👋 🛳️ Ahoy, welcome to Kaggle! You're in the right place.

This is the legendary Titanic ML competition – the best, first challenge for you to dive into ML competitions and familiarize yourself with how the Kaggle platform works.

The competition is simple: use machine learning to create a model that predicts which passengers survived the Titanic shipwreck.

The Challenge

The sinking of the Titanic is one of the most infamous shipwrecks in history.

On April 15, 1912, during her maiden voyage, the widely considered "unsinkable" RMS Titanic sank after colliding with an iceberg. Unfortunately, there weren't enough lifeboats for everyone onboard, resulting in the death of 1502 out of 2224 passengers and crew.

While there was some element of luck involved in surviving, it seems some groups of people were more likely to survive than others.

In this challenge, we ask you to build a predictive model that answers the question: "what sorts of people were more likely to survive?" using passenger data (ie name, age, gender, socio-economic class, etc).

https://www.kaggle.com/c/titanic

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

https://www.kaggle.com/c/titanic

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

Variable Notes

pclass: A proxy for socio-economic status (SES)

1st = Upper

2nd = Middle

3rd = Lower

age: Age is fractional if less than 1. If the age is estimated, is it in the form of xx.5

sibsp: The dataset defines family relations in this way...

Sibling = brother, sister, stepbrother, stepsister

Spouse = husband, wife (mistresses and fiancés were ignored)

parch: The dataset defines family relations in this way...

Parent = mother, father

Child = daughter, son, stepdaughter, stepson

Some children travelled only with a nanny, therefore parch=0 for them.

https://www.kaggle.com/c/titanic

Load and display the data

```
1 df = pd.read_csv('data.csv')
2 df
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	290	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

Check data structure (column names, value types)

```
1  df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

Take a quick look at some data statistics

```
1 df.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

Take a quick look at non-numerical columns too

```
1 df.describe(include = 'all')
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
count	891.000000	891.000000	891.000000	891	891	714.000000	891.000000	891.000000	891	891.000000	204	889
unique	NaN	NaN	NaN	891	2	NaN	NaN	NaN	681	NaN	147	3
top	NaN	NaN	NaN	290	male	NaN	NaN	NaN	347082	NaN	B96 B98	S
freq	NaN	NaN	NaN	1	577	NaN	NaN	NaN	7	NaN	4	644
mean	446.000000	0.383838	2.308642	NaN	NaN	29.699118	0.523008	0.381594	NaN	32.204208	NaN	NaN
std	257.353842	0.486592	0.836071	NaN	NaN	14.526497	1.102743	0.806057	NaN	49.693429	NaN	NaN
min	1.000000	0.000000	1.000000	NaN	NaN	0.420000	0.000000	0.000000	NaN	0.000000	NaN	NaN
25%	223.500000	0.000000	2.000000	NaN	NaN	20.125000	0.000000	0.000000	NaN	7.910400	NaN	NaN
50%	446.000000	0.000000	3.000000	NaN	NaN	28.000000	0.000000	0.000000	NaN	14.454200	NaN	NaN
75%	668.500000	1.000000	3.000000	NaN	NaN	38.000000	1.000000	0.000000	NaN	31.000000	NaN	NaN
max	891.000000	1.000000	3.000000	NaN	NaN	80.000000	8.000000	6.000000	NaN	512.329200	NaN	NaN

Divide data to features and labels

```
X = df.drop(['Survived'], axis=1)
X
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	3	290	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...
886	887	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

```
Y = df[['Survived']]
Y
```

	Survived
0	0
1	1
2	1
3	1
4	0
...	...
886	0
887	1
888	0
889	1
890	0

Remove unused columns

```
1 X = X.drop(['PassengerId', 'Ticket'], axis=1)
2 X
```

	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	3	290	male	22.0	1	0	7.2500	NaN	S
1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
2	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
3	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...
886	2	Montvila, Rev. Juozas	male	27.0	0	0	13.0000	NaN	S
887	1	Graham, Miss. Margaret Edith	female	19.0	0	0	30.0000	B42	S
888	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	23.4500	NaN	S
889	1	Behr, Mr. Karl Howell	male	26.0	0	0	30.0000	C148	C
890	3	Dooley, Mr. Patrick	male	32.0	0	0	7.7500	NaN	Q

Treat missing values, Nans etc

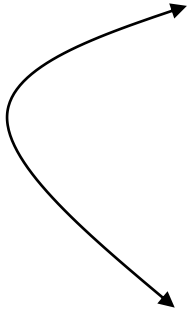
```
1  df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

Replace missing values with column's median

```
1 X['Age'].fillna(X['Age'].median(), inplace = True)
2 X
```

	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	3	290	male	22.0	1	0	7.2500	NaN	S
1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
2	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
3	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...
886	2	Montvila, Rev. Juozas	male	27.0	0	0	13.0000	NaN	S
887	1	Graham, Miss. Margaret Edith	female	19.0	0	0	30.0000	B42	S
888	3	Johnston, Miss. Catherine Helen "Carrie"	female	28.0	1	2	23.4500	NaN	S
889	1	Behr, Mr. Karl Howell	male	26.0	0	0	30.0000	C148	C
890	3	Dooley, Mr. Patrick	male	32.0	0	0	7.7500	NaN	Q



888	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	23.4500	NaN	S

Replace missing values with column's most frequent value

```
1 X['Embarked'] = X['Embarked'].fillna(X.Embarked.dropna().mode()[0])
2 X
```

	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	3	290	male	22.0	1	0	7.2500	NaN	S
1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
2	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
3	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...
886	2	Montvila, Rev. Juozas	male	27.0	0	0	13.0000	NaN	S
887	1	Graham, Miss. Margaret Edith	female	19.0	0	0	30.0000	B42	S
888	3	Johnston, Miss. Catherine Helen "Carrie"	female	28.0	1	2	23.4500	NaN	S
889	1	Behr, Mr. Karl Howell	male	26.0	0	0	30.0000	C148	C
890	3	Dooley, Mr. Patrick	male	32.0	0	0	7.7500	NaN	Q

Deal with rare/equivalent values

	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked	Title	CabinNumber
0	3	male	22.0	1	0	7.2500	9	S	Unclear	0
1	1	female	38.0	1	0	71.2833	3	C	Mrs	85
2	3	female	26.0	0	0	7.9250	9	S	Miss	0
3	1	female	35.0	1	0	53.1000	3	S	Mrs	123
4	3	male	35.0	0	0	8.0500	9	S	Mr	0
...	...	...	...	...	...	...	...	...	...	...
886	2	male	27.0	0	0	13.0000	9	S	Rev	0
887	1	female	19.0	0	0	30.0000	2	S	Miss	42
888	3	female	28.0	1	2	23.4500	9	S	Miss	0
889	1	male	26.0	0	0	30.0000	3	C	Mr	148
890	3	male	32.0	0	0	7.7500	9	Q	Mr	0

```
1 X[['Title']].value_counts()
```

```
Title
Mr          516
Miss        182
Mrs         125
Master       40
Dr            7
Rev           6
Major         2
Col           2
Mlle          2
Sir           1
Ms            1
Capt          1
Mme           1
Lady          1
Jonkheer      1
Don           1
Countess      1
Unclear       1
```

Deal with rare/equivalent values

```
X['Title'] = X['Title'].replace(['Lady', 'Countess','Capt', 'Col', 'Don', 'Dr', 'Major', 'Rev', 'Sir',
                                 'Jonkheer', 'Dona', 'Unclear'], 'Rare')
X['Title'] = X['Title'].replace('Mlle', 'Miss')
X['Title'] = X['Title'].replace('Ms', 'Miss')
X['Title'] = X['Title'].replace('Mme', 'Mrs')
X
```

	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked	Title	CabinNumber
0	3	male	22.0	1	0	7.2500	9	S	Rare	0
1	1	female	38.0	1	0	71.2833	3	C	Mrs	85
2	3	female	26.0	0	0	7.9250	9	S	Miss	0
3	1	female	35.0	1	0	53.1000	3	S	Mrs	123
4	3	male	35.0	0	0	8.0500	9	S	Mr	0
...	...	...	...	...	...	...	...	...	...	...
886	2	male	27.0	0	0	13.0000	9	S	Rare	0
887	1	female	19.0	0	0	30.0000	2	S	Miss	42
888	3	female	28.0	1	2	23.4500	9	S	Miss	0
889	1	male	26.0	0	0	30.0000	3	C	Mr	148
890	3	male	32.0	0	0	7.7500	9	Q	Mr	0

```
X[['Title']].value_counts()
```

```
Title
Mr        516
Miss      185
Mrs       126
Master     40
Rare       24
```

Enumerate categorical features (other encodings might be more effective)

```
1 X['Sex'] = X['Sex'].map( {'female': 1, 'male': 0} ).astype(int)
2 X
```

	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked	Title	CabinNumber
0	3	0	22.0	1	0	7.2500	9	S	5	0
1	1	1	38.0	1	0	71.2833	3	C	3	85
2	3	1	26.0	0	0	7.9250	9	S	2	0
3	1	1	35.0	1	0	53.1000	3	S	3	123
4	3	0	35.0	0	0	8.0500	9	S	1	0
...	...	...	...	...	...	...	...	...	...	...
886	2	0	27.0	0	0	13.0000	9	S	5	0
887	1	1	19.0	0	0	30.0000	2	S	2	42
888	3	1	28.0	1	2	23.4500	9	S	2	0
889	1	0	26.0	0	0	30.0000	3	C	1	148
890	3	0	32.0	0	0	7.7500	9	Q	1	0

Welcome to the year 2912, where your data science skills are needed to solve a cosmic mystery. We've received a transmission from four lightyears away and things aren't looking good.

The *Spaceship Titanic* was an interstellar passenger liner launched a month ago. With almost 13,000 passengers on board, the vessel set out on its maiden voyage transporting emigrants from our solar system to three newly habitable exoplanets orbiting nearby stars.

While rounding Alpha Centauri en route to its first destination—the torrid 55 Cancri E—the unwary *Spaceship Titanic* collided with a spacetime anomaly hidden within a dust cloud. Sadly, it met a similar fate as its namesake from 1000 years before. Though the ship stayed intact, almost half of the passengers were transported to an alternate dimension!

To help rescue crews and retrieve the lost passengers, you are challenged to predict which passengers were transported by the anomaly using records recovered from the spaceship's damaged computer system.

Help save them and change history!

https://www.kaggle.com/competitions/spaceship-titanic/

File and Data Field Descriptions

- **train.csv** - Personal records for about two-thirds (~8700) of the passengers, to be used as training data.
  - `PassengerId` - A unique Id for each passenger. Each Id takes the form `gggg_pp` where `gggg` indicates a group the passenger is travelling with and `pp` is their number within the group. People in a group are often family members, but not always.
  - `HomePlanet` - The planet the passenger departed from, typically their planet of permanent residence.
  - `CryoSleep` - Indicates whether the passenger elected to be put into suspended animation for the duration of the voyage. Passengers in cryosleep are confined to their cabins.
  - `Cabin` - The cabin number where the passenger is staying. Takes the form `deck/num/side`, where `side` can be either `P` for *Port* or `S` for *Starboard*.
  - `Destination` - The planet the passenger will be debarking to.
  - `Age` - The age of the passenger.
  - `VIP` - Whether the passenger has paid for special VIP service during the voyage.
  - `RoomService`, `FoodCourt`, `ShoppingMall`, `Spa`, `VRDeck` - Amount the passenger has billed at each of the *Spaceship Titanic*'s many luxury amenities.
  - `Name` - The first and last names of the passenger.
  - `Transported` - Whether the passenger was transported to another dimension. This is the target, the column you are trying to predict.

PassengerId	HomePlanet	CryoSleep	Cabin	Destination	Age	VIP	RoomService	FoodCourt	ShoppingMall	Spa	VRDeck	Name	Transported
0001_01	Europa	False	B/0/P	TRAPPIST-1e	39.0	False	0.0	0.0	0.0	0.0	0.0	Maham Ofracculy	False
0002_01	Earth	False	F/0/S	TRAPPIST-1e	24.0	False	109.0	9.0	25.0	549.0	44.0	Juanna Vines	True
0003_01	Europa	False	A/0/S	TRAPPIST-1e	58.0	True	43.0	3576.0	0.0	6715.0	49.0	Altark Susent	False
0003_02	Europa	False	A/0/S	TRAPPIST-1e	33.0	False	0.0	1283.0	371.0	3329.0	193.0	Solam Susent	False
0004_01	Earth	False	F/1/S	TRAPPIST-1e	16.0	False	303.0	70.0	151.0	565.0	2.0	Willy Santantines	True
...	...	...	...	...	...	...	...	...	...	...	...	...	...
9276_01	Europa	False	A/98/P	55 Cancri e	41.0	True	0.0	6819.0	0.0	1643.0	74.0	Gravior Noxnuther	False
9278_01	Earth	True	G/1499/S	PSO J318.5-22	18.0	False	0.0	0.0	0.0	0.0	0.0	Kurta Mondalley	False
9279_01	Earth	False	G/1500/S	TRAPPIST-1e	26.0	False	0.0	0.0	1872.0	1.0	0.0	Fayey Connon	True
9280_01	Europa	False	E/608/S	55 Cancri e	32.0	False	0.0	1049.0	0.0	353.0	3235.0	Celeon Hontichre	False
9280_02	Europa	False	E/608/S	TRAPPIST-1e	44.0	False	126.0	4688.0	0.0	0.0	12.0	Propsh Hontichre	True

8693 rows × 14 columns