Behaviour analysis and Classification
– SQL, Tableau, Python

Using SQL to analyse customer behaviour

Bank Customer Analysis: What are we trying to find out?

- Bank contains data about customer loans, repayments and status, as well as some customer demographics

Overall TASKs are:

1) Identify the good clients to whom we can offer other services
2) Identify the bad clients that have to watch carefully to minimize the losses

- Can I use my data analysis toolkit to analyse customer behaviour and start identifying good/ bad customers? Can I identify any problems that the Bank has in general ?

How do banks make money?

1. **Profits from debt interest -** When you deposit your money in a bank account, the bank uses that money to make loans to other people and businesses to whom they charge interest.

2. **Banking fees -** Another way banks make money is through regular or case-by-case fees, such as account maintenance fees, withdrawal / transfer fees or unapproved overdraft fees.

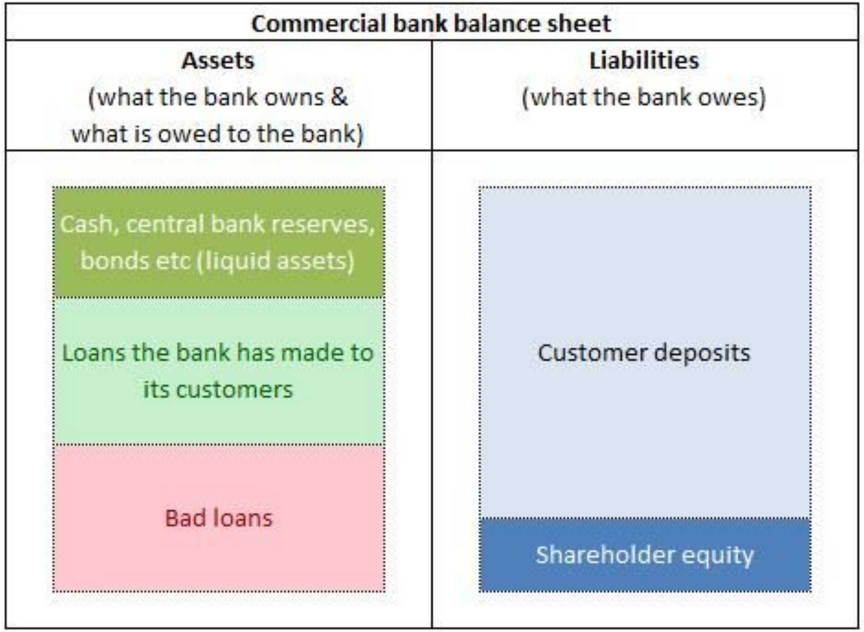
3. **Interchange fees** While swiping your debit or credit card is generally free to you, a transaction or processing fee called interchange is typically generated. This fee is charged by your bank to the merchant's bank (merchant being the store where you made the purchase) as a percentage of your transaction.

How do banks LOSE money? (/bank failure)

A bank fails when it can’t meet its financial obligations to creditors and depositors. This could occur because the bank in question has become insolvent, or because it no longer has enough liquid assets to fulfill its payment obligations.

The most common cause of bank failure occurs when the value of the bank’s assets falls to below the market value of the bank’s liabilities, which are the bank's obligations to creditors and depositors. This might happen because the bank loses too much on its investments (eg negative equity on mortgages) or has a larger than expected amount of loan defaults, plus loans being written off which it cannot absorb

How do banks LOSE money? (/bank failure)



With our bank database we do not have records of deposits, only loans, so we cannot establish the balance sheet

Banking Key Driver - Account Holder Churn

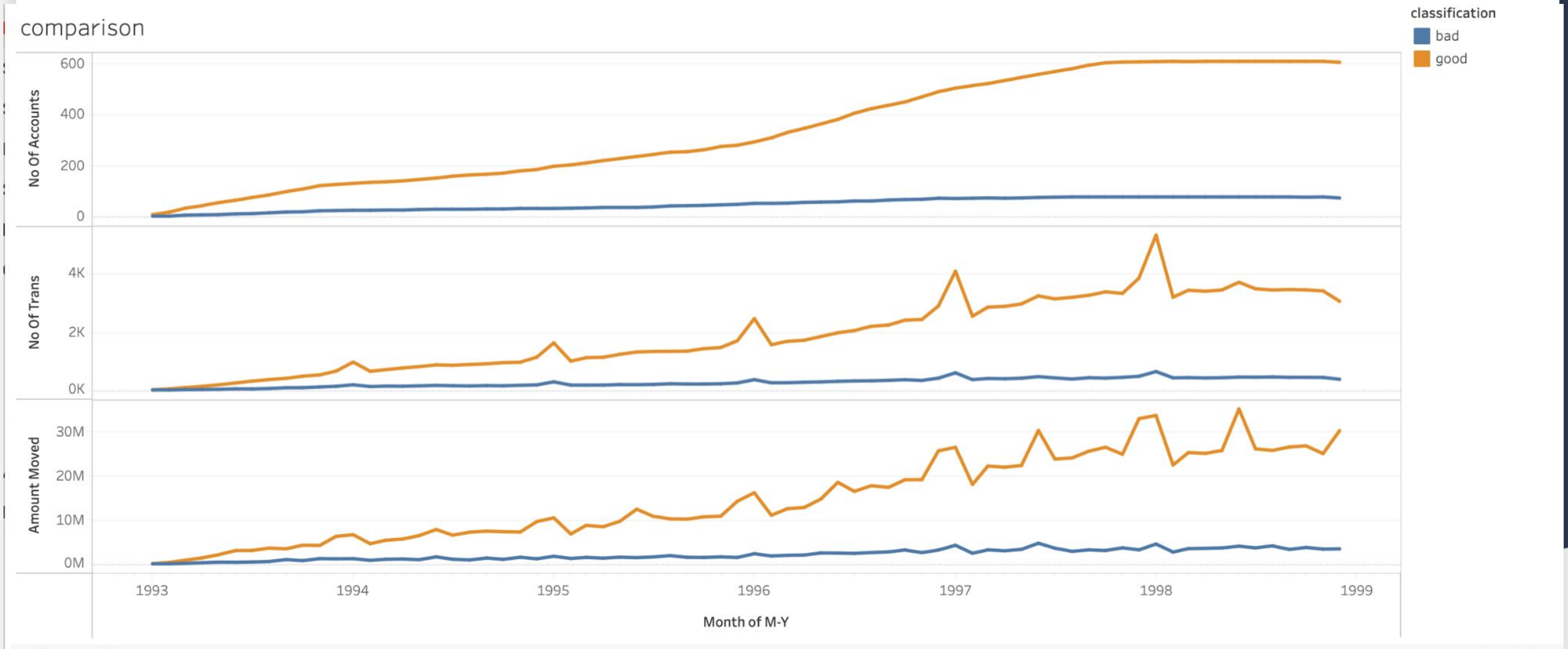
- Over time # accounts
- Over time # transactions
- Over time amount moved
- Over time # loans

+ Classify clients as good or bad based on their loan status (simplified)
+ Add this classification to a time series analysis

+ Is the bank churning bad or good clients?

Bank Churn Analysis: problem and approach (1)

- Problem # 1 = Customer Churn / attrition - *TIME SERIES*
- Let's find out over time, how many accounts/trans total we have which belong to 'good' or 'bad' clients
- Tool = **SQL aggregation with a WHERE filter**
- Approach - join Trans to Loans /Accounts (Disp) - using a grouping in tableau to flag bad/good customers based on Loan status and a subquery
  (or create a intermediary view with a bad/good cust flag)
- Then, **extract the data** as a csv and **visualise** the time series in Tableau


comparison
classification
bad
good
No Of Accounts
600
400
200
0
No Of Trans
4K
2K
0K
Amount Moved
30M
20M
10M
0M
1993
1994
1995
1996
1997
1998
1999
Month of M-Y

Review of group by Clause

Employee

EmployeeID	Ename	DeptID	Salary
1001	John	2	4000
1002	Anna	1	3500
1003	James	1	2500
1004	David	2	5000
1005	Mark	2	3000
1006	Steve	3	4500
1007	Alice	3	3500

SELECT *DeptID, AVG(Salary)*
FROM *Employee*
GROUP BY *DeptID;*

GROUP BY Employee Table using DeptID

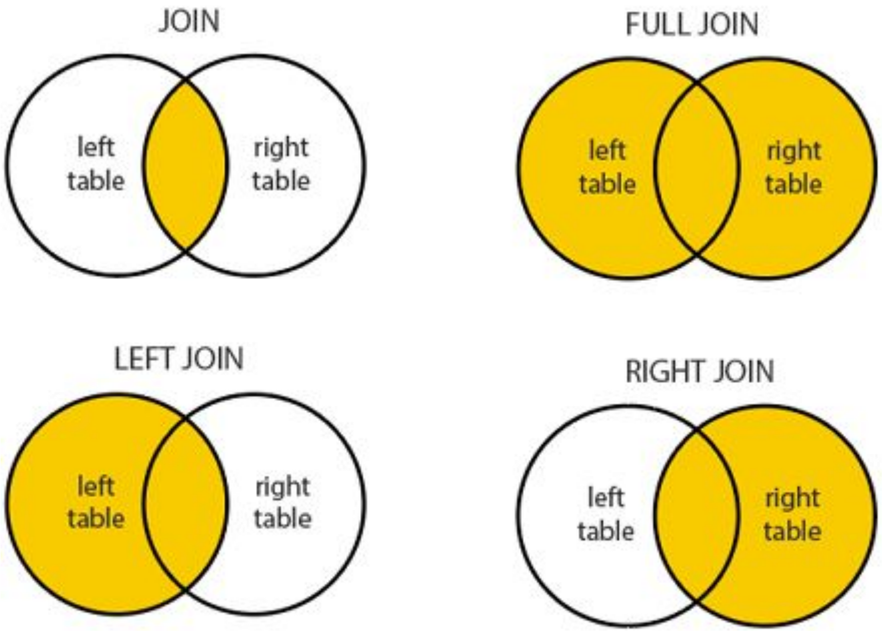
DeptID	AVG(Salary)
1	3000.00
2	4000.00
3	4250.00

SELECT *DeptID, AVG(Salary)*
FROM *Employee*
GROUP BY *DeptID*
HAVING *AVG(Salary) > 3000;*

HAVING

DeptID	AVG(Salary)
2	4000.00
3	4250.00

Review of SQL joins and subqueries



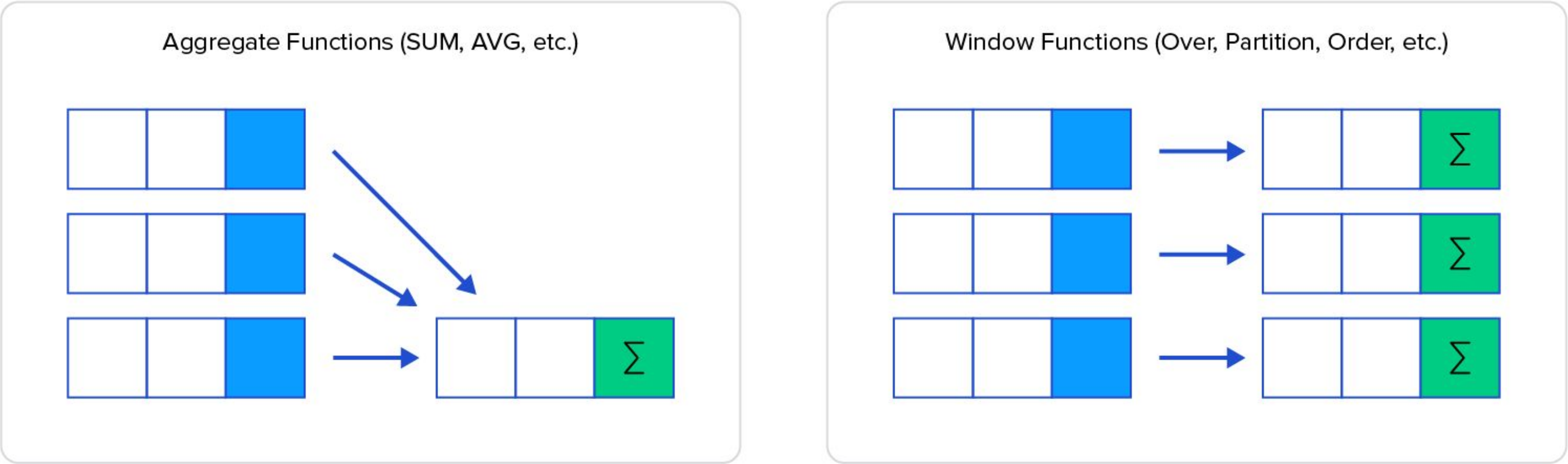

```sql
SELECT ProductID,
       Name,
       ListPrice
FROM   production.Product
WHERE  ListPrice > (SELECT AVG(ListPrice)
                    FROM   Production.Product)
```

subquery

Bank Churn Analysis: problem and approach (2)

- Problem # 1 = Customer Churn / attrition - *MoM distinct*
- Let's find out month to month, how many unique account holders were making transactions, as an indicator of churn.
- Tool = Window **LAG** function
- Approach - create **Views** for re-usability/ease of coding
- Then, **extract the data** as a csv and **visualise** the time series in Tableau

Review of Window functions



Review of Window functions

Select SUM(sale)
OVER (
Partition by fiscal_year)
As 'total_sales'
From salestable

fiscal_year	sales_employee	sale	total_sales
2016	Alice	150.00	450.00
2016	Bob	100.00	450.00
2016	John	200.00	450.00
2017	Alice	100.00	400.00
2017	Bob	150.00	400.00
2017	John	150.00	400.00
2018	Alice	200.00	650.00
2018	Bob	200.00	650.00
2018	John	250.00	650.00

Review of Window functions

- Window = “Partition” (PARTITION BY) + “Ordering” (ORDER BY) + “Frame” (ROW/RANGE)
- OVER defines the set where the function will be applied

OVER (
PARTITION BY -> Divides the set in “chunks” where the function is applied independently (similar to GROUP BY)
ORDER BY -> How to sort the elements within the partition
ROW or **RANGE** -> Set “boundaries” within each “chunk”

) **AS** [alias]

Window function in use - LAG

How the LAG() window function works

ID	value	sale_year	prev_value
1	125.00	2016	NULL
1	340.00	2017	125.00
1	340.00	2018	340.00
1	100.00	2019	340.00
2	80.00	2016	NULL
2	0.00	2017	80.00
2	120.00	2018	0.00
2	150.00	2019	120.00

Window frames:

			previous row with partition
2	80 00	2016	current row

2	80.00	2016	previous row with partition
2	0 00	2017	current row

2	0.00	2017	previous row with partition
2	120 00	2018	current row

2	120.00	2018	previous row with partition
2	150.00	2019	current row

Bank Churn Analysis: problem and approach (3)

- Problem # 1 = Customer Churn / attrition *STICKINESS*
- Let's find out how many customers keep doing their transactions at this bank in subsequent months- how many active accounts has the bank retained?
- Tool = **self join**
- Approach - create **View+CTE** for re-usability/ease of coding
- https://www.sisense.com/blog/use-self-joins-to-calculate-your-retention-churn-and-reactivation-metrics

Bank Churn Analysis: problem and approach (3)

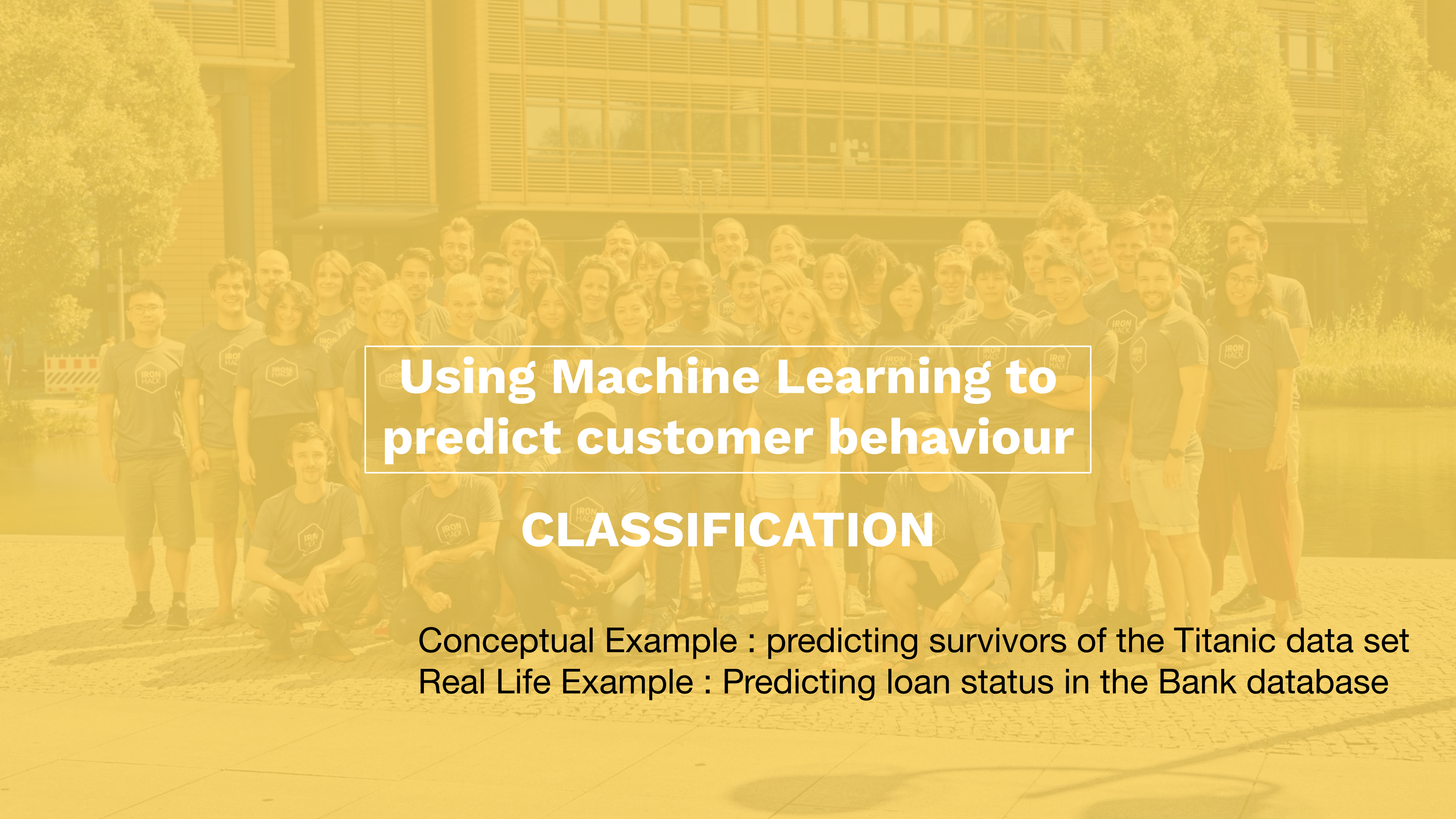
Table 1

Count dist account_id	Activity Year	Activity Month
93	1993	01
183	1993	02
281	1993	03

Table 2

Count dist account_id	Activity Year	Activity Month
93	1993	01
183	1993	02
281	1993	03

+ Calculate the Difference



Using Machine Learning to predict customer behaviour

CLASSIFICATION

Conceptual Example : predicting survivors of the Titanic data set
Real Life Example : Predicting loan status in the Bank database

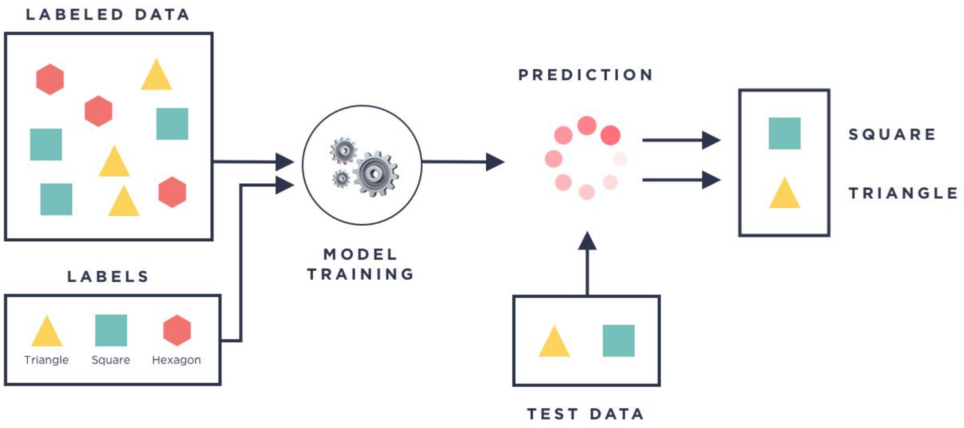
What data insights can we get from doing ML?

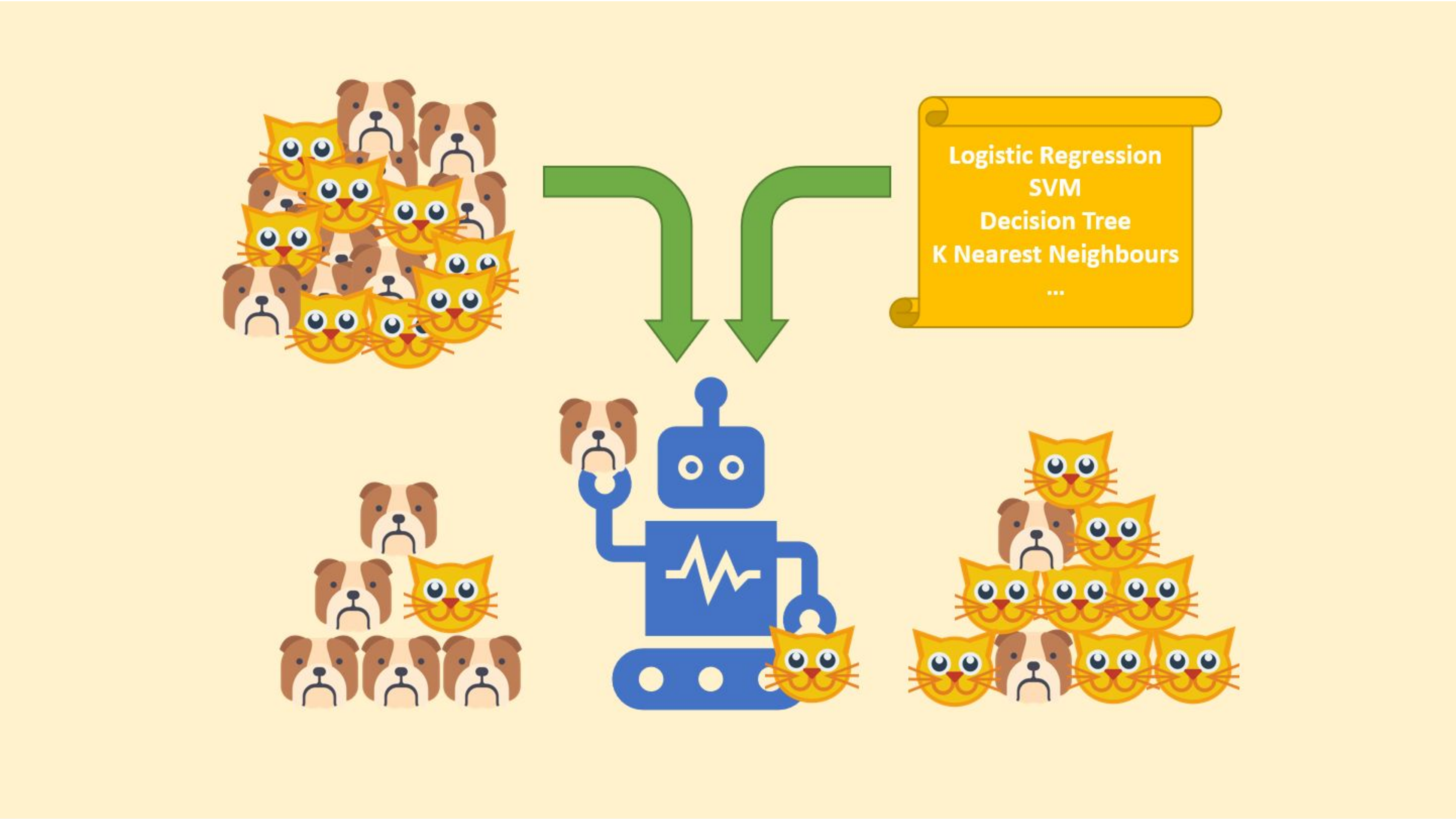
- Which features have the greatest impact on the outcome?
- Which features are irrelevant to the outcome?
- Do we have any unexpected outcomes?

- Can we change the marketing or product strategy to promote better health of the Bank lending book?
- Are there some demographics or behavioural profiles for customers we wish to attract or deter ?

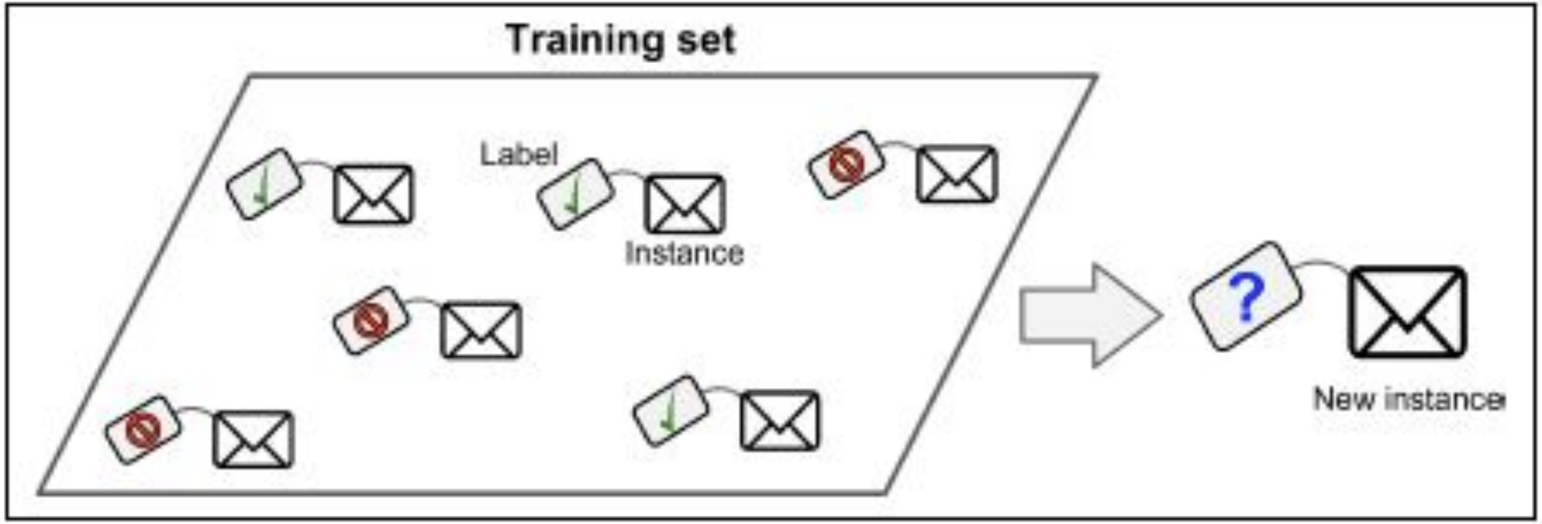
SUPERVISED MACHINE LEARNING MODELS

- Supervised machine learning models are all the model which needs to be trained based on examples:

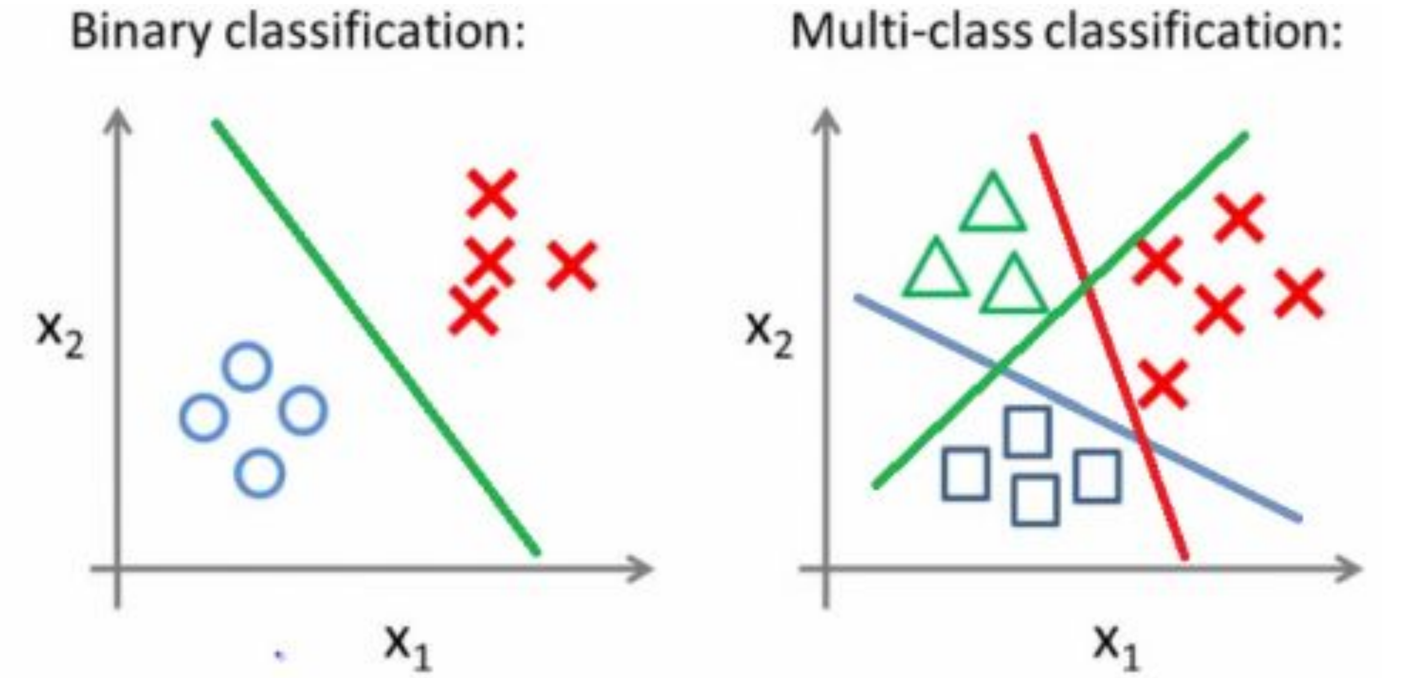



Logistic Regression
SVM
Decision Tree
K Nearest Neighbours
...

Logistic Regression - binary classification



Logistic Regression - binary v multi class



Logistic Regression - binary v multi class

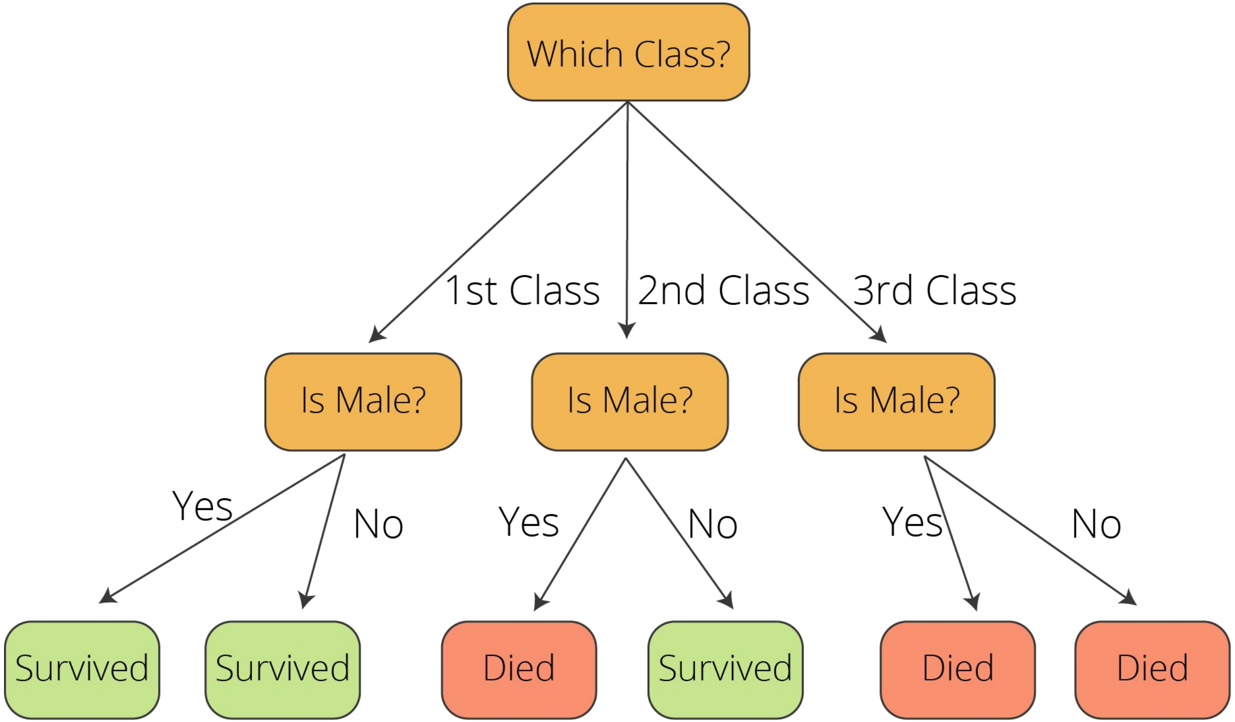
Binary classification:

In our Bank scenario our binary ML problem is :

Multi-class classification:

In our Bank scenario our multiclass ML problem is :

Logistic Regression - conceptual example



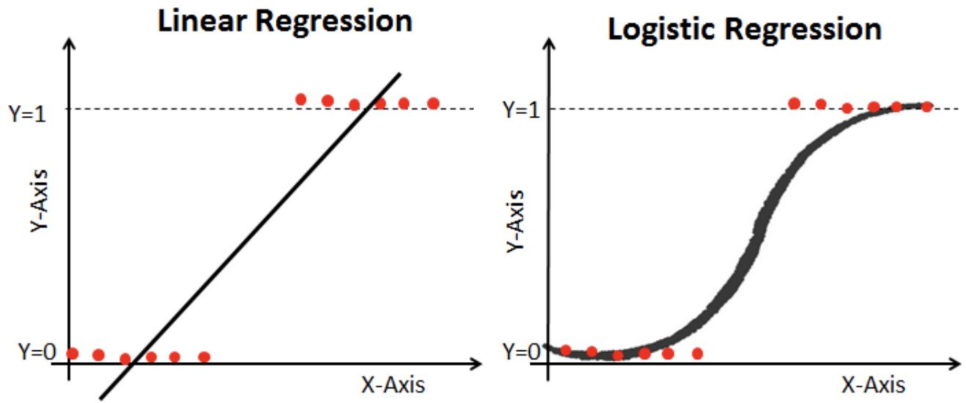
What is logistic regression?

Logistic regression is a technique for modelling the probability of an event.
Just like linear regression, it helps you understand the relationship between one or more variables and a target variable
Except that our target variable is binary: its value is either 0 or 1.

Example:

- It's possible to say that "smoking can increase your risk of having lung cancer by 20%", since having lung cancer is a binary variable: you either have it or not.
- From that, we can infer classification probability based on other factors
- like whether someone will have lung cancer given that he/she does not smoke, lives in a polluted city and has a family history of lung cancer.

Understanding logistic regression



Source: https://bit.ly/35MhQwg

To better understand the difference between a linear and a logistic regression, imagine we plotted the lung cancer variable in the Y-axis (Y = 1 if patient has lung cancer and 0 otherwise) and the patient's age in the X-axis.

Here we have the resulting lines from each regression. Which one seems more fit to our data?

When to use logistic regression?

- When linear regression makes no sense for our data - eg if we follow a line the linear approach would take us > 100%

Logistic ensures a curve cannot exceed 100% or go below 0

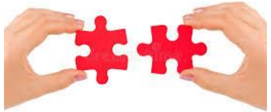
- Where response variable is a proportion
- When we want a model to predict probabilities of different possible outcomes of a categorically distributed response variable, given a set of independent variables

**NB dont confuse with logarithmic scale*

NO FREE-LUNCH THEOREM

- Every model makes assumptions about data. If those assumptions are not met, the model is doomed to fail.
- **Model assumptions determine the pre-processing steps required**.

Preproc
essing



Model

Preproces
sing

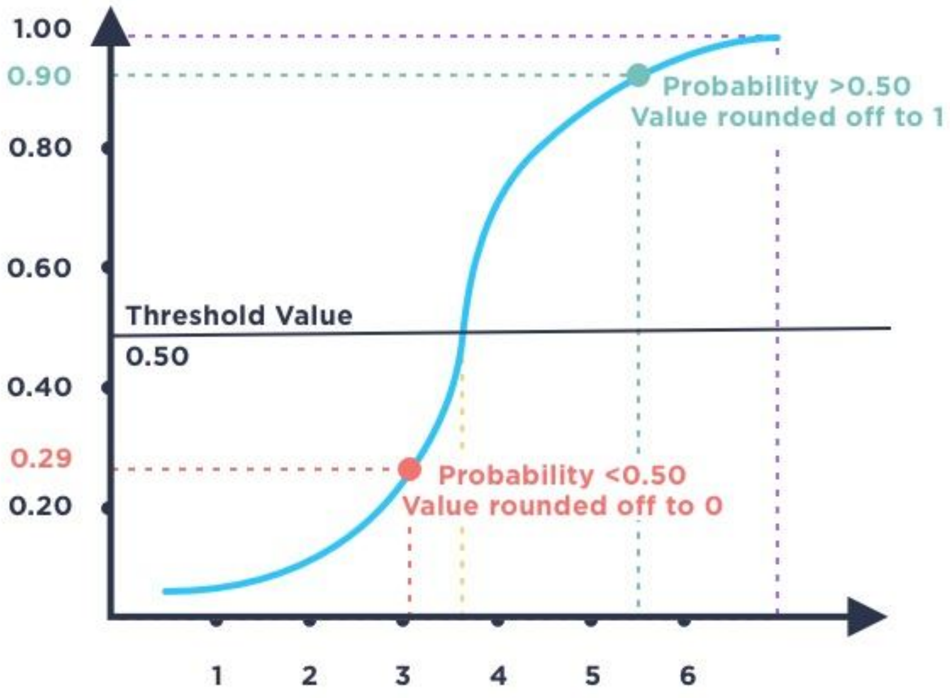
Model

LOGISTIC MODEL (for binary classification)

- The logistic model makes the following **assumptions**:

- **DOES NOT** require a **linear relationship between the dependent and independent variables.**
- The error terms (**residuals**) **DO NOT** need to be **normally distributed**.
- Independent observations (not repeated).
- **Multicollinearity NOT RELEVANT**.
- Dependent variable **MUST be categorical and can only have two possible values**. (yes/no, 1/o, True/False).
- Larger dataset, at least 10 cases for each possible target value.

DETERMINING THE PREDICTION OF THE LOGISTIC MODEL



Terminology

Regression Problem:

Predicting an amount

Target_D is the dependent variable.

The features are the independent variables.

Features | Labels

HV1	IC1	IC2	IC3	IC4	IC5	AVGGIFT	TARGET_D
2346	420	446	468	503	14552	15.5	21
497	350	364	357	384	11696	3.08	3
1229	469	502	507	544	17313	7.5	20
325	148	181	171	209	6334	6.7	5
768	174	201	220	249	7802	8.78571429	10
557	211	188	221	205	5550	13	16
2145	474	492	522	554	18340	11.5714286	15
2184	351	376	394	419	16480	12.5	20
1442	369	394	445	488	26462	7.84615385	10
1708	437	586	551	684	29098	9.76923077	20
1054	584	644	652	726	26074	13.5384615	20
1062	486	550	555	584	17908	15.3333333	20
849	457	508	470	519	16386	12.8	25
213	222	273	283	329	12227	5.125	5
574	289	318	315	363	11250	3.55555556	4
2506	449	455	501	517	16302	8.875	50
622	347	378	401	416	15808	15	25
764	272	361	346	424	16257	7.91304348	15
681	335	398	356	419	14011	30.75	51

Terminology

Classification Problem:

Predicting target variable that are labels

... in this case a binary (A or B, True or False, Yes or No, 1 or 0)

Features | Labels

	loan_id	account_id	date	amount	duration	payments	status
0	5314	1787	930705	96396	12	8033.0	B
1	5316	1801	930711	165960	36	4610.0	A
2	6863	9188	930728	127080	60	2118.0	A
3	5325	1843	930803	105804	36	2939.0	A
4	7240	11013	930906	274740	60	4579.0	A
5	6687	8261	930913	87840	24	3660.0	A
6	7284	11265	930915	52788	12	4399.0	A
7	6111	5428	930924	174744	24	7281.0	B
8	7235	10973	931013	154416	48	3217.0	A
9	5997	4894	931104	117024	24	4876.0	A
10	7121	10364	931110	21924	36	609.0	A
11	6077	5270	931122	79608	24	3317.0	A
12	6228	6034	931201	464520	60	7742.0	B
13	6356	6701	931208	95400	36	2650.0	A
14	5523	2705	931208	93888	36	2608.0	A
15	6456	7123	931209	47016	12	3918.0	A

A credit card company receives thousands of applications for new cards. Each application contains information about an applicant:

- Age
- Marital status
- Annual salary
- Outstanding debts
- Credit rating

Others

Typical
Classification Problem

ML Problem:

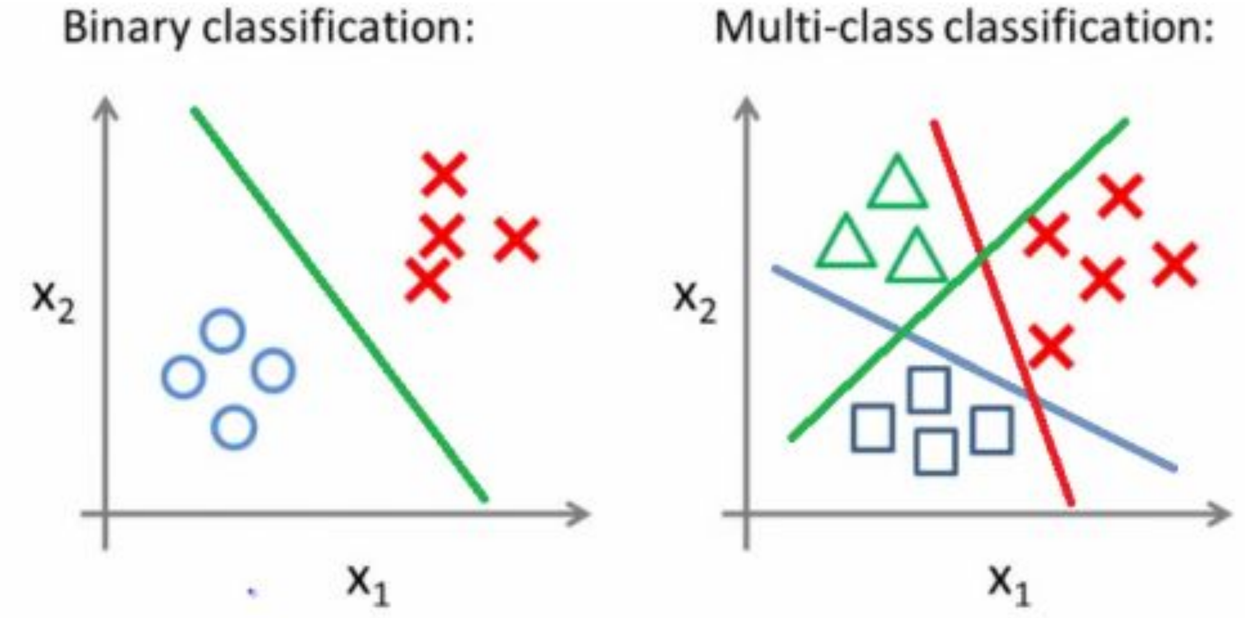
Decide whether an applicant should be approved.

Classify applications into two categories: *approved*, and *not-approved*

Bank Customer Analysis: What are we trying to find out?

- Problem # 2 = a proactive approach to bad/good customers based on their loan status
- Can we predict the status of a loan based on the data from the loan and trans table?
- Approach - develop a simple ML classification model to predict the status of the loan.
- Tool = logistic regression

Types of classification problems



WHAT DO WE WANT TO PREDICT?

- We want to predict the "STATUS" of a bank customer from a single transaction
- If we choose binary classification:

Can we predict A or B status ?

- Multi class classification:

Which status will it be, A,B,C or D?

Stage 1 - IMPORTING SQL DATA INTO PYTHON

- We can do a left join to bring in all trans and loans they are linked to
- Ignoring any Id columns, renaming columns as appropriate

```
query = '''select t.type, t.operation, t.amount as t_amount,
t.balance, t.k_symbol, l.amount as l_amount, l.duration,
l.payments, l.status from trans t
left join loan l
on t.account_id = l.account_id
where l.status in ('A', 'B');'''
data = pd.read_sql_query(query, engine)
data.head()
```

Next stages

1. Create a data frame in pandas/python
2. Use EDA methods to explore the data
3. Identify any needed clean steps
4. What pre-processing needed for this model?
5. T-T split and Run the model

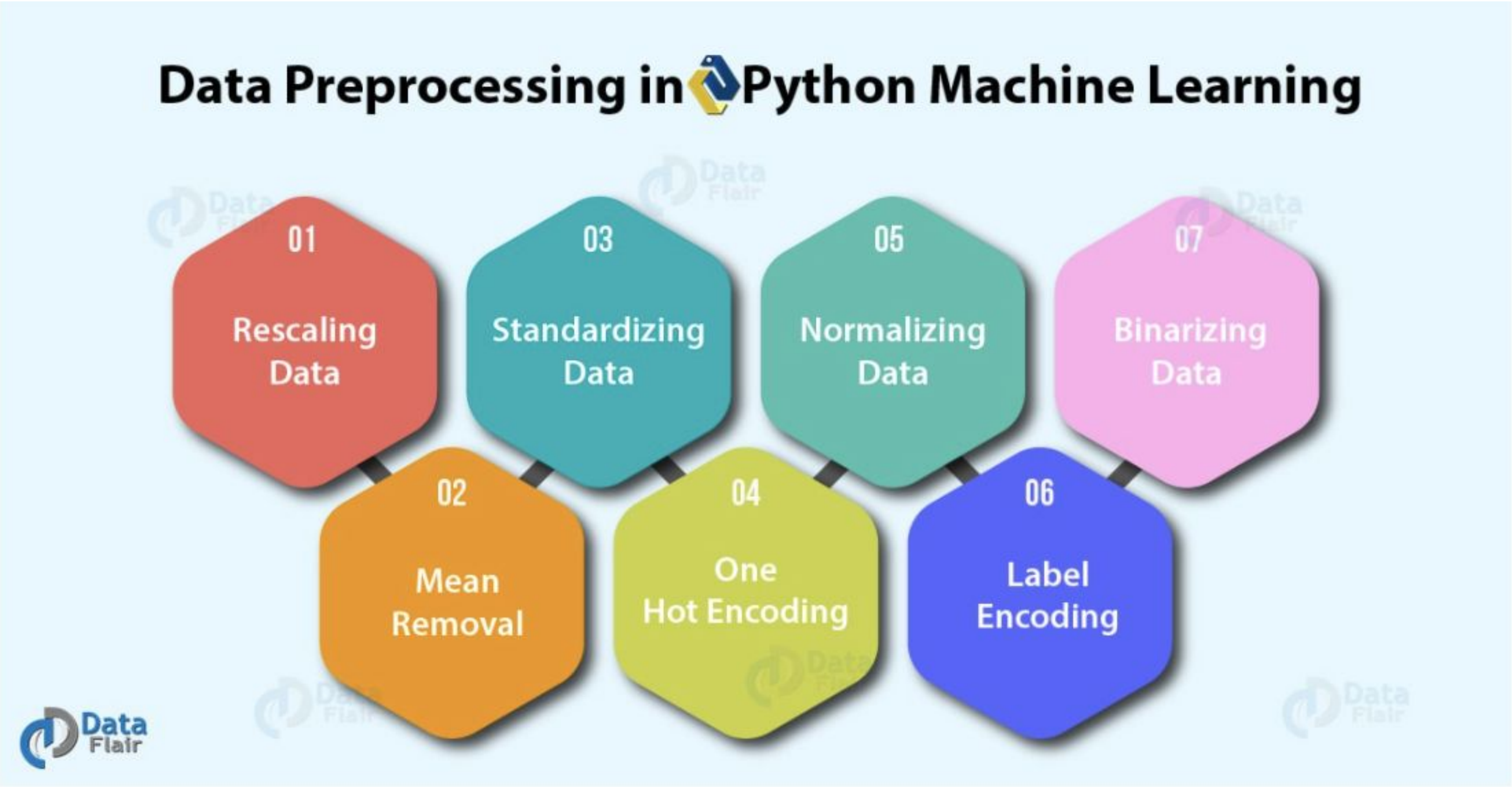
4. Pre-processing

Q: are the data types correct? Do we need to encode categorical data? Are there any numeric data types which could be treated as categorical data

T: Check for multicollinearity (not necessarily a concern for logistic regression!)

Q: Are the features skewed? If yes, normalise/scale

NOTE - for logistic regression, skewness in independent variable is not a problem as is with linear regression. But transformation may help the model fit /run



Data Preprocessing in Python Machine Learning
01
Rescaling Data
02
Mean Removal
03
Standardizing Data
04
One Hot Encoding
05
Normalizing Data
06
Label Encoding
07
Binarizing Data
Data Flair

5. Train Test split and Run the model

Test size 30%

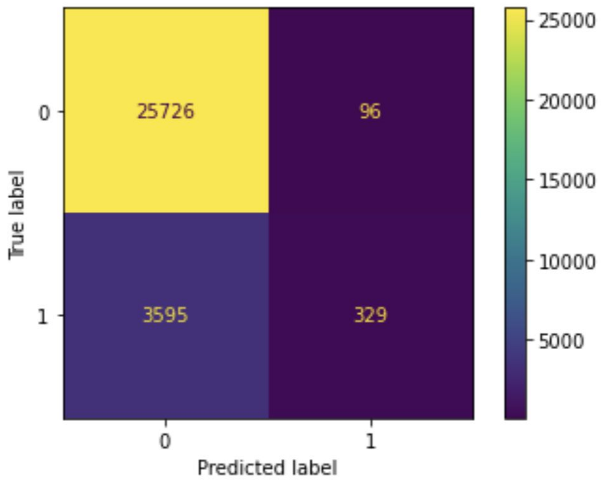
Import logistic regression

Classification score

Next stages

6. Evaluation of the model - accuracy, confusion matrix

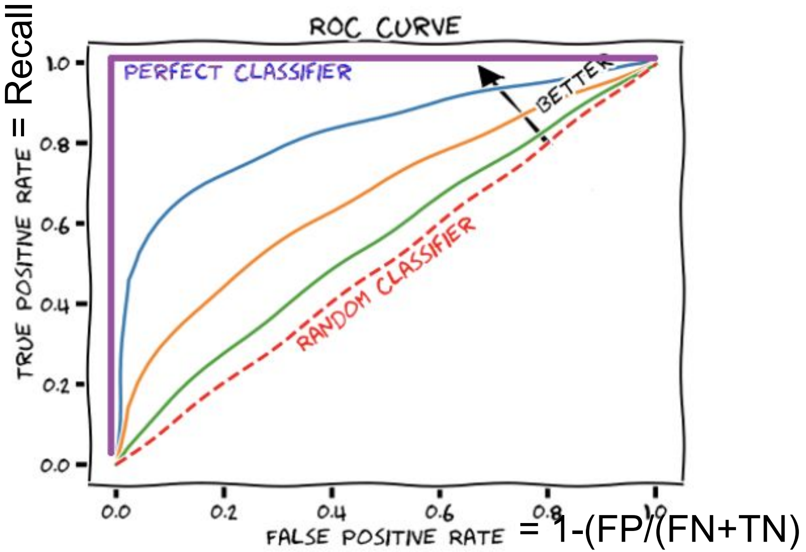
7. Consider if the model is overfit



- Main Challenges of Machine Learning
  - Insufficient Quantity of Training Data
  - Nonrepresentative Training Data
  - Poor-Quality Data
  - Irrelevant Features
  - Overfitting the Training Data
  - Underfitting the Training Data

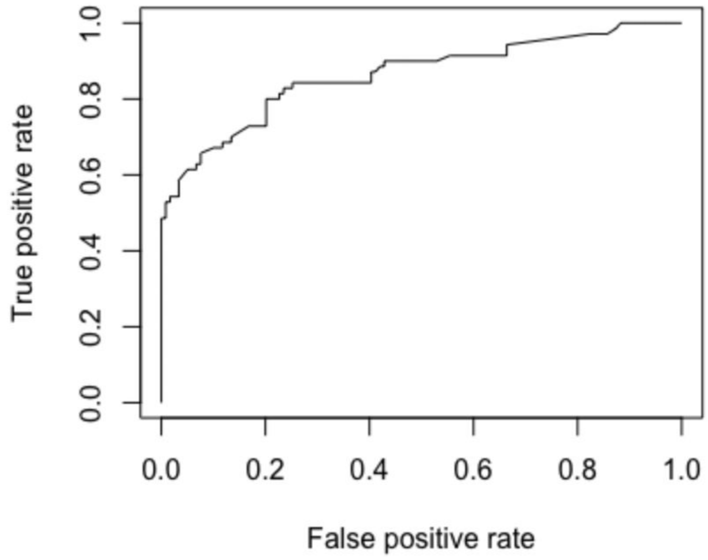
Stage 6 - how can we evaluate a model?

- The ROC curve is a method to compare different settings for your model (threshold).
- It answer the question of: “What is the best threshold to use?”
- Each dot of the curve corresponds to a different value of the threshold



Stage 6 - how can we know a model is useful?

- Accuracy = x- is this good ? how many would we have got correct with guesswork?
- ROC (receiver operating characteristic)- plots true positives against false positives
- Plot the curve - bigger the AUC (area under the curve) the better the model
- We want a curve as close to top left as possible



Stage 6 - how can we evaluate a model?

GETTING THE ROC FROM SKLEARN

```
from sklearn import metrics
import matplotlib.pyplot as plt

y_pred_proba = classification.predict_proba(X_test)[::,1]
fpr, tpr, _ = metrics.roc_curve(y_test,  y_pred_proba)
auc = metrics.roc_auc_score(y_test, y_pred_proba)
plt.plot(fpr,tpr)
```

Confusion matrix - imagine we want to predict if a handwritten digit is a 5 (or not). If our model is good, it will often be right.

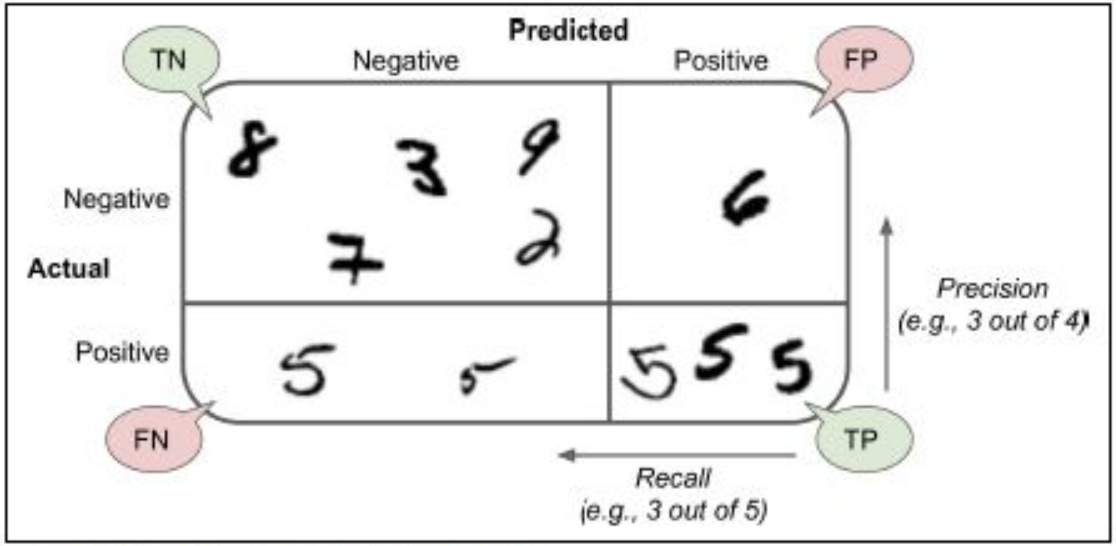


Figure 3-2. An illustrated confusion matrix

Stage 6 - how can we evaluate a model?

GETTING THE CONFUSION MATRIX FROM SKLEARN

```
>>> from sklearn.metrics import confusion_matrix
>>> y_true = [2, 0, 2, 2, 0, 1]
>>> y_pred = [0, 0, 2, 2, 0, 2]
>>> confusion_matrix(y_true, y_pred)
array([[2, 0, 0],
       [0, 0, 1],
       [1, 0, 2]])
```

Stage 6 - how can we evaluate a model?

GETTING THE CONFUSION MATRIX FROM SKLEARN

```
>>> from sklearn.metrics import confusion_matrix
>>> y_true = ["cat", "ant", "cat", "cat", "ant", "bird"]
>>> y_pred = ["ant", "ant", "cat", "cat", "ant", "cat"]
>>> confusion_matrix(y_true, y_pred, labels=["ant", "bird",
"cat"])
array([[2, 0, 0],
       [0, 0, 1],
       [1, 0, 2]])
```

CONFUSED? USE THE CONFUSION MATRIX!

		Real: Positive	Real: Negative
Predicted	Positive	TP	FP
	Negative	FN	TN

CONFUSED? USE THE CONFUSION MATRIX!

		Real	
		Positive	Negative
Predicted	Positive	TP	FP Type I error
	Negative	FN Type II error	TN

If you are deemed to fail what kind of error do you prefer to make?

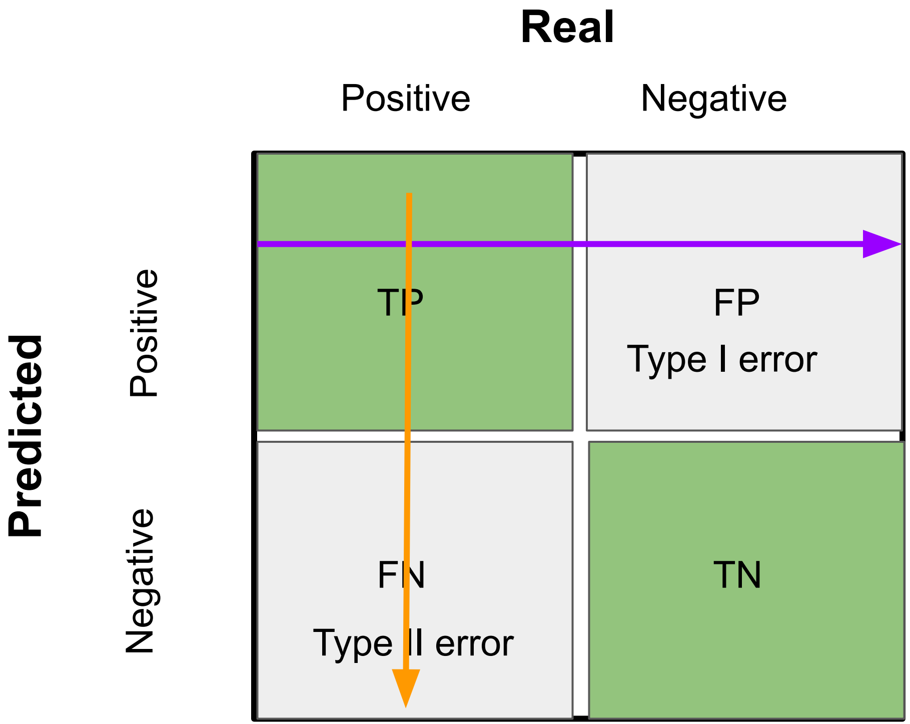
CONFUSED? USE THE CONFUSION MATRIX!

		Real	
		Positive	Negative
Predicted	Positive	TP	FP Type I error
	Negative	FN Type II error	TN

CONFUSED? USE THE CONFUSION MATRIX!

		Real: Positive	Real: Negative
Predicted	Positive	TP	FP Type I error
	Negative	FN Type II error	TN

CONFUSED? USE THE CONFUSION MATRIX!



Accuracy = (TP+TN)/(TP+TN+FP+FN)

Precision = (TP)/(TP+FP)

Recall = (TP)/(TP+FN)

F1 = 2 *(Precision*Recall)/(Precision + Recall)

Can we improve the confusion matrix?

Conceptual example - Confusion Matrix with Titanic data

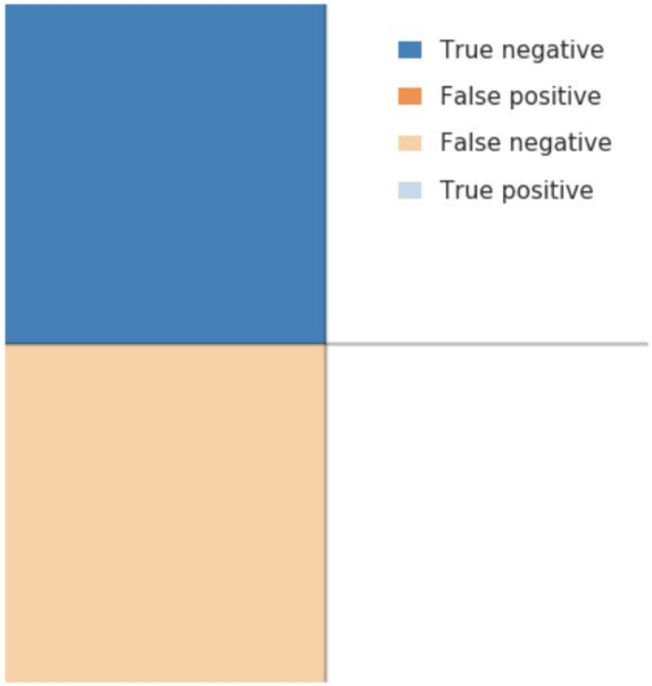
Lets assert everyone died

Classic example of a NULL hypothesis

Accuracy = 0.61 (errors 0 39%)

Reducible error - add some variables / generalisations into our model to iterate towards better accuracy.

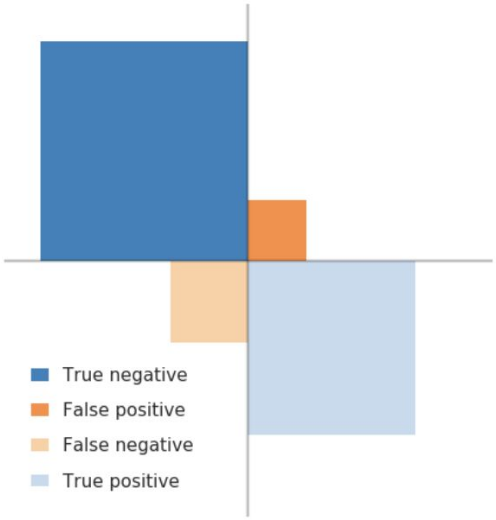
Irreducible error - survival based on chance - people jumped into the water to save relatives, jumped into lifeboats while they were being lowered, or jumped into the water.

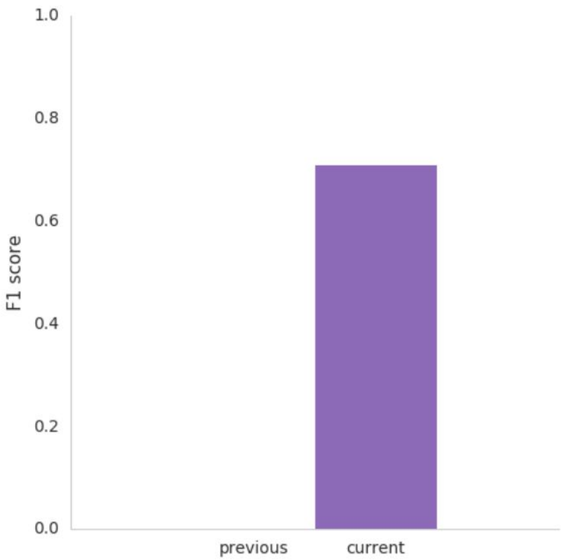


Confusion Matrix with Titanic data- generalisation A

“women and children first” … lets assert all women survived

```
confusion_scores.append(get_confusion_scores(LogisticRegression(), X, y))
plot_confusion_mat(null_mat, get_confused_mat(X, y), scores=confusion_scores)
```

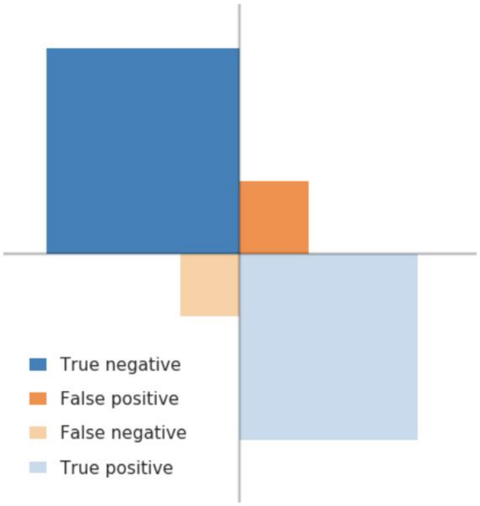


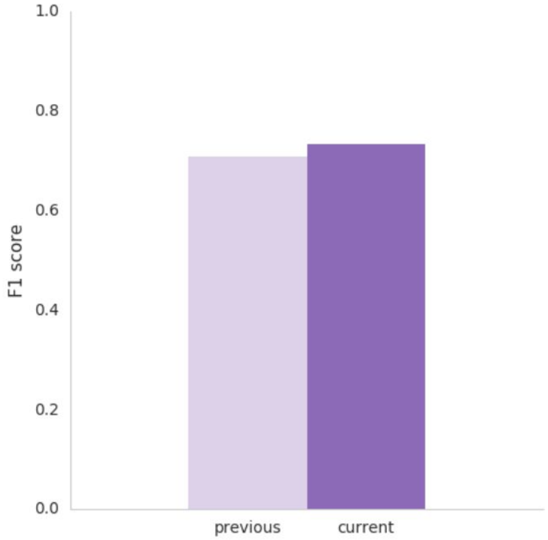


Confusion Matrix with Titanic data- generalisation B

“women and children first” … split the children out

```
confusion_scores.append(get_confusion_scores(LogisticRegression(), X, y))
plot_confusion_mat(null_mat, get_confused_mat(X, y), scores=confusion_scores)
```

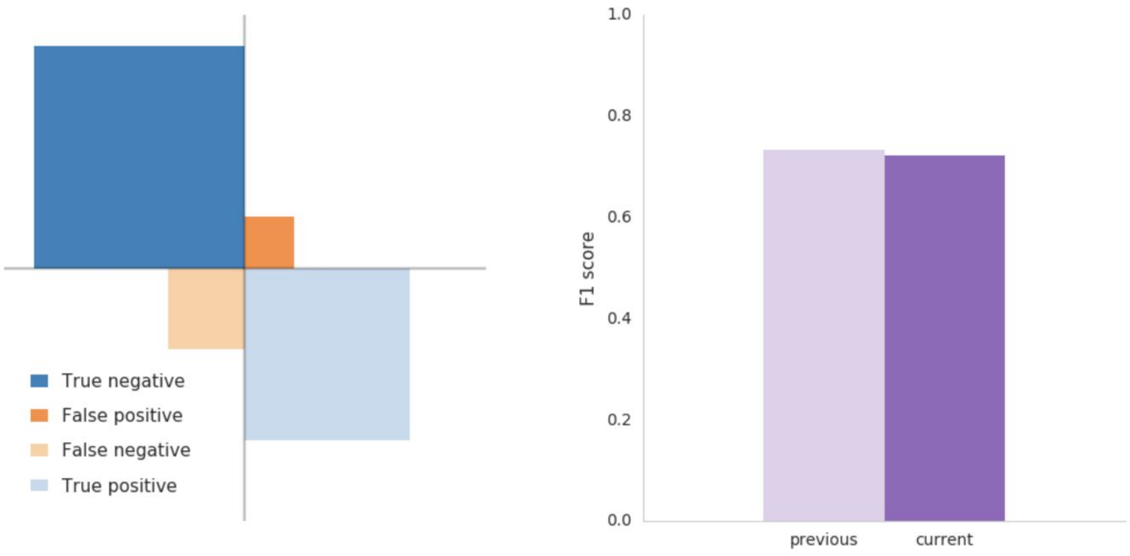




Confusion Matrix with Titanic data- generalisation C

Class matters ... split the passengers by class

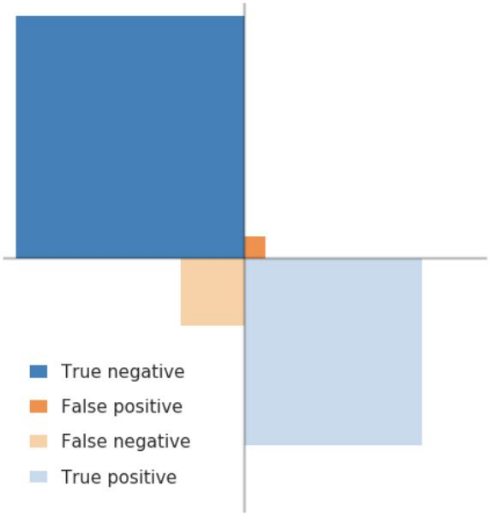
```
confusion_scores.append(get_confusion_scores(LogisticRegression(), X, y))
plot_confusion_mat(null_mat, get_confused_mat(X, y), scores=confusion_scores)
```

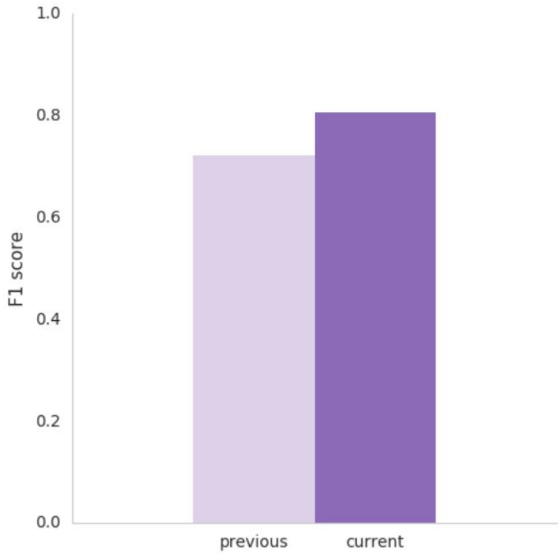


Confusion matrix... perishing mothers/wives

No of siblings/spouses, parents/children is an influence on survival

```
confusion_scores.append(get_confusion_scores(LogisticRegression(), X, y))
plot_confusion_mat(null_mat, get_confused_mat(X, y), scores=confusion_scores)
```

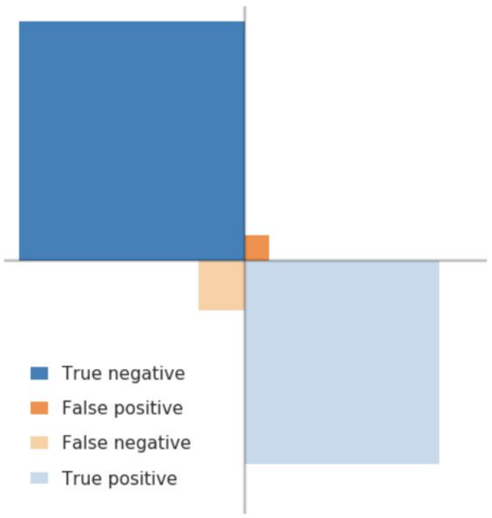


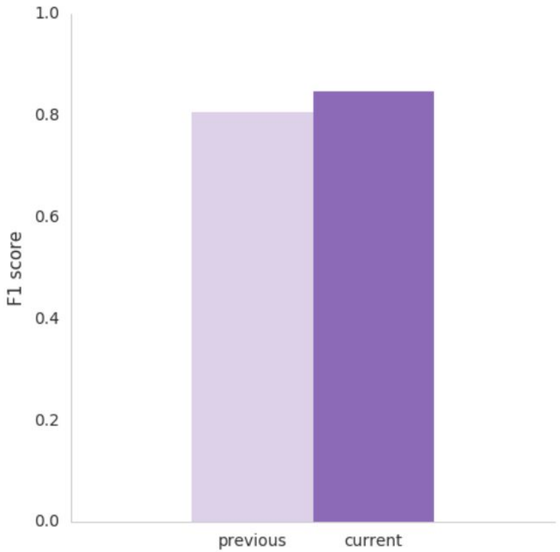


Confusion matrix... surviving_father_husband

No of siblings/spouses, parents/children is an influence on survival

```
confusion_scores.append(get_confusion_scores(LogisticRegression(), X, y))
plot_confusion_mat(null_mat, get_confused_mat(X, y), scores=confusion_scores)
```





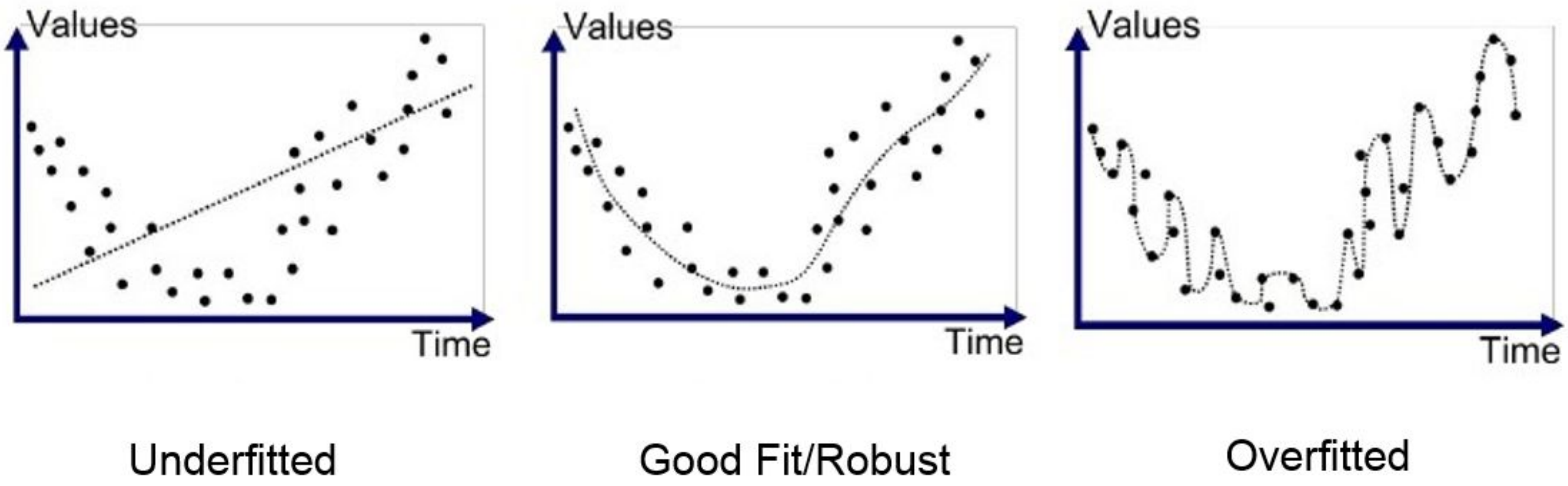
Next stages

6. Evaluation of the model - accuracy, confusion matrix

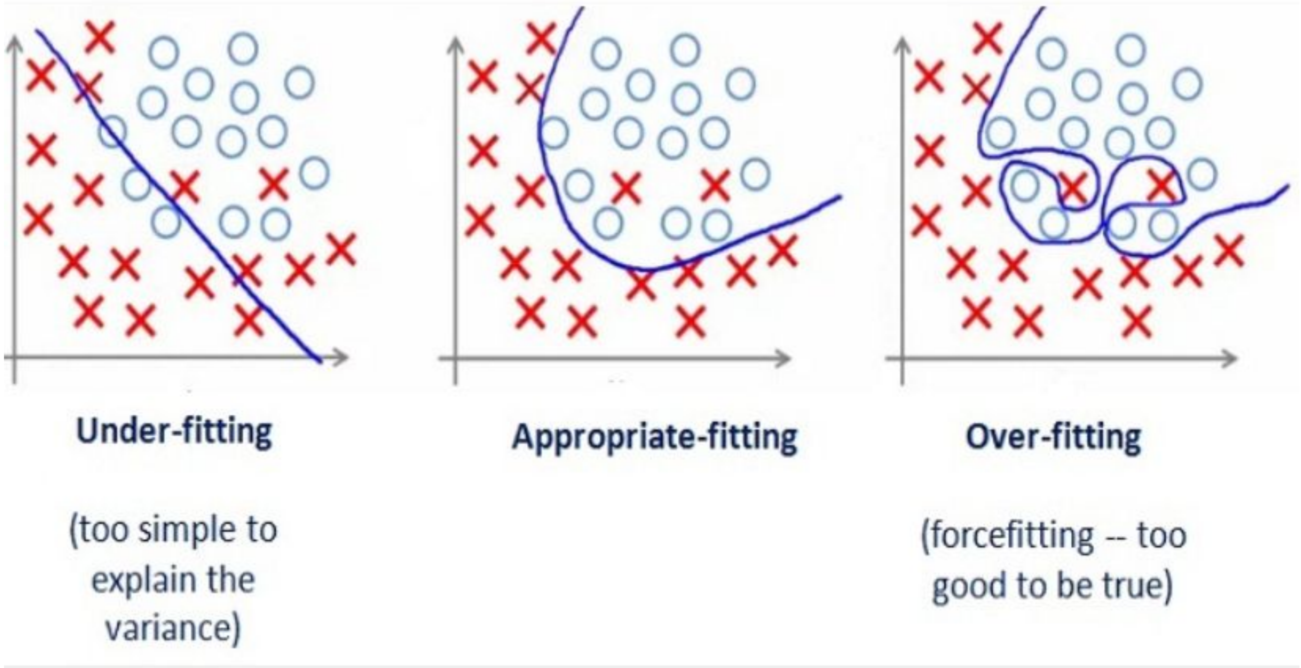
7. **Consider if the model is overfit**

8. Optional - pickle the code in sections for reuse

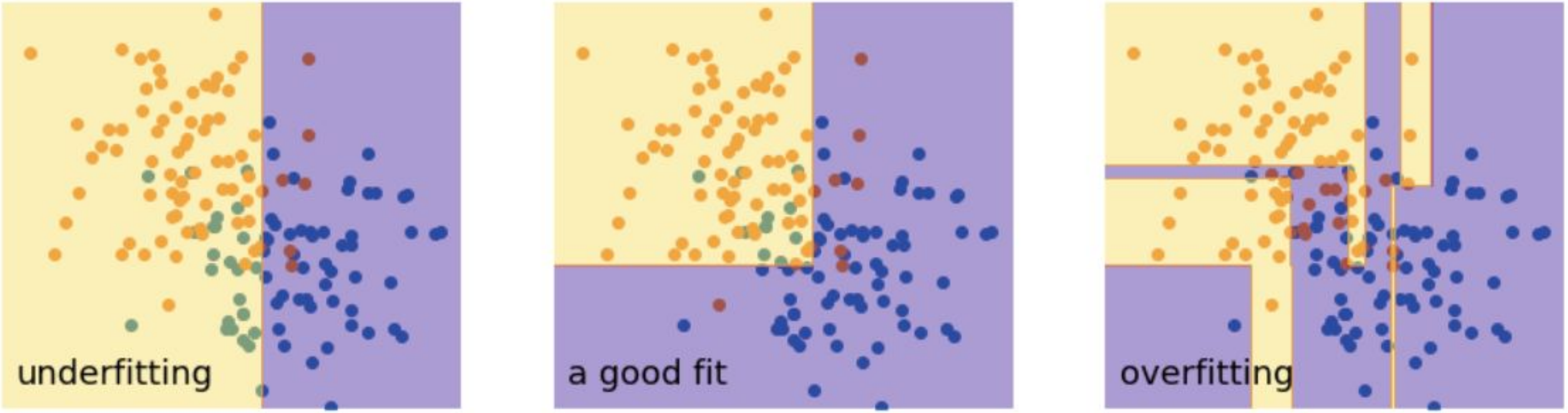
Stage 7 - overfitting (it happens...)



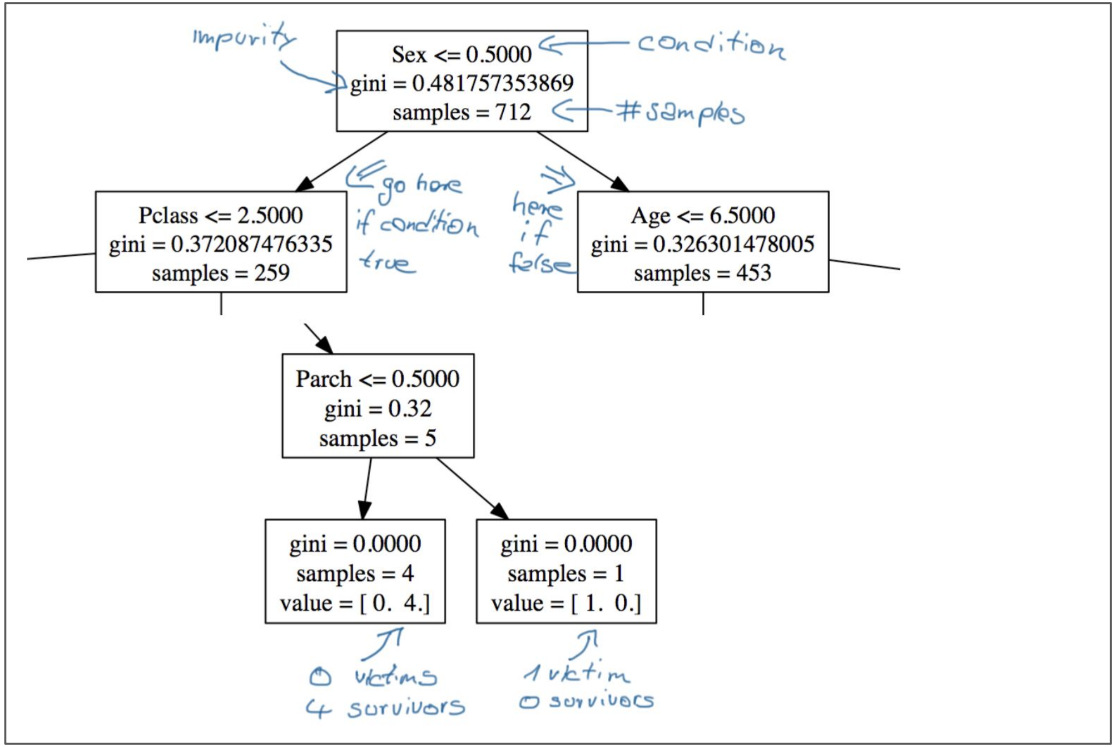
Stage 7 - overfitting (it happens...)



Stage 7 - overfitting (it happens...)

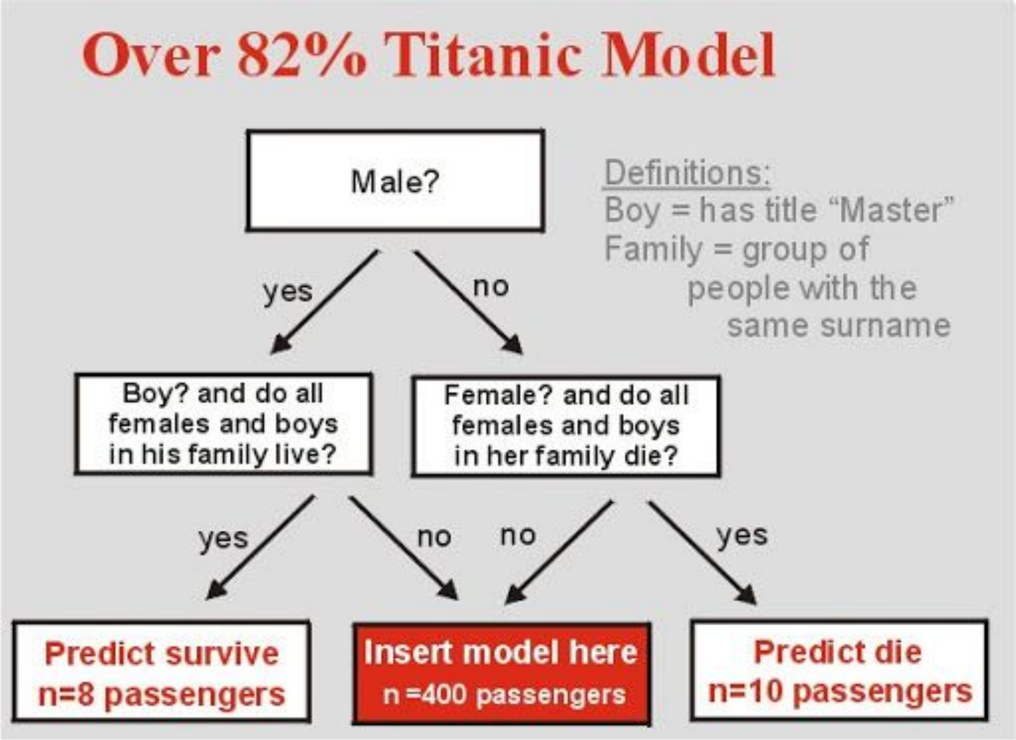


Output Titanic data - overfitting?

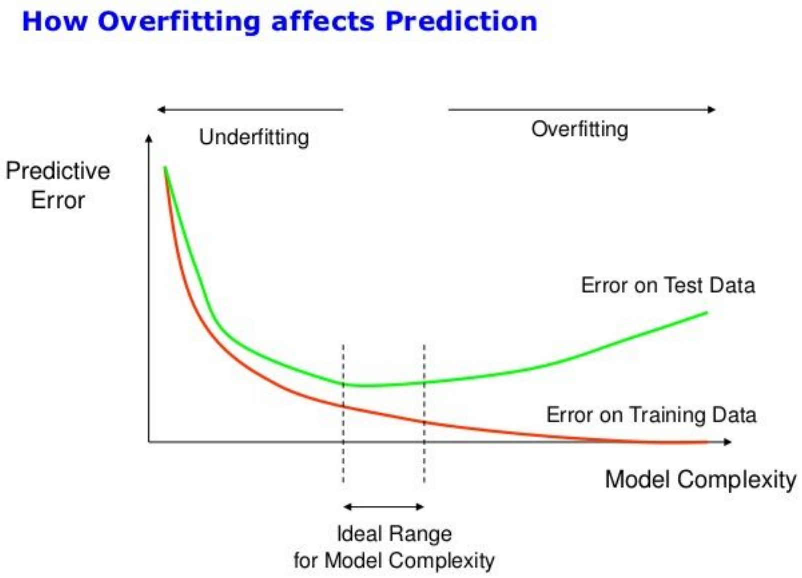


Output Titanic data - overfitting?

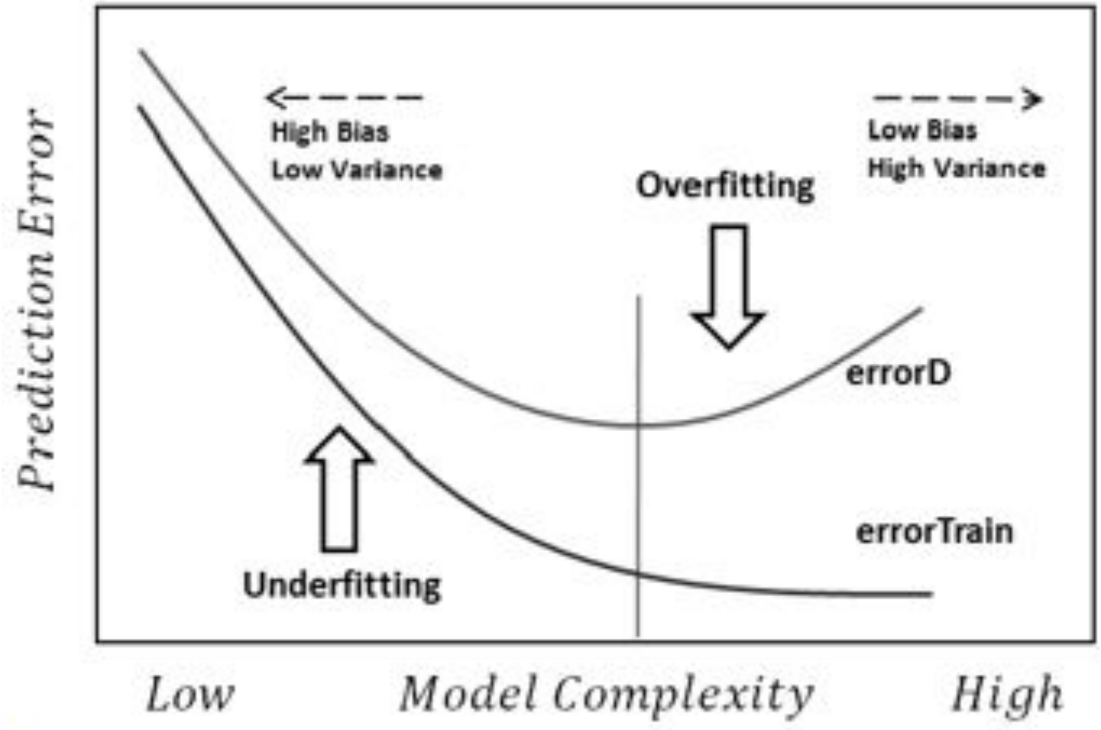
Imagine if you used surname to train the model - would this result in overfit ?



Impact of overfitting to train data



Bias to overfit

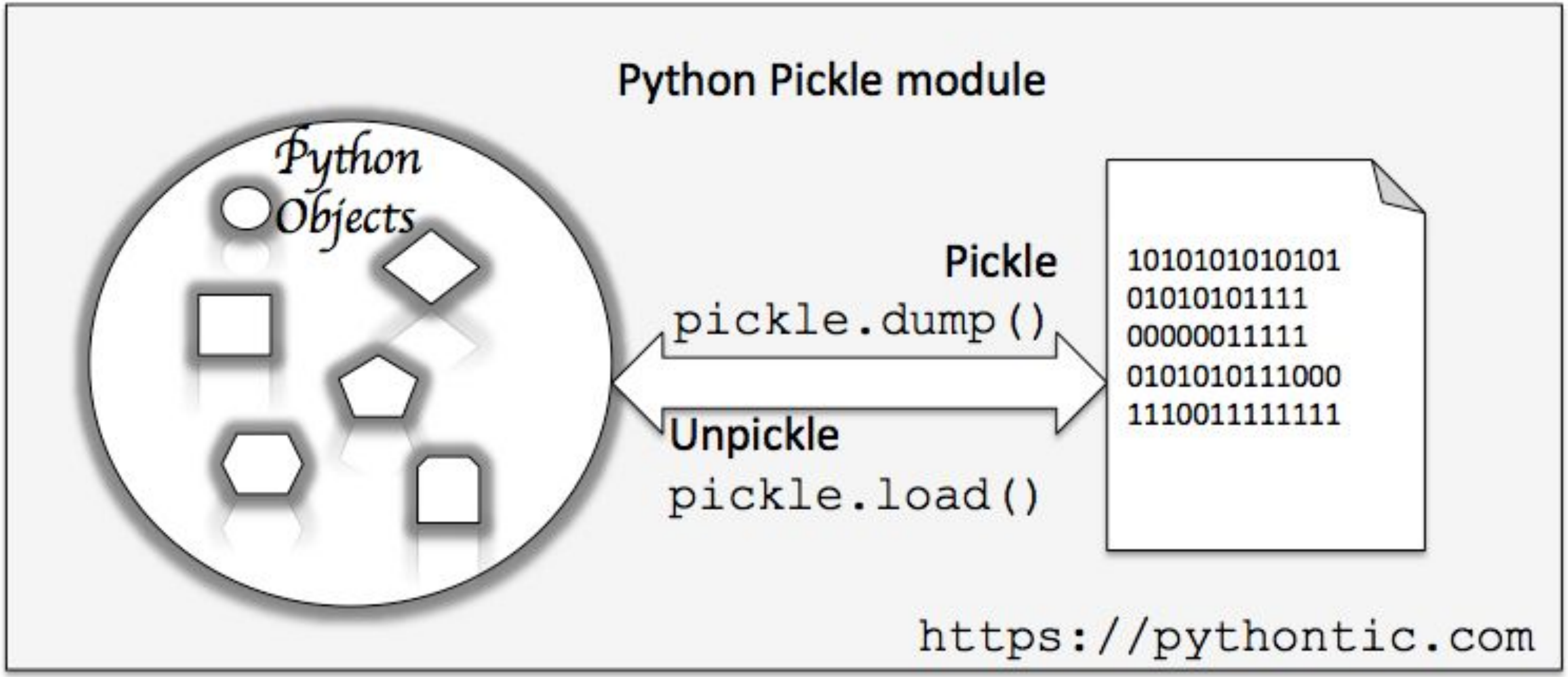


Next stages

6. Evaluation of the model - accuracy, confusion matrix

7. Consider if the model is overfit

8. **Pickle the code in sections for reuse**


Python Pickle module
Python Objects
Pickle
pickle.dump()
Unpickle
pickle.load()
1010101010101
01010101111
00000011111
0101010111000
1110011111111
https://pythontic.com

Some self paced resources on ML

What is Machine Learning?, Google Cloud Platform (5:22 min) - https://www.youtube.com/watch?v=HcqpanDadyQ

- Intro to Machine Learning (ML Zero to Hero - Part 1), Tensorflow (7:17 min) - https://www.youtube.com/watch?v=KNAWp2S3w94

- A Gentle Introduction to Machine Learning, StatQuest (12:44 min) - https://www.youtube.com/watch?v=Gv9_4yMHFhI&t=0s

Reading list :