**۱. مقدمه**

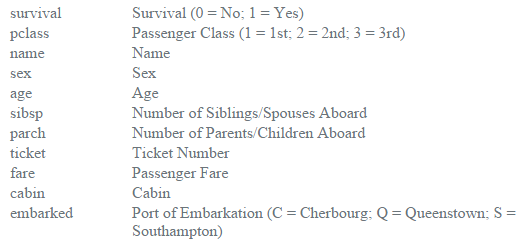
در حادثه ی برخورد کشتی بخار بزرگ تایتانیک به کوه یخ، که در تاریخ 15 آوریل 1912، از بندر ساوت همپتون انگلستان به مقصد نیویورک آمریکا در سفر بود، 1514 نفر از 2224 مسافر و خدمه ی کشتی، کشته شدند؛ که یکی از دلایل اصلی آن عدم تعبیه ی قایق نجات، به تعداد لازم بود.

هر چند که می توان شانس را یکی از عوامل تأثیر گذار در نجات یافتن افراد بیان کرد؛ امّا برای برخی افراد، مانند زنان و بچّه ها

و افرادی که در قسمت های با درجه ی بالاتر جای داشتند، احتمال بیشتری وجود داشت که نجات پیدا کنند.

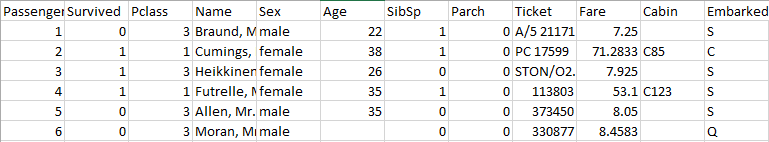
داده های مسئله، به صورت فایل های 3csv، در اختیار کاربران قرار داده شده اند.

در شکل زیر ، اطّلاعاتِ داده شده و توضیحات آن ها آورده شده است.



شکل 1- اطّلاعات داده شده راجع به مسافران

قسمتی از داده ها نیز در شکل پایین، قابل مشاهده می باشند.



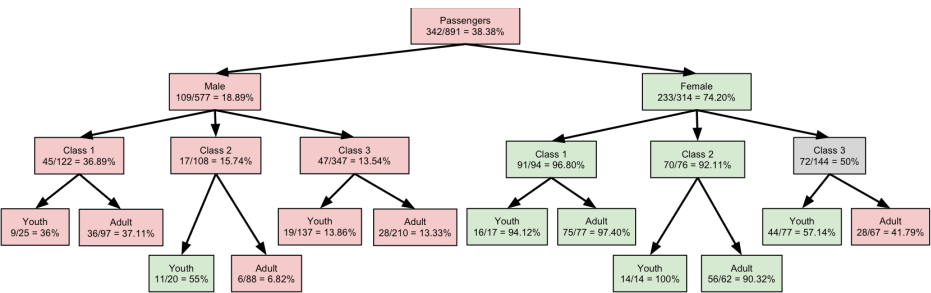
شکل 2- قسمتی از داده ها

همان طور که در شکل 2 مشخّص است، برای برخی افراد، بعضی ویژگی ها، دارای مقدار نمی باشند.

مجموعه ی داده ها، دارای 819 نمونه می باشد.

مجموعه ی داده شده جهت تست، دارای 418 نمونه است.

در صورتی که یک بررسی اوّلیه بر روی داده ها انجام دهیم، نتایج شکل زیر به دست می آیند.



شکل 3- بررسی اوّلیه

همان طور که مشاهده می کنید، جنسیت مسافر، در این که نجات پیدا می کند یا خیر، بسیار تأثیر گذار می باشد؛ چنان که 74% زنان زنده ماندند؛ در حالی که تنها 18% مردان توانستند نجات پیدا کنند.

این اطّلاعات، یک درک کلّی از شرایط مسئله، در اختیار ما قرار می دهد.

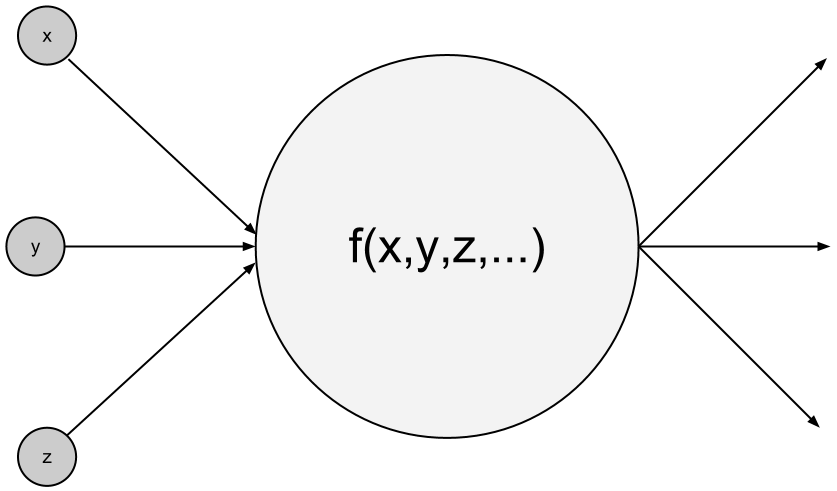
در منابع ذکر شده، روش های جنگل تصادفی، درخت تصمیم، شبکه های عصبی، ماشین بردار پشتیبان، Naive Bayes، افزایش شیب دار و... استفاده شده اند.

اِن شاءَ اللّه ما در این پروژه، سعی خواهیم کرد روش شبکه های عصبی4 را در حدّ امکان و قابل فهم بررسی کنیم.

ابتدا کلّیت روش شبکه های عصبی را توضیح داده، و سپس این روش را برای مسئله ی داده شده به کار خواهیم برد.

**شبکه های عصبی**

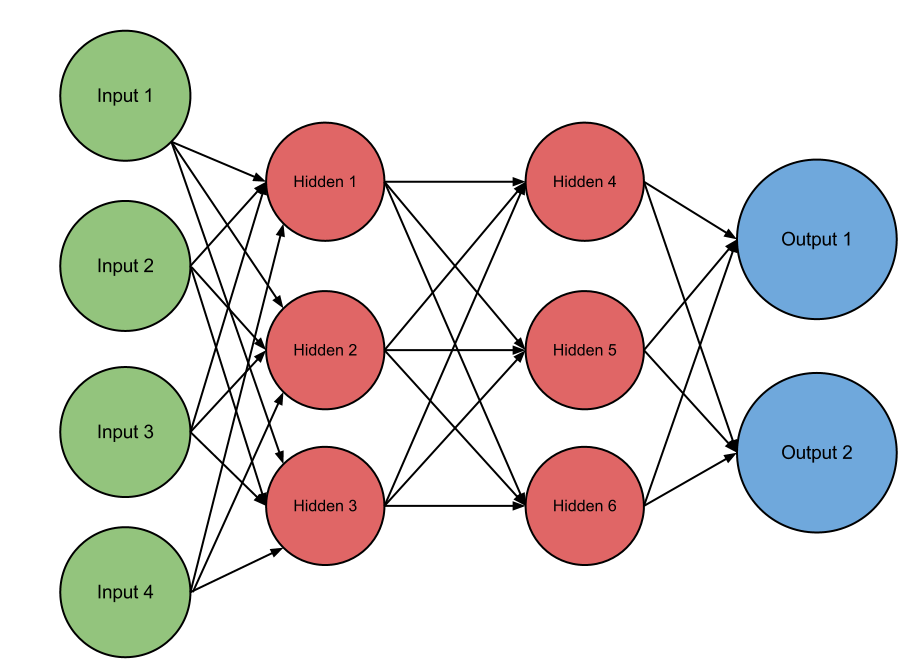
شبکه‌های عصبی، مدلی می‌باشند که با الهام از سیستم‌های زیستی به وجود آمده‌اند. در یک شبکه از عصب‌های زیستی مانند مغز، عصب‌ها سلول‌های جداگانه‌ای می‌باشند که ورودی دریافت کرده؛ بر روی آن عملیات انجام می‌دهند؛ سپس حاصل را به عصب یا عصب‌های دیگری که به آنها متّصل می‌باشند، منتقل می‌نمایند. این را رفتار را ‌می‌توان با استفاده از یک عصب مصنوعی مدل سازی نمود.



شکل 4- یک عصب در شبکه ی عصبی

شکل بالا یک عصب را نشان می‌دهد که ورودی‌هایی را دریافت کرده؛ با استفاده از f، آن ها را پردازش می‌کند و خروجی‌هایی را تولید می‌نماید.

می‌توان مجموعه‌ای از این عصب‌ها را در چند لایه به صورتی که در شکل زیر مشاهده می‌کنید به کار برد.



شکل 5- یک شبکه ی عصبی با چهار لایه

یک لایه، مربوط به ورودی ( input ) می‌باشد. تعداد عصب‌ها در این لایه معمولاً با تعداد ویژگی‌ها ( در اینجا سن، کلاس، جنسیت، ...) برابر می‌باشد.

یک شبکه ی عصبی می‌تواند فاقد لایه ی میانی ( hidden ) باشد یا شامل یک یا چند لایه از این نوع باشد ( در قسمت آزمایش، یک لایه ی میانی در نظر گرفته شده است ).

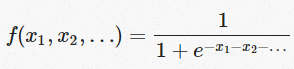
لایه ی آخر که لایه خروجی ( output ) نام دارد نیز می‌تواند شامل یک یا چند عصب، بسته به طرّاحی شبکه باشد ( در قسمت آزمایش، یک عصب برای این لایه در نظر گرفته شده است ).

هم چنین در این شبکه، هر ارتباط بین عصب‌ها دارای یک وزن می‌باشد. بنا بر این برای مثال، در شکل بالا، تابع فعّال سازی برای عصب Hidden 1 به شکل زیر می‌باشد:



شکل ۶

یکی از توابع فعّال سازی معمول، تابع logistic می‌باشد؛ که یکی از مزایای آن این است که قابل مشتق‌گیری ‌می‌باشد. در زیر فرمول تابع را مشاهده می کنید.



شکل ۷

روش یادگیری: داده ها در یک شبکه ی عصبی، در دو حالت جریان پیدا می‌کنند. حالت اوّل زمانی است که شبکه در حال یادگیری می‌باشد. حالت دوم زمانی است که شبکه در حالت عادی ( پس از یادگیری ‌می‌باشد ). داده‌ها از طریق عصب‌های ورودی وارد می‌شوند و پس از عبور از لایه‌های میانی وارد عصب‌های خروجی ‌می‌گردند.

این روش طراحی feedforward نامیده می‌شود. شبکه‌های عصبی با استفاده از یک عنصر باز خوردِ یادگیری که backpropagation نامیده می‌شود یاد گیری می‌کنند.

در این روش، مقادیری که در شبکه تولید می‌شود، با مقادیری که ‌می‌بایست تولید شود مقایسه می‌گردد و از تفاوت بین آن ها برای اصلاح ورژن‌های بین ارتباطات استفاده می‌شود.

این فرایند از لایه ی خروجی شروع می شود؛ سپس به لایه‌های میانی و سرانجام، به سمت لایه ی ورودی به طرف عقب می‌رود.

**۲. کارهای مرتبط**

در رابطه پیاده سازی پیشبینی نجات یافتگان با شبکه عصبی چند کار صورت گرفته که در سایت kaggle قرار دارند.

در پروژه اول توسط کتابخانه keras پیشبینی نجات یافتگان با شبکه عصبی را پیاده سازی کرده است و بر روی داده های train تغییراتی مثل نگاشت جنسیت و هر کدام از شهر های که مسافر سوار کرد به یک عدد و پر کردن مقادیر خالی با 0 داده است. [۶]

در پروژه دوم از کتابخانه tensorflow و tflearn استفاده شده است و بر روی داده های train دستکاری بهتری روی سن های خالی دارد و تغییرات جزیی دیگری نیز اعمال کرده است. [۵]

در پروژه هوش مصنوعی دیگری در همین سایت بر روی مساله پیشبینی نجات یافتگان کار شد که به دلیل اینکه از چند روش یادگیری ماشین استفاده شد خوب بود، ولی چون عملیات زیادی بر روی داده های train انجام ندادند نتایج کاملا خوبی در روش شبکه های عصبی بدست نیاوردند. [9]

**۳. آزمایش‌ها**

**1-3- پیش نیاز**

جهت اجرای کد برنامه که در گیت هاب قابل دسترسی است، نیاز به ماژول‌هایی می‌باشد که در فایل requirements.txt موجود در فایل ذکر شده اند.

برای نصب ماژول‌ها دو راه وجود دارد:

راه اوّل: همان ابتدا با استفاده از دستور زیر ماژول‌های موجود در فایل را نصب کنید. این روش معمولاً به خطا می‌انجامد؛ چرا که بعضی از ماژول‌ها به راحتی قابل نصب نمی‌باشند:

pip install –r requirements.txt

راه دوم: نرم‌افزار miniconda متناسب با سیستم عامل خود را از [اینجا](https://conda.pydata.org/miniconda.html) دریافت و نصب نمایید. Conda یک برنامه ی package manager می‌باشد که روش سریعی را جهت استفاده از package ها فراهم می‌آورد.

پس از این که مراحل نصب تمام شد، با استفاده از دستور conda create -n myenv python یک virtual environment بسازید و با دستور

activate myenv، آن را فعّال کنید ( می‌توانید از اسم دل خواه خود به جای myenv استفاده نمایید ).

اکنون با دستور conda install numpy ماژول numpy را نصب کنید.

سایر ماژول ها با استفاده از دستوری که در روش اوّل ذکر شد، قابل نصب می‌باشند.

**2-3- پیش پردازش**

داده‌های مسئله که در سایت مسابقه دریافت شده اند،در github پروژه قابل دسترسی می‌باشند.

برای این که داد‌‌ه‌ها در الگوریتم‌های به کار رفته قابل استفاده باشند، لازم است که یک عملیات پیش‌پردازش بر روی آن ها انجام شود.

برای خواندن و پردازش فایل‌های csv از کتابخانه pandas پایتون استفاده شده است. داده‌های مسئله دارای یک فایل train.csv می‌باشد و داده های مربوط به test را از روی داده ی اولیه بدست می آوریم و جهت سادگی کار، داده‌های train.csv را که حاوی اطّلاعات کامل می‌باشند، به دو قسمت، به صورت 30% برای تست و 70% برای یادگیری تقسیم می‌کنیم تا فرایند آزمایش، بصورت آفلاین قابل انجام ‌باشد.

بسیاری از داده‌های داده شده، دارای مقادیر غیر عددی می‌باشند که لازم است جهت کارکرد صحیح روش‌ها، به مقادیر عددی تبدیل شوند.

هم چنین مقادیر بسیاری از سطر‌های جدول دارای مقادیر null می‌باشند که این مقادیر، قابل قبول نمی‌باشند.

ستون cabin را به علّت تعداد زیاد مقادیر null، ستون name و ستون ticket را نیز به این دلیل که برای هر فرد، منحصر به فرد می‌باشد و یافتن روابط در آن ها دشوار می‌باشد از داده‌ها حذف می‌کنیم.

ستون‌های دارای مقادیر غیر عددی یا ستون‌هایی که عدد آن ها نشان دهنده ی مقدار آن ها نمی باشد را نیز به ستون‌هایی، به تعداد مقادیر مختلفی که می‌توانند داشته باشند، تقسیم می‌کنیم؛

برای مثال ستون Pclass که می‌تواند دارای مقادیر 1، 2 یا 3 باشد، به سه ستون 1 و 2 و 3 تقسیم می‌شود که برای فردی که در کلاس 3 بوده، این ستون‌ها به ترتیب مقادیر 0 و 0 و 1 خواهند داشت.

این عملیات را به طور مشابه برای ستون های Sex و Embarked هم اعمال می کنیم. با این کار به جای این دو ستون به ترتیب ستون های female,male و ستون های C,Q,S قرار می گیرند.

نسخه ی قبلی این پروژه در سایت بوته، طبق روش هایی که در تابع get\_manipulated\_train آمده، برای تعیین سن مسافرانی که محتوایی ندارد از تابع interpolate استفاده شده بود . (تعداد ۱۷۷ مسافر بدون سن هستند. برای پیشبینی بهتر توسط شبکه های عصبی بهتر است داده های مناسب تر به عنوان ورودی بدهیم پس باید فیلد های خالی را پر کنیم.)

در حال حاضر ابتدا برای این کار مقادیر random بین age\_avg-age-std و age\_avg+age\_std را برای حدس زدن سن های خالی انتخاب کردیم، که نسبت به روش قبل برای تخمین سن بهتر بود چون اعداد را تصادفی بین حدود خاصی نسبت به میانگین تعیین می کرد.

در مرحله بعد تخمین سن را بهتر نمودیم. با توجه به ستون های sex,pclass,title (نحوه بدست آوردن title هر فرد در ادامه توضیح داده شده است) از روی داده اولیه و میانگین گرفتن می توان به اطلاعاتی رسید .

که از این اطلاعات می فهمیم برای مسافرانی که سن ندارند به ازای جنسیت،pclass و عنوانشان چه سنی را باید به آن ها تخصیص بدهیم.

یک معیار دیگر برای بهبود عملیات یادگیری نرمال سازی5 داده ها است، به همین دلیل برای مثال بازه های مختلف سن را به اعداد 0,1,2,3,4 نگاشت می دهیم یعنی به سن های کمتر از ۱۶ مقدار ۰، به سن های بین ۱۶ و ۳۲ مقدار ۱، به سن های بین ۳۲ تا ۴۸ مقدار ۲ و... می دهیم.

به همین ترتیب عملیات نرمال سازی را برای مقدار کرایه های مسافران کشتی نیز اعمال می کنیم.

دو ستون sibSp,Parch به ترتیب به معنی تعداد والدین،فرزندان و تعداد خواهر،برادر و همسر هر فرد در کشتی می باشند. به این ترتیب می توان تعداد اعضای خانواده هر فرد را با جمع کردن فیلد های sibSp,Parch با عدد ۱(یعنی خودش) بدست آورد و نتیجه را در ستون FamilySize هر فرد بنویسیم. سپس رابطه بین تعداد اعضای خانواده و زنده ماندن افراد را تحلیل میکنیم .

با توجه به مقادیر موجود می توان فهمید افرادی که تنها در کشتی بودند با احتمال بیشتری از بین رفته اند. با توجه به این که تعداد افراد تنها در کشتی ۵۳۷ نفر هستند (که بیشتر از نصف می شود) می توان یک ستون به نام isAlone به منظور یادگیری بهتر تعریف نماییم.

با بررسی دقیق تر اطلاعات مسافران می فهمیم در ستون نام ، عنوان هر شخص نیز درج شده است که می تواند مورد استفاده قرار گیرد. ما توسط تابع get\_title با کمک عبارات منظم این عناوین را از اسم هر شخص در می آوریم و در ستون title قرار می دهیم.

با تحلیل این القاب می فهمیم در مجموع ۱۷ عنوان مختلف وجود دارد که ۴ لقب Mr,Miss,Mrs,Master خیلی بیشتر نسبت به بقیه تکرار شده اند.

برای بهینه کردن عملیات یادگیری خوب است انواع هر ستون را کمتر و مقادیر عددی به هر کدام بدهیم ، بنابراین می توان بقیه عنوان های کمتر تکرار شده را مقدار Rare بهشان بدهیم. پس در مجموع پنج دسته عنوان Mr,Miss,Mrs,Master,Rare داریم که به هر کدام یک عدد بین ۱ تا ۵ تخصیص می دهیم. با تحلیل رابطه ی بین عنوان و زنده ماندن هر فرد جدولی را تهیه میکنیم .

با توجه به این که درصد زنده ماندن بر اساس هر عنوان به طور کلی زیاد نزدیک به نصف نیست، پس ما معیار عنوان را به درستی به داده یادگیری مان اضافه کردیم.

حال به ورودی ای که در نهایت آماده یادگیری توسط شبکه های عصبی می شود ، میرسیم .

در این جدول مقادیر همه ی ستون ها عددی و در بازه ی مشخص می باشد که آماده ی استفاده توسط روش های مختلف یادگیری ماشین مثل درخت تصمیم6 ، جنگل تصادفی7، شبکه های عصبی و ... می باشد.

**3-3- بررسی نتایج توسط روش شبکه های عصبی**

در پیاده سازی این قسمت، از ماژول neurolab در زبان پایتون و شبکه ی feed forward multilayer perceptron استفاده شده است.

در ماژول ذکر شده، الگوریتم‌های زیادی برای یادگیری در نظر گرفته شده است.

اگر مقدار عصب بیشتر از 0.5 بوده باشد، فرد، زنده و در غیر این صورت، غرق شده پیش بینی شده است.

قبل از بررسی نتایج، لازم است که عبارت epoch توضیح داده شود:Epoch یک تکرارِ کاملِ روندِ یادگیریِ داده‌های داده شده، جهت یادگیری می‌باشد.

**۴. تکمیل پیاده سازی و بهبود نتایج**

برای بهبود نتایج می توان داده های موجود برای مساله پیشبینی نجات یافتگان را بیشتر تحلیل کرد تا به گونه ای داده های یادگیری را تغییر دهیم که باعث بهتر شدن درصد های بدست آمده توسط روش شبکه های عصبی شویم.

کد های مربوط به این فاز هم در گیت هاب پروژه می باشد.

با بررسی دقیق داده ها می توان فهمید هنوز محل سوار شدن به کشتی برای دو مسافر با PassengerId های ۶۲ و ۸۳۰ خالی می باشند برای پر کردن این مقادیر می توان به صورت تصادفی یک محل سوار شدن برای آن ها تخصیص بدهیم که زیاد مناسب نیست. بنابراین می توان به کرایه و pclass این دو مسافر توجه کرد و این مقادیر را با بقیه مسافران مقایسه کرد تا محل سوار شدن این دو مسافر هم بتوانیم حدس بزنیم.

با توجه به این که دو مسافر مد نظر ما هر دو در کلاس ۱ هستند و کرایه ۸۰ دلار پرداخت کرده اند، با رجوع به نمودار بالا می توان فهمید هر دو از Charbourg سوار شده اند.(خطچین قرمز نشان دهنده کرایه ۸۰ دلار می باشد.)

در مرحله بعد می توان دو ستون وابسته به سن دیگر Child و Mother را به ستون ها اضافه کرد. به این ترتیب که هر مسافری که سنش کمتر از ۱۸ باشد پس بچه است و هر مسافری که سنش بیشتر از ۱۸ ، زن و فیلد Parch اش بیشتر از یک باشد و همچنین عنوانش Miss نباشد پس مادر است.

**1-4- تکرار بررسی نتایج توسط روش شبکه های عصبی**

نسبت به فاز قبل این بار تعداد epoch ها را تا ۱۰۰۰ افزایش دادیم ، هر چه تعداد epoch ها بیشتر باشند به دلیل این که تکرار های کامل بیشتر می شوند یادگیری در شبکه های عصبی بهتر صورت می گیرد.

این بار معیار دیگری به نام نرخ یادگیری هم که هنگام استفاده از تابع train به همراه تعداد epoch ها می توان مقدار آن را تعیین کرد را نیز در نظر می گیریم.

مدل های یادگیری عمیق به طور معمول توسط یک بهینه ساز گرادیان کاهشی تصادفی آموزش داده می شود. همه این الگوریتم ها به شما اجازه می دهد نرخ یادگیری را تنظیم کنید. این پارامتر ، بهینه ساز را به جایی می برد که وزن را در جهت گرادیان برای یک mini-batch حرکت دهد.

اگر نرخ یادگیری کم باشد، آموزش قابل اعتمادتر است، اما بهینه سازی زمان زیادی را صرف خواهد کرد، زیرا گام های در جهت حداقل کردن تابع loss کوچک هستند. اگر نرخ یادگیری بالا باشد، ممکن است training همگرا یا حتی واگرا شود.

بنابراین باید نرخ یادگیری را یک عدد مناسب که نه زیاد کوچک باشد و نه زیاد بزرگ انتخاب کنیم. (در کتابخانه استفاده شده در حالت پیش فرض نرخ یادگیری را ۰.۰۱ درنظر گرفته است که به نسبت بزرگ است.)

ما آزمایش ها را در دو نرخ یادگیری مختلف ۰.۰۱ و ۰.۰۰۱ انجام دادیم.

همان طور که مشاهده می شود در این آزمایش ها به طور کلی سه معیار تعداد epoch، نرخ یادگیری و تعداد عصب های میانی هر چه بیشتر شدند درصد پیشبینی بهتر شده است.(البته در مواردی به این ترتیب نیست.) بهترین درصدی که به آن دست یافتیم ۸۳.۹ می باشد.

**مراجع**

[1] Vaish P. 4 different ways to predict survival on Titanic – part 3. a data analyst. Available at: URL: <http://adataanalyst.com/kaggle/4-different-ways-predict-survival-titanic-part-3/>. Accessed 30 Jan,2018.

[2] Gibiansky A. Machine Learning: Neural Networks. Andrew Gibiansky. Available at: URL: <http://andrew.gibiansky.com/blog/machine-learning/machine-learning-neural-networks/>. Accessed 30 Jan,2018.

[3]Khorami S. Titanic best working Classifier. Kaggle. Available at: URL:

<https://www.kaggle.com/sinakhorami/titanic-best-working-classifier/notebook>. Accessed 30 Jan,2018.

[4] Liu J. RailsConf 2017: Predicting Titanic Survivors with Machine Learning by Ju Liu. Youtube. Available at: URL: <https://www.youtube.com/watch?v=4l-aB_Sk41Y&t=1371s>. Accessed 30 Jan,2018.

[5] Bakhtiarova A. Titanic survivorship Neural Network model. Kaggle. Available at: URL:

<https://www.kaggle.com/ledadel/titanic-survivorship-neural-network-model?scriptVersionId=1555583>. Accessed 30 Jan,2018.

[6] Titanic by Neural Network with Keras. Kaggle. Available at: URL:

<https://www.kaggle.com/kay2mj/titanic-by-neural-network-with-keras>. Accessed 30 Jan,2018.

[7]Tripathi P. Prediction of Titanic Survival using R. Kaggle. Available at: URL:

<https://www.kaggle.com/pradeeptripathi/prediction-of-titanic-survival-using-r/code#L354>. Accessed 30 Jan,2018.

[8]NeuroLab 0.3.5 documentation. Python Software Foundation. Available at: URL: <https://pythonhosted.org/neurolab/lib.html>. Accessed 30 Jan,2018.

[9]ضیاشهابی ا. پیش‌بینی مسافران نجات‌یافته. بوته. .Available at: URL: <http://www.boute.ir/iust-ai-94/survived-passengers>

Accessed 30 Jan,2018.