



Kaggle Titanic 入門 01

Kaggle を始めてみよう！

庭師

庭子アバター v01.50
Avatar Museum12に展示中

自己紹介:庭師 / ni-wa-shi

- 大手製造業で新規事業開発だった、データサイエンティストやっています。
- 結構な年のおじさんです。(_ _)
- AI/データサイエンティストの方が儲かるなあ、と思いつつ。アバター作成の方が楽しいと感じています。アバター作成は定年後の盆栽趣味的にやっています。
- Kaggleは2019年から。最近、やってないので、活動復活させたいな、誰か一緒にやる人いないかなあ、と思っています。

誰か一緒に遊んで！

<https://www.kaggle.com/marutama>

The image shows a screenshot of a Kaggle user profile for 'marutama' (Niwashi). The profile includes a circular avatar of a character with black hair and green eyes, a bio stating 'Engineer and Manager at Semiconductor' from 'Funabashi, Chiba, Japan', and a 'Notebooks Master' badge with 3,221 of 57,175 points. Below the profile are tabs for 'About', 'Competitions (25)', 'Datasets (7)', 'Code (19)', 'Discussion (58)', 'Followers (73)', and 'Following (14)'. The 'Kaggle Achievements' section displays four categories: 'Competitions Expert' (Rank 6,252 of 204,663), 'Datasets Contributor' (Rank none yet), 'Notebooks Master' (Rank 3,221 of 57,175), and 'Discussions Contributor' (Rank 143 of 57,175). A red text overlay on the right side of the achievements section reads: 'Notebook Masterは持っています。Master取得狙うならば、Notebookがお勧め。' (I have the Notebook Master badge. If you are aiming for the Master badge, I recommend Kaggle Notebooks.)

marutama
Niwashi
Engineer and Manager at Semiconductor
Funabashi, Chiba, Japan
Joined 6 years ago · last seen in the past day

Notebooks Master
3,221 of 57,175

About Competitions (25) Datasets (7) Code (19) Discussion (58) Followers (73) Following (14)

Kaggle Achievements

Competitions Expert
MEDALS: 2
RANK: 6,252 of 204,663 | 1,668 highest ever

Datasets Contributor
MEDALS: none yet
RANK: none yet

Notebooks Master
MEDALS: 4 8 4
RANK: 3,221 of 57,175 | 143 highest ever

Discussions Contributor
MEDALS: 4 1 21
RANK: none yet

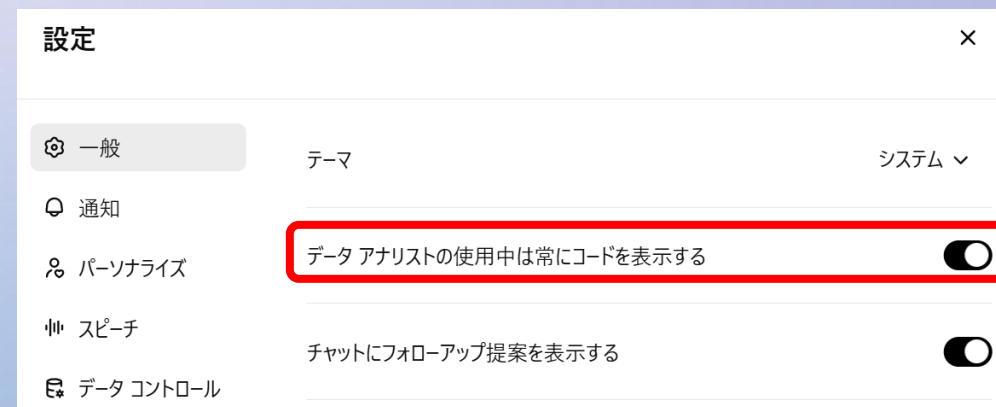
Notebook Masterは持っています。
Master取得狙うならば、Notebookがお勧め。

準備1

- Kaggleのアカウント作しましょう！ 下記などを参考に！
 - [2025]Kaggleサイトのアカウント登録方法の手順まとめ！登録してからやることも紹介
 - <https://ai-kenkyujo.com/software/kaggle/kaggleaccountregistration/>
- 翻訳ツールを入れておこう！
 - Kaggleは基本英語です。**ChromeのGoogle翻訳 拡張機能**など入れておきましょう。コンペの説明、ディスカッションなどほとんどは翻訳できます。
 - ノートブック(Code)翻訳用に bilzard/**kaggle-notebook-translation-helper** を入れましょう
 - Kaggle ノートブック(Code)だけはGoogle翻訳できません。ですが、このツールでiFrameとして変換表示してしまえば翻訳可能！
 - Kaggleの公開ノートブックを日本語に翻訳するChrome拡張機能
 - <https://tech.aru-zakki.com/kaggle-notebook-translation/>

準備2: ChatGPT設定

- ChatGPT Plus(月額 \$20:約3,000円)に入っておこう！
 - ここではChatGPT使いまわりの入門方法を紹介する予定です。実践したい人は入っておいてください。見るだけでもあり。Gemini/Cloudeでもでも同様の事ができると思います。
 - 【方法】 ChatGPTのチャット画面左側に表示されているサイドバーの下部か、画面右上のユーザーアイコンのメニューから「プランをアップグレードする」を選択
- プログラムコードを表示する設定にしておきましょう
 - 右上、自分のプロフィールアイコンから「設定」を選択
 - 「一般」→「データアナリストの使用中にコードを表示する」をonに。



Kaggle Titanic コンペ

<https://www.kaggle.com/competitions/titanic>

Titanic - Machine Learning from Disaster

Start here! Predict survival on the Titanic and get familiar with ML basics

[Overview](#) [Data](#) [Code](#) [Models](#) [Discussion](#) [Leaderboard](#) [Rules](#) [Team](#) [Submissions](#)

Overview

∞ This competition runs indefinitely with a rolling leaderboard. [Learn more](#)

Description

👋📖 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.

If you want to talk with other users about this competition, come join our Discord! We've got channels for competitions, job postings and career discussions, resources, and socializing with your fellow data scientists. Follow the link here: <https://discord.gg/kaggle>

タイタニック号 - 災害からの機械学習

ここから始めましょう！タイタニック号の生存者を予測し、機械学習の基礎を学びましょう

[概要](#) [データ](#) [コード](#) [モデル](#) [議論](#) [リーダーボード](#) [ルール](#) [チーム](#) [提出物](#)

概要

∞ このコンテストは無期限で開催され、リーダーボードは随時更新されます。詳細は[こちら](#)

説明

👋📖 こんにちは、Kaggle へようこそ! ここは最適な場所です。

これは伝説的な Titanic ML コンテストです。ML コンテストに飛び込み、Kaggle プラットフォームの仕組みを理解するための、最初の最高のチャレンジです。

このコンペティションについて他のユーザーと話したい方は、ぜひDiscordにご参加ください! コンペティション、求人情報、キャリアに関するディスカッション、リソース、そしてデータサイエンティスト仲間との交流のためのチャンネルをご用意しています。こちらのリンクからアクセスしてください: <https://discord.gg/kaggle>

必要なら、翻訳してどんどん読んでいこう
でも、有名なTitanicコンペはChatGPTが教えてくれる。



Titanicはどんなコンペ？

タイタニック号は1912年4月サザンプトンからニューヨークへの航行中氷山に衝突して沈没した。1316人の乗客と885人の乗組員が居たが、救助されたのは712人で生存率は32.2%である。

Titanicコンペとは？

概要

1912年に沈没したタイタニック号の乗客データを使い、生存者を予測する 二値分類問題（生きたか死んだか）。

- 入力：乗客の特徴（年齢、性別、船室クラス、同乗家族数など）
- 出力：生存 or 死亡（`Survived` = 1 or 0）

提供されるデータ

ファイル名	説明
<code>train.csv</code>	学習用（891人の乗客。 <code>Survived</code> 列あり）
<code>test.csv</code>	予測対象（418人の乗客。 <code>Survived</code> 列なし）
<code>gender_submission.csv</code>	提出用CSVの例。性別だけを用いた予測

タスク内容

目的：

`train.csv` を使ってモデルを訓練し、`test.csv` に対して 乗客が生存したか（1）死亡したか（0）を予測し、`submission.csv` を提出。

評価方法

- 提出された `submission.csv` の `Survived` 列と、Kaggleが保持する 正解データ を比較
- 評価指標：Accuracy（正解率）
- Leaderboardは2段階（Public ~~40%~~ 50%、Private ~~60%~~ 50%）で構成されている

技術的に学べること

学べるテーマ	内容
データ前処理	欠損値補完、カテゴリ変数処理など
特徴量エンジニアリング	家族人数、敬称（Mr, Mrsなど）、船室デッキなどの作成
モデル学習	ロジスティック回帰 / ランダムフォレスト / LightGBM / XGBoost など
モデル評価	クロスバリデーション、精度評価（Accuracy）
提出プロセス	Kaggleに予測結果を提出し、スコアで比較





Accuracyの計算式は？

✓ Accuracy の計算式

$$\text{Accuracy} = \frac{\text{正しく予測できた件数}}{\text{全体の件数}} = \frac{TP + TN}{TP + TN + FP + FN}$$

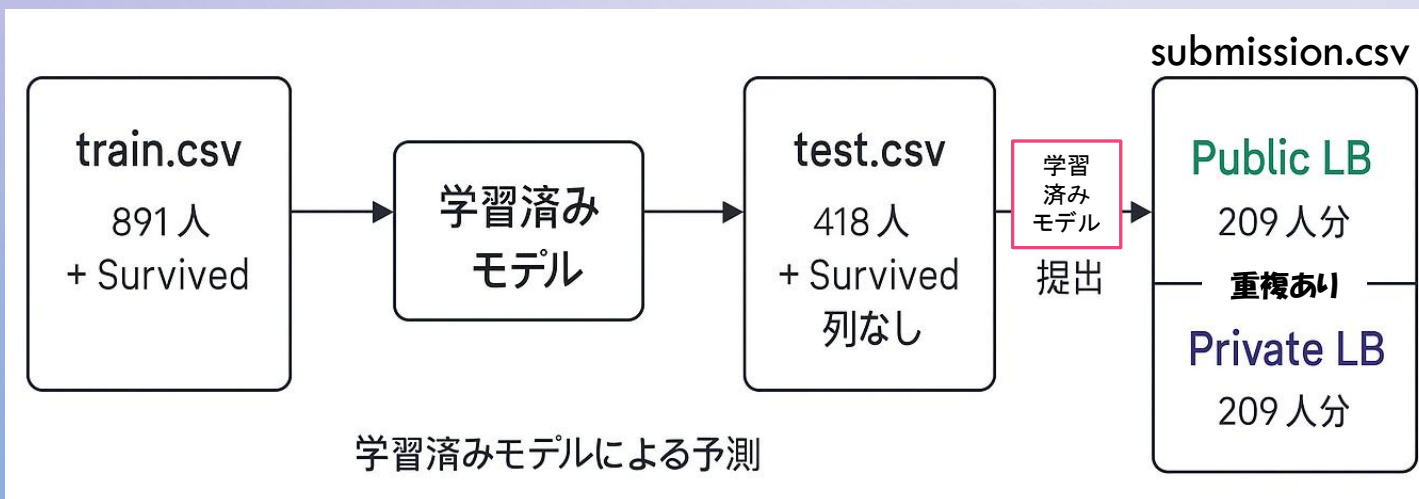
abc 用語の意味（2値分類の場合）

記号	意味
TP (True Positive)	正解は1、生存 → モデルも1と予測（正しく生存と予測）
TN (True Negative)	正解は0、死亡 → モデルも0と予測（正しく死亡と予測）
FP (False Positive)	正解は0、死亡 → モデルが1と誤予測（生存と誤判定）
FN (False Negative)	正解は1、生存 → モデルが0と誤予測（死亡と誤判定）



Kaggle titanic の train データ+test データ、LB 評価データの関係は？

ファイル名	役割	Kaggleでの使用目的
train.csv	学習用データ(正解付き)	あなたのモデルを訓練するために使う
test.csv	テスト用データ(正解なし)	あなたのモデルが予測し、提出する対象
gender_submission.csv	サンプル提出データ	提出フォーマットの例 女性なら生存(Survived=1)、男性なら死亡(Survived=0)」 Kaggle LBで0.76555 (Accuracy 精度: 76.6%)
Private/Public LB用データ	test.csv内部に存在	Kaggleがスコア評価する対象(正解は非公開)



図と表はChatGPTが作成しました。

Kaggleは、test.csv のうち一部だけを評価に使うという 秘密分割 を行っています。

Titanicでは...

Public LB(予選ランキング用) : 50%

Private LB (最終結果用) : 50%

Public LB と Private LB で重複データが存在するときもあります。

Public Leader Board(LB)の例

submission.csv(ファイル名は任意)の提出はここから。
24時間(日本時間 9:00amにリセット)に10回提出できる。

Titanic - Machine Learning from Disaster						
Overview	Data	Code	Models	Discussion	Leaderboard	Submissions
48					1.00000	1 2mo
49					1.00000	3 1mo
50 - 8687		See 8638 More				
8688					0.77511	2 7d
8689					0.77511	3 6d
8690					0.77511	1 7d

Titanicは完全正解データが世の中にあるので、スコア1.0を作ることにも可能。無視しておきましょう。

データ取得

Titanic - Machine Learning from Disaster Submit Prediction

Overview **Data** Code Models Discussion Leaderboard Rules Team Submissions

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

gender_submission.csv (3.26 kB) 📄 📏 ➤

Detail Compact Column 2 of 2 columns ▼

About this file

An example of what a submission file should look like.

These predictions assume only female passengers survive.

PassengerId	# Survived
892	0
1309	1

Data Explorer

93.08 kB

- gender_submission.csv
- test.csv
- train.csv

Summary

- 3 files
- 25 columns

Download All

三つのcsvファイルが入ったzipファイル

ChatGPT 4oで行っています

ChatGPT 4o ▼

📄 新しいチャット

🔍 チャットを検索

📁 ライブラリ

🌐 image generator

⚙️ GPT

🎥 Sora

📁 プロジェクトを新規作成

今日

どんなことをしてますか？

titanic.zip

Zip アーカイブ

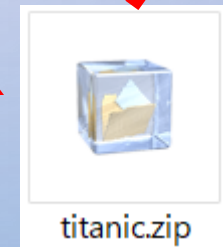
🗑

👁 質問してみましょう

+ ⚙️ ツール

🔊

ChatGPTにDrag & Drop





train.csvを表示して

ぱっと見て、文字の入っている列は
カテゴリ変数だと認識しておきましょう

Train DataFrame

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25		S
2	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38.0	1	0	PC 17599	71.2833	C85	C
3	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.925		S
4	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1	C123	S
5	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05		S
6	6	0	3	Moran, Mr. James	male		0	0	330877	8.4583		Q

Survived がある。
これで学習する。



test.csvを表示して

Survived がない。1列すくない。

Test DataFrame

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292		Q
2	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0		S
3	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875		Q
4	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625		S
5	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875		S



gender_submission.csvを表示して

女性なら生存 (Survived=1)、
男性なら死亡 (Survived=0)」
Kaggle LBで0.76555
(Accuracy 精度: 76.6%)

☐	PassengerId	Survived
1	892	0
2	893	1
3	894	0
4	895	0
5	896	1

ちなみに・・・

0.62200 : submission_all_zero.csv 全員死亡
 0.37799 : submission_all_one.csv 全員生存
 0.50239 : submission_random.csv ほぼ半々
 生存率32.2%がなんとなくわかる。

gender_submissionの0.76555を超えるのが最
初の目標



train.csvはどんなデータ？

◆ カラム一覧と意味

カラム名	説明
PassengerId	乗客ID（一意）
Survived	生存フラグ（0 = 死亡, 1 = 生存）← 目的変数（ターゲット）
Pclass	乗客のチケットクラス（1 = 上級, 2 = 中級, 3 = 下級）
Name	名前
Sex	性別
Age	年齢（欠損あり）
SibSp	タイタニックに同乗している兄弟・配偶者の数
Parch	タイタニックに同乗している親・子供の数
Ticket	チケット番号
Fare	チケット料金
Cabin	キャビン番号（欠損多数）
Embarked	出港地（C = Cherbourg, Q = Queenstown, S = Southampton）

◆ 件数と構成

- 行数（乗客数）：891人
- 列数（特徴量含む）：12列
- 欠損あり: Age, Cabin, Embarked

◆ ターゲット変数

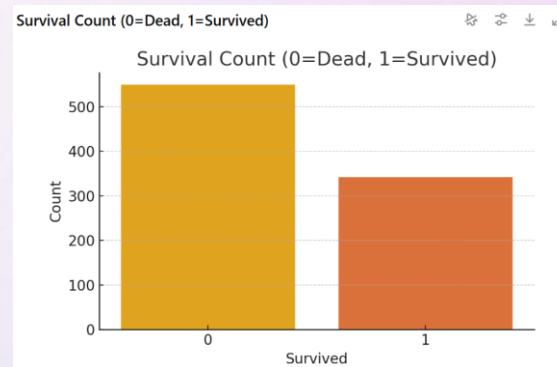
- Survived は 0 or 1 の 二値分類タスク
- モデルはこの列を他の情報から予測する



EDAして！

	Column	Missing Count
0	Cabin	687
1	Age	177
2	Embarked	2
3	PassengerId	0
4	Survived	0

Cabinはほとんどが欠損値



死亡者数の方が多い

Train Data Summary													
		PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	count	891.0	891.0	891.0	891	891	714.0	891.0	891.0	891	891.0	204	889
2	unique				891	2				681		147	3
3	top				Braund, Mr. Owen Harris	male				347082		896 898	S
4	freq				1	577				7		4	644
5	mean	446.0	0.3838383838383838	2.308641975308642			29.69911764705882	0.5230078563411896	0.38159371492704824		32.204207968574636		
6	std	257.3538420152301	0.4865924542648575	0.836071240977049			14.526497332334042	1.1027434322934317	0.8060572211299483		49.6934285971809		
7	min	1.0	0.0	1.0			0.42	0.0	0.0		0.0		
8	25%	223.5	0.0	2.0			20.125	0.0	0.0		7.9104		
9	50%	446.0	0.0	3.0			28.0	0.0	0.0		14.4542		
10	75%	668.5	1.0	3.0			38.0	1.0	0.0		31.0		
11	max	891.0	1.0	3.0			80.0	8.0	6.0		512.3292		



次のEDAをして！

Sex-Wise Survival Rate

	Sex	Survived
1	female	0.7420382165605095
2	male	0.18890814558058924

Pclass-Wise Survival Rate

	Pclass	Survived
1	1	0.6296296296296297
2	2	0.47282608695652173
3	3	0.24236252545824846

Age Group-Wise Survival Rate

	AgeBin	Survived
1	(0, 10]	0.59375
2	(10, 20]	0.3826086956521739
3	(20, 30]	0.3652173913043478
4	(30, 40]	0.44516129032258067
5	(40, 50]	0.38372093023255816

◆ 性別ごとの生存率

- 女性の生存率が大幅に高く、男性よりも優遇されていたことがわかります。

◆ チケットクラス(Pclass)ごとの生存率

- 上級クラス(1等)ほど生存率が高く、3等の乗客はかなり低い傾向があります。

◆ 年齢層ごとの生存率(10歳ごとに区切り)

- 子ども(0-10歳)の生存率が高い(約59%)
- 若年～中年層(20-50歳)の生存率は全体的に低め(35-45%)



さらに次のEDAをして！

◆ 性別 × チケットクラス (Sex × Pclass) の生存率

- 1等女性是非常に高い生存率 (90%以上)
- 3等男性は極めて低い生存率 (10%未満)
- → 社会的地位と性別で強く生存率が左右されていたと推測されます

◆ 同乗していた家族数の影響

SibSp (兄弟・配偶者) ごと：

- 単独乗船 (SibSp=0) は生存率が低い (~34%)
- SibSp=1あたりが生存率が高く、その後はやや減少

Parch (親・子供) ごと：

- Parch=1~3程度までは生存率が高め (50%以上)
- Parch=4以上は急激に低下 (Parch=4では生存率0)

◆ Fare (運賃) ごとの生存率

- 運賃が高いグループほど、生存率が高い傾向
 - 最も高額な運賃帯は生存率が60%超
 - 最も低い運賃帯では生存率が約25%程度
 - → 社会的階級の影響を反映

◆ Embarked (乗船港) ごとの生存率

- C港 (Cherbourg) から乗った乗客の生存率が最も高い
- S港 (Southampton) は生存率がやや低め

◆ 相関係数マトリクス (Survivedとの関係)

特徴量	相関係数 (Survivedとの関係)
Pclass	-0.34 (負の相関：等級が高いほど生存率高)
Fare	0.26 (正の相関)
Age	-0.08 (弱い負の相関)
SibSp/Parch	ほぼ無相関～弱い正相関



さらに深掘りして！

◆ Cabin情報の有無と生存率

- Cabin 情報が ある 乗客の生存率は 高い
 - ⇒ 上級クラスの乗客ほどCabinが割り当てられており、生存率も高い

◆ 名前の中の敬称（Title）と生存率

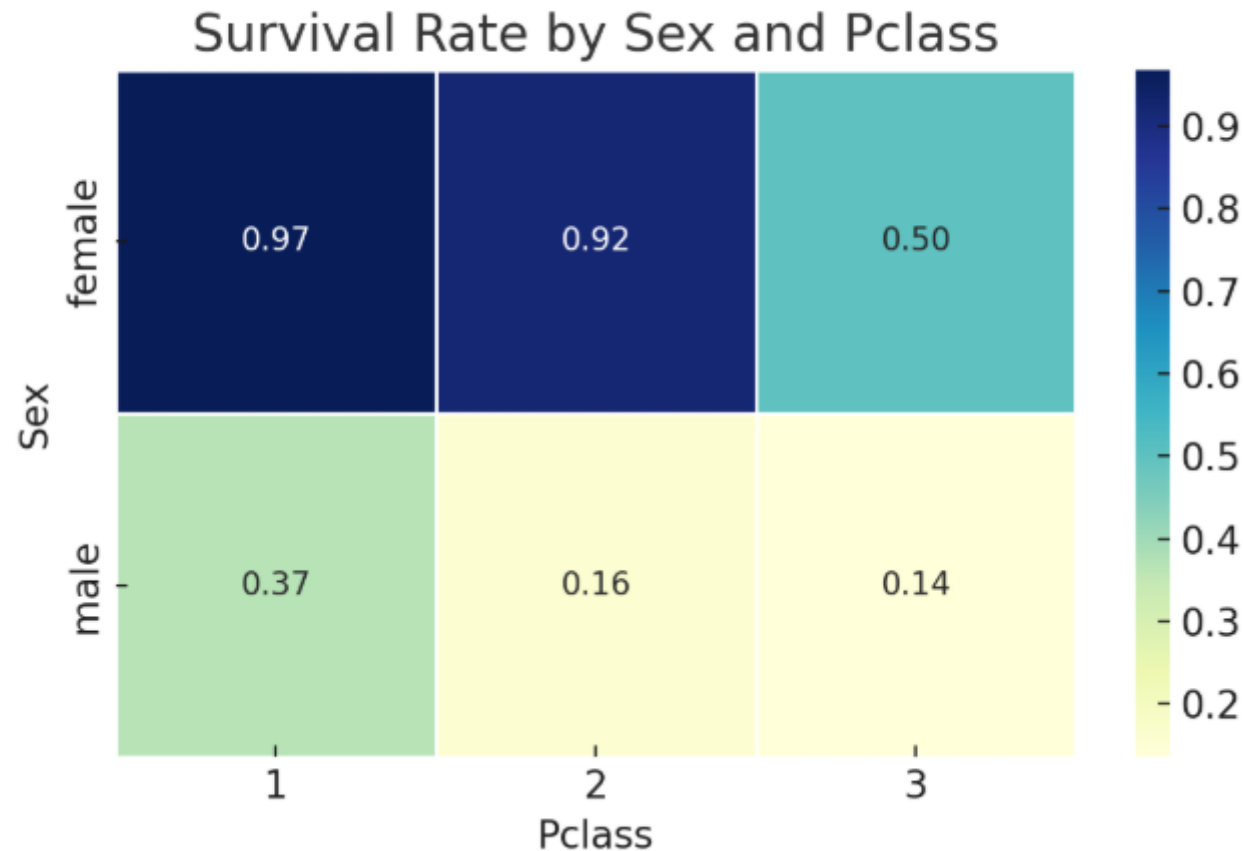
- Mr, Mrs, Miss, Master などの敬称を抽出し、敬称ごとに生存率を確認
- 結果（抜粋）：
 - Mrs（既婚女性）や Miss（未婚女性）は生存率が高め
 - Mr（男性）は比較的低め
 - Master（若年男性、特に少年）は高い
 - Dr, Col, Capt などの特殊敬称はサンプル数が少ないが注目点あり



ヒートマップで性別×Pclassごとの生存率を表示して

途中、ChatGPTがお勧めしてきた。

Survival Rate By Sex And Pclass



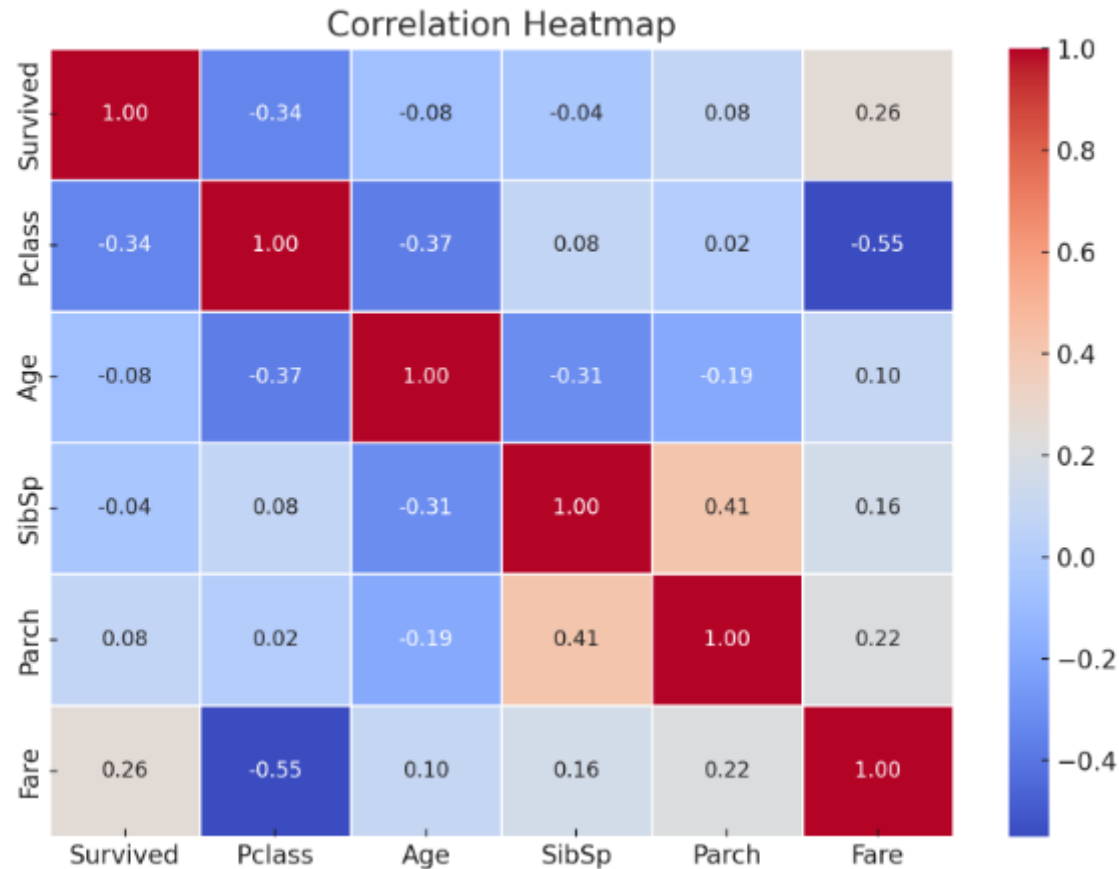
🔍 読み取りポイント

- 1等女性 は生存率が非常に高い（ほぼ 1.00）
- 3等男性 は生存率が非常に低い（0.10 以下）
- クラスが高いほど、そして女性であるほど生存率が高い傾向



お勧めのヒートマップを表示して

Correlation Heatmap



🔍 注目ポイント：

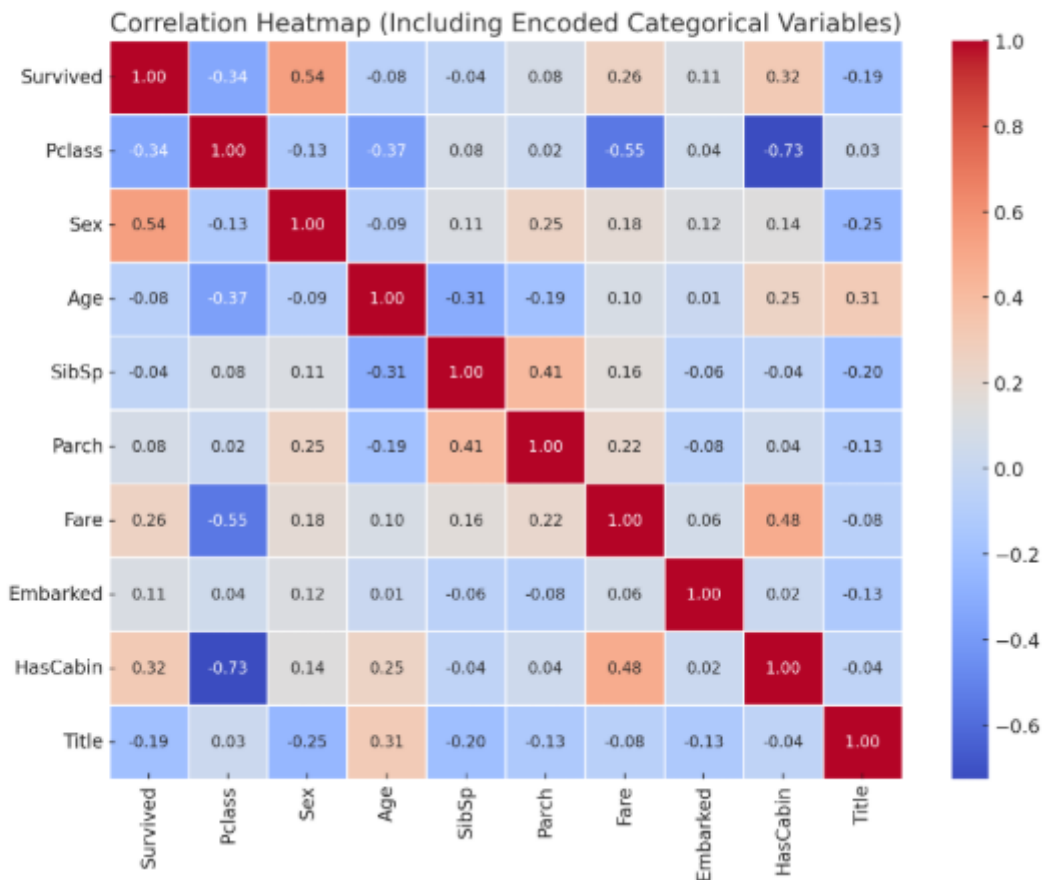
- Pclass（客室等級）と Survived の負の相関が強め（-0.34）：
 - 等級が高い（1等）ほど生存しやすい
- Fare（運賃）と Survived にも正の相関（0.26）：
 - 高額なチケットの方が生存率が高い傾向
- Age, SibSp, Parch は比較的相関が弱い

他にも、カテゴリ変数をエンコードして含めた拡張相関マトリクスを表示できます。



カテゴリ変数も含めた、相関ヒートマップを表示して

Correlation Heatmap (Including Encoded Categorical Variables)



カテゴリ変数を数値に変換 (簡易エンコーディング)

```
encoded_df = train_df.copy()
encoded_df['Sex'] = encoded_df['Sex'].map({'male': 0, 'female': 1})
encoded_df['Embarked'] = encoded_df['Embarked'].map({'S': 0, 'C': 1, 'Q': 2})
encoded_df['HasCabin'] = encoded_df['Cabin'].notnull().astype(int)
```

敬称 (Title) をラベルエンコーディング

```
encoded_df['Title'] = encoded_df['Title'].astype('category').cat.codes
```

🔍 注目ポイント

- Sex (性別) と Survived : 強い正の相関 (0.54)
 - 女性 (1) の方が生存率が高い
- HasCabin (Cabin情報の有無) と Survived : 中程度の正の相関
- Pclass、Fare も引き続き重要
- Title (敬称) もある程度の関連がある



特徴量エンジニアリングを行って！

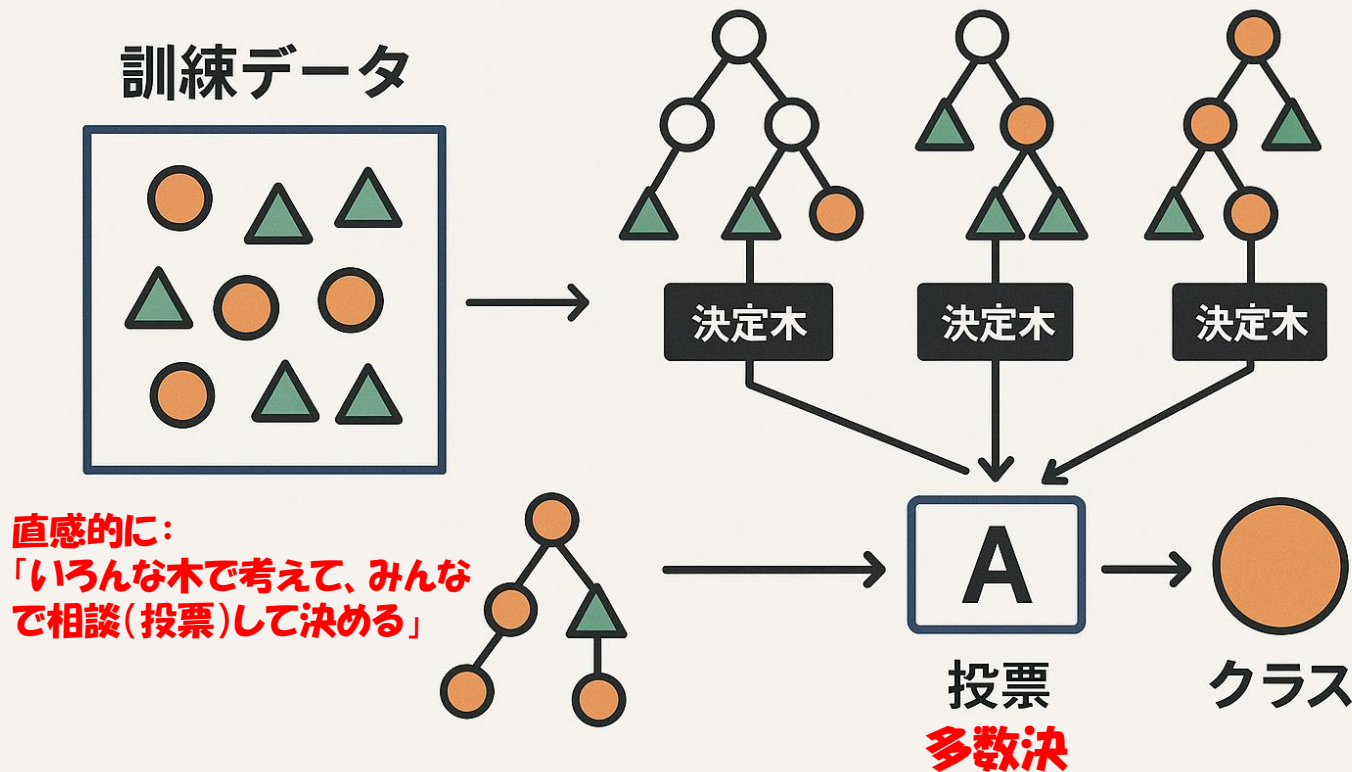
新たに追加・変換した特徴量

特徴量名	内容
Sex	male=0, female=1 に変換
Embarked	S=0, C=1, Q=2 に変換
HasCabin	Cabin情報の有無（1=あり, 0=なし）
Title	名前から抽出した敬称（カテゴリ数値化）
FamilySize	同乗家族人数（SibSp + Parch + 1）
IsAlone	単独乗船（FamilySize==1）
AgeBin	年齢のビン分割（0～6のカテゴリ）
FareBin	運賃の四分位カテゴリ化（0～3）



ランダムフォレストのわかりやすい説明図を作って

ランダムフォレスト



ChatGPTには最初からRandom Forestが入っている。

■ 左: 訓練データ

- ・四角の中にある「オレンジの丸」と「緑の三角」は、それぞれ異なるクラス(分類カテゴリ)を表しています。
- ・この訓練データからモデルを学習させます。

→ 中央: 複数の決定木(Decision Trees)

- ・ランダムフォレストは 多数の決定木 を使います(図中に3本描かれている)。
- ・各決定木は訓練データの一部(フートストラップサンプル)と一部の特徴量のみを使って構築されるため、「ランダム」な性質を持ちます。

↓ 下部: 投票(Voting)

- ・各決定木が新しいデータのクラスを「予測」します。
- ・それぞれの予測結果が「投票(多数決)」にかけられます。

→ 右: 最終的なクラス

- ・最も票を得たクラス(例: オレンジの丸)が、ランダムフォレスト全体としての予測結果になります。

多数決により高い精度: 単独の決定木より過学習に強く、汎化性能が高い。

ランダム性により多様性: 特徴量やデータのランダム選択によって、多様な決定木ができる。

並列化しやすい: 木ごとの学習・予測は独立しているため、高速化が可能。



これらをもとにランダムフォレストでsubmission.csvを作成して！

```
File ~/.../python3.11/site-packages/sklearn/utils/validation.py:146, in _assert_all_finite
    124     if (
    125         not allow_nan
    126         and estimator_name
    (...) 130         # Improve the error message on how to handle missing values in
    131         # scikit-learn.
    132         msg_err += (
    133             f"\n{estimator_name} does not accept missing values"
    134             " encoded as NaN natively. For supervised learning, you might want"
    (...) 144             "#estimators-that-handle-nan-values"
    145         )
--> 146     raise ValueError(msg_err)
    148 # for object dtype data, we only check for NaNs (GH-13254)
    149 elif X.dtype == np.dtype("object") and not allow_nan:

ValueError: Input X contains NaN.
RandomForestClassifier does not accept missing values encoded as NaN natively. For supervised l
```

エラーの原因は、`X_train` に欠損値（NaN）が含まれているため、`RandomForestClassifier` がそれを処理できずに停止しています。

✅ 解決策：`X_train` の欠損値を補完

- `Age` や `Embarked` などが `train.csv` にも一部欠損していたことが原因です。
- 今から補完して、再度 `RandomForestClassifier` を使って `submission.csv` を作成します。

続けます。👉

途中、エラーで落ちるが、自分で欠損値補完を行う修正を入れて、継続！

`X_test` 側にも欠損値（NaN）が残っていたため、再び `RandomForestClassifier` がエラーになりました。

もう一度エラー。今度はtest側も欠損値補完。

さらに継続...

`RandomForestClassifier` により予測を行い、`submission.csv` を正常に作成しました。

📄 ダウンロードリンク: [submission.csv](#)

このファイルをそのまま Kaggle に提出可能です。

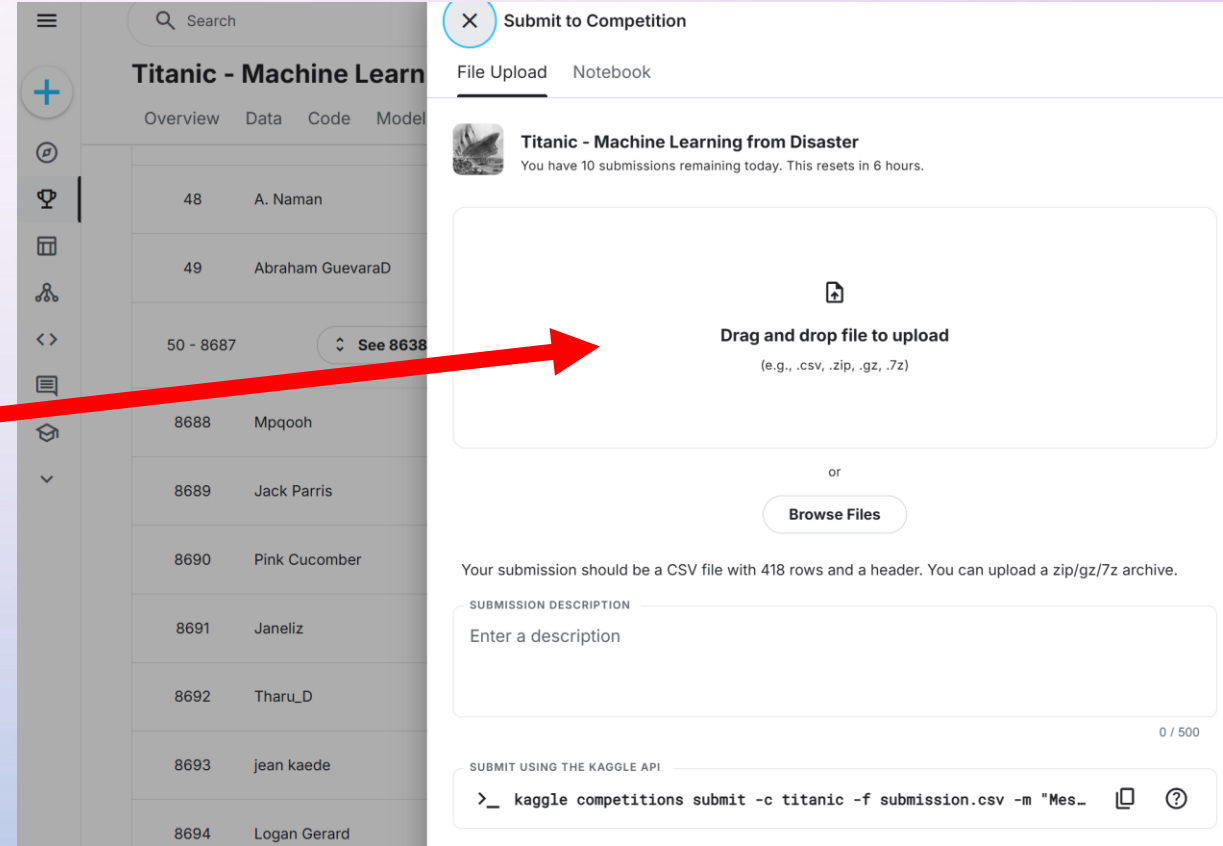
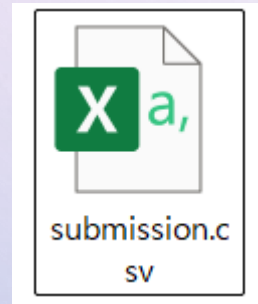
成功！

提出: *submission!*

RandomForestClassifier により予測を行い、`submission.csv` を正常に作成しました。

ダウンロードリンク: [submission.csv](#)

このファイルをそのまま Kaggle に提出可能です。



が、しかし！
gender_submission.csv
(LB: 0.76555)
よりもスコアが低い！



submission.csv

Complete · now · ChatGPT, feature engineering, random forest

0.73444



0.73444 だった。gender_submission.csvよりもスコアが低いなぜ？

- スコアが 0.73444 で、gender_submission.csv(おそらく 0.76555 程度)より低い理由は、主に以下のような要因が考えられます：

✓ gender_submission.csv は「女性は生存、男性は死亡」のルールベース予測

- 非常にシンプルですが、Titanicデータでは 女性の生存率が高く、男性の生存率が低い ため、このルールだけでも そこそこ高い精度 を出せます。
- 実際、Kaggle公式のベースラインとしても提示されています。

🔍 あなたの RandomForest モデルのスコアが低い理由

1. 過学習 or 過剰な特徴量

- Title, FareBin, AgeBin, IsAlone などの特徴量を多数導入しましたが、適切に効いていない or ノイズになっている可能性があります。

2. 欠損値補完の方法が単純

- 年齢や運賃などを「中央値」や「最頻値」で一律補完していますが、より賢い方法（例: Sex や Pclass を使ったグループ中央値補完）だと精度が上がります。

3. 特徴量のスケールや調整がされていない

- RandomForestはスケールに強いですが、AgeBin や FareBin などカテゴリ化が精度を落としている可能性も。

4. パラメータチューニング未実施

- n_estimators, max_depth, min_samples_split などのチューニングが行われていません。
- デフォルト設定のままでは精度が出ないケースも多いです。

5. 学習データとテストデータで分布がズレている

- 特に FareBin や Title などがテスト側に未出現カテゴリを含んでいた可能性も。

✓ 改善案（提案）

1. シンプルな特徴量に絞る

- Pclass, Sex, Age, Fare などだけでまずは構成

2. モデルをチューニング

- RandomForest のパラメータ調整
- または LightGBM に変更（高性能）

3. クロスバリデーションで検証

- train.csv 内で交差検証して精度を確認
- Public LB を基準にせず過学習を防ぐ

← END 結論

gender_submission.csv は実は「データに対して最適なバイアスのルール」に基づいていて、意外と強いベースラインです。

一方でモデルを使う場合は適切な特徴量選択・欠損処理・チューニングが必要です。

希望があれば、「精度を gender_submission.csv 以上に上げるための改良案」をコード付きで出せます。どうしますか？

続く！！