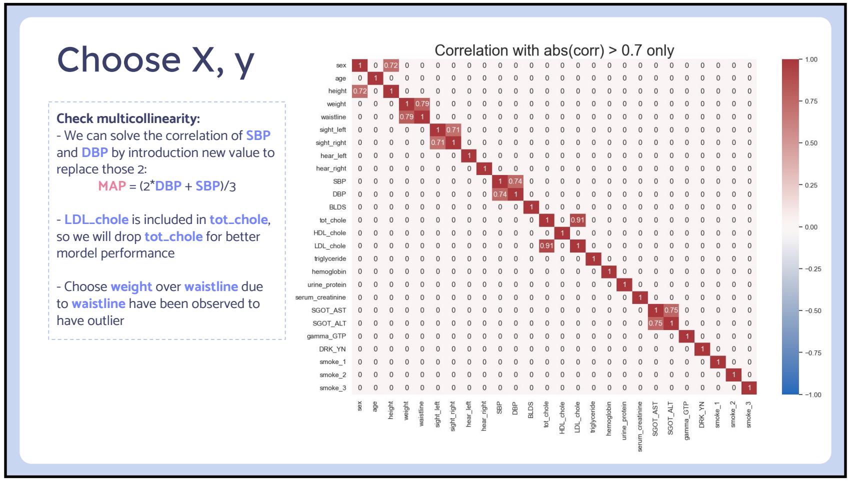
So we need to do 2 things here:
- Remove rows with wrong calculation
- Remove outliers that > 1000 mg/dL in tot_chole

gamma_GTP: Keeping them because it does not seem to be outliers



Data Quality Control - Target distribution

The number of people **Drinking** vs **Not Drinking** after cleaning is nearly equal.



Choose X, y

Check multicollinearity:
- We can solve the correlation of SBP and DBP by introduction new value to replace those 2:

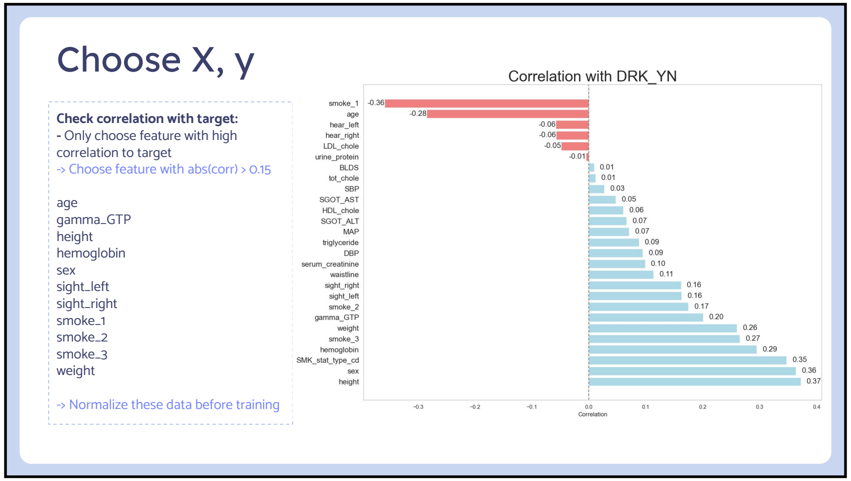
MAP = (2*DBP + SBP)/3

- LDL_chole is included in tot_chole, so we will drop tot_chole for better mordel performance

- Choose weight over waistline due to waistline have been observed to have outlier

Correlation with abs(corr) > 0.7 only

	sex	age	height	weight	waistline	sight_left	sight_right	hear_left	hear_right	SBP	DBP	BLDS	tot_chole	HDL_chole	LDL_chole	triglyceride	hemoglobin	urine_protein	serum_creatinine	SGOT_AST	SGOT_ALT	gamma_GTP	DRK_YN	smoke_1	smoke_2	smoke_3
sex	1	0	0.72	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
age	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
height	0.72	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
weight	0	0	0	1	0.79	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
waistline	0	0	0	0.79	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
sight_left	0	0	0	0	0	1	0.71	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
sight_right	0	0	0	0	0	0.71	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
hear_left	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
hear_right	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SBP	0	0	0	0	0	0	0	0	0	1	0.74	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DBP	0	0	0	0	0	0	0	0	0	0.74	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BLDS	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
tot_chole	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0.91	0	0	0	0	0	0	0	0	0	0	0
HDL_chole	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
LDL_chole	0	0	0	0	0	0	0	0	0	0	0	0	0.91	0	1	0	0	0	0	0	0	0	0	0	0	0
triglyceride	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
hemoglobin	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
urine_protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
serum_creatinine	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
SGOT_AST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.75	0	0	0	0	0
SGOT_ALT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.75	1	0	0	0	0	0
gamma_GTP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
DRK_YN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
smoke_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
smoke_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
smoke_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

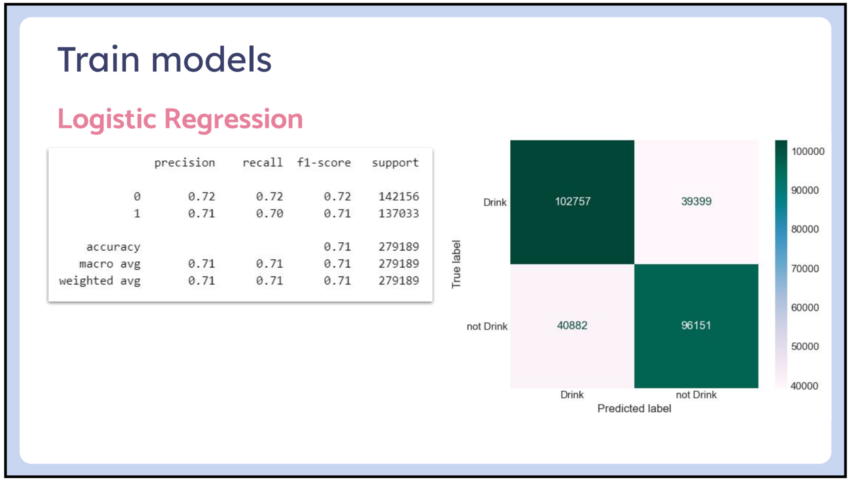


Choose X, y

Check correlation with target:
- Only choose feature with high correlation to target
-> Choose feature with abs(corr) > 0.15

age
gamma_GTP
height
hemoglobin
sex
sight_left
sight_right
smoke_1
smoke_2
smoke_3
weight

-> Normalize these data before training



Train models

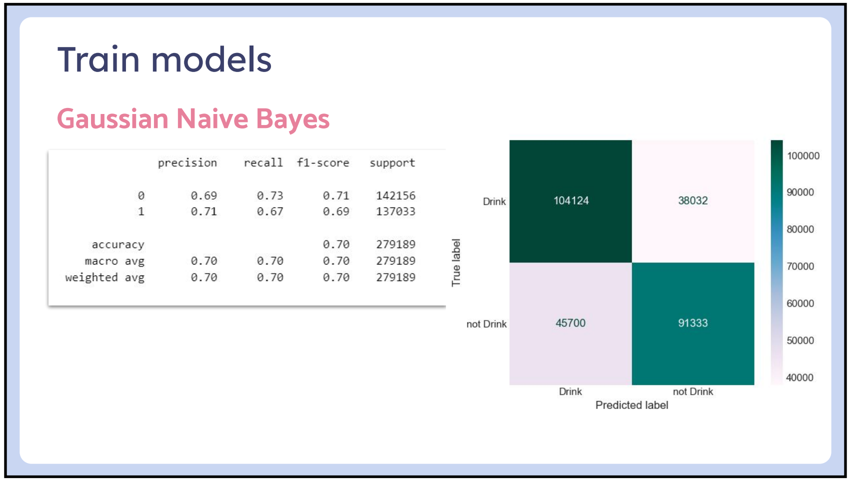
Logistic Regression

```
              precision    recall  f1-score   support

           0       0.72      0.72      0.72    142156
           1       0.71      0.70      0.71    137033

    accuracy                           0.71    279189
   macro avg       0.71      0.71      0.71    279189
weighted avg       0.71      0.71      0.71    279189
```

True label \ Predicted label	Drink	not Drink
Drink	102757	39399
not Drink	40882	96151



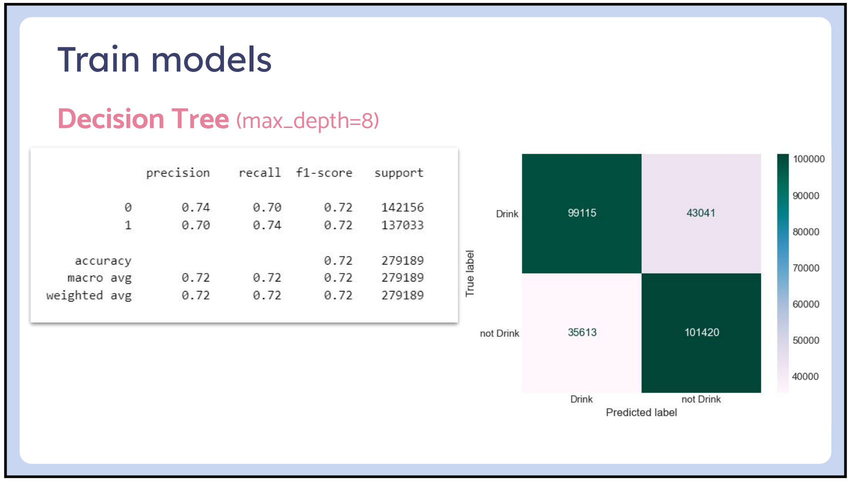
Train models

Gaussian Naive Bayes

```
              precision    recall  f1-score   support

           0       0.69      0.73      0.71    142156
           1       0.71      0.67      0.69    137033

    accuracy                           0.70    279189
   macro avg       0.70      0.70      0.70    279189
weighted avg       0.70      0.70      0.70    279189
```



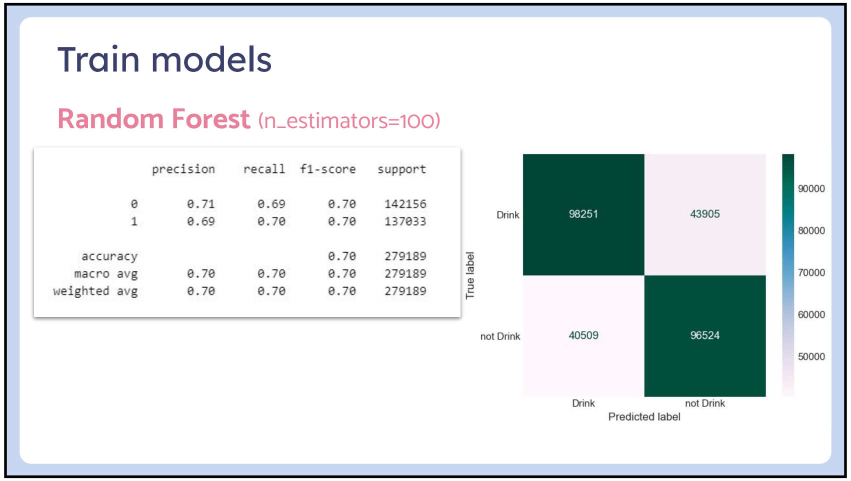
Train models

Decision Tree (max_depth=8)

```
              precision    recall  f1-score   support

           0       0.74      0.70      0.72    142156
           1       0.70      0.74      0.72    137033

    accuracy                           0.72    279189
   macro avg       0.72      0.72      0.72    279189
weighted avg       0.72      0.72      0.72    279189
```



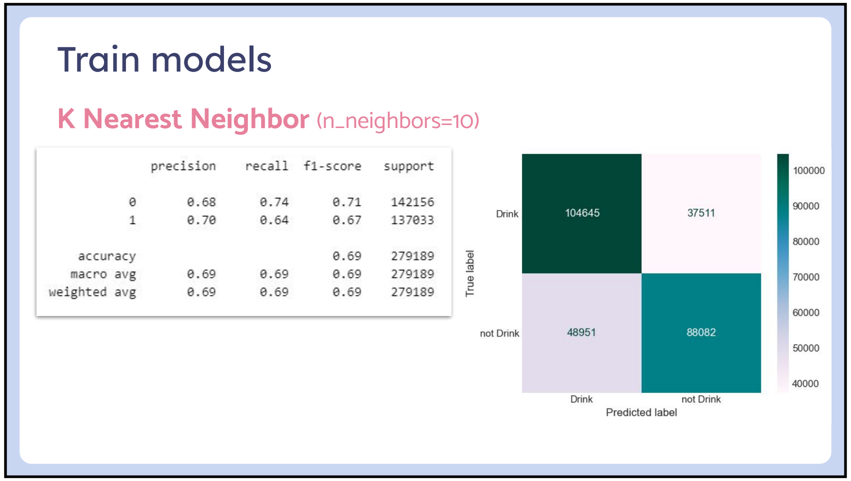
Train models

Random Forest (n_estimators=100)

```
              precision    recall  f1-score   support

           0       0.71      0.69      0.70    142156
           1       0.69      0.70      0.70    137033

    accuracy                           0.70    279189
   macro avg       0.70      0.70      0.70    279189
weighted avg       0.70      0.70      0.70    279189
```



Train models

K Nearest Neighbor (n_neighbors=10)

```
              precision    recall  f1-score   support

           0       0.68      0.74      0.71    142156
           1       0.70      0.64      0.67    137033

    accuracy                           0.69    279189
   macro avg       0.69      0.69      0.69    279189
weighted avg       0.69      0.69      0.69    279189
```

Compare Results

Comments:

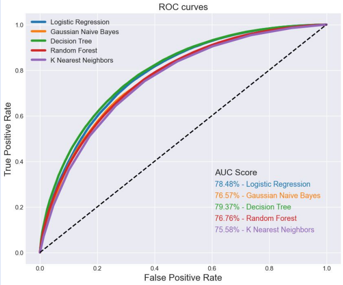
- **Logistic Regression** has best Precision
- While **Decision Tree** is having the best overall performance

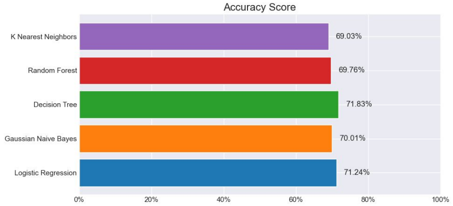
=> But why **Decision Tree** is having better scores than **Random Forest** here?
=> Result of Decision Tree model **might be overfitted**

For this dataset, it is recommended to use **Logistic Regression** for overall second best performance.

	Model	Params	TP	FN	TN	FP	Precision	Recall	F1_Score	Accuracy Score	AUC Score
0	Logistic Regression		102757	39399	40882	96151	0.7093	0.7017	0.7055	0.7124	0.7848
1	Gaussian Naive Bayes		104124	38032	45700	91333	0.7060	0.6665	0.6857	0.7001	0.7657
2	Decision Tree	max_depth = 8	99115	43041	35613	101420	0.7021	0.7401	0.7206	0.7183	0.7937
3	Random Forest	n_estimators = 100	98251	43905	40509	96524	0.6874	0.7044	0.6958	0.6976	0.7676
4	K Nearest Neighbors	n_neighbors = 10	104645	37511	48951	88082	0.7013	0.6428	0.6708	0.6903	0.7558

Compare Results


ROC curves
Logistic Regression
Gaussian Naive Bayes
Decision Tree
Random Forest
K Nearest Neighbors
True Positive Rate
False Positive Rate
AUC Score
78.48% - Logistic Regression
76.57% - Gaussian Naive Bayes
79.37% - Decision Tree
76.76% - Random Forest
75.58% - K Nearest Neighbors


Accuracy Score
K Nearest Neighbors
Random Forest
Decision Tree
Gaussian Naive Bayes
Logistic Regression