

AKDENİZ UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING



Machine Learning – Course Project

Temmuz Burak Yavuzer - 20160808026

Problem Definition

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.

Problem for this project is pretty obvious. Main problem is use machine learning to create a model that predicts which passengers survived the Titanic shipwreck. You need to show the main ingredients of machine learning such as: Dataset Manipulation, Training a Classifier, Testing.

Based on machine learning method that is used, try to find accuracy, precision, recall, F-score of for the method that you have implemented.

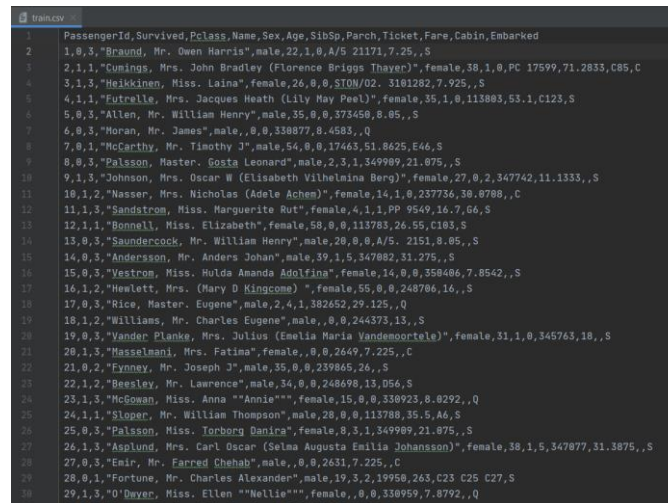
Compare the results that you have got after the executing the code.

You are free and encouraged to use the following machine learning and deep learning frameworks in your projects.

Data Set

In this dataset, we have two similar datasets that include passenger information like name, age, gender, socio-economic class, etc. One dataset is titled `train.csv` and the other is titled `test.csv`.

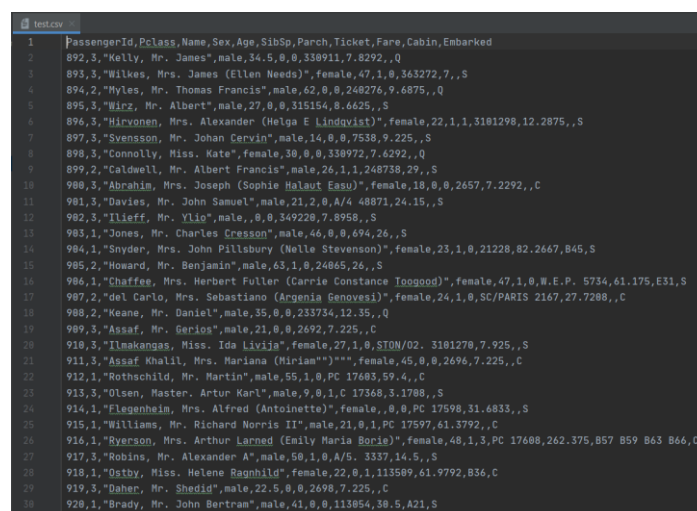
Train.csv will contain the details of a subset of the passengers on board (891 to be exact) and importantly, will reveal whether they survived or not, also known as the “ground truth”.



	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	1,0,3	"Braund, Mr. Owen Harris"	male	22,1,0,A/5	21171,7.25,,S							
2	2,1,1	"Cumings, Mrs. John Bradley (Florence Briggs Thayer)"	female	38,1,0,PC	17599,71.2833,C65,C							
3	3,1,3	"Heikkinen, Miss. Laina"	female	26,0,0,STON/O2	3101282,7.925,,S							
4	4,1,1	"Futrelle, Mrs. Jacques Heath (Lily May Peel)"	female	35,1,0,113803	53.1,C123,S							
5	5,0,3	"Allen, Mr. William Henry"	male	35,0,0,373450	8.05,,S							
6	6,0,3	"Moran, Mr. James"	male	0,0,358877	8.4583,,Q							
7	7,0,1	"McCarthy, Mr. Timothy J"	male	54,0,0,17463	51.8625,E46,S							
8	8,0,3	"Palsson, Master. Gosta Leonard"	male	2,3,1,349989	21.075,,S							
9	9,1,3	"Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)"	female	27,0,2,347742	11.1333,,S							
10	10,1,2	"Nasser, Mrs. Nicholas (Adele Achen)"	female	14,1,0,237736	36.8708,,C							
11	11,1,3	"Sandstrom, Miss. Marguerite Rut"	female	4,1,1,PP	9549,16.7,66,S							
12	12,1,1	"Bonnell, Miss. Elizabeth"	female	58,0,0,113783	26.55,C183,S							
13	13,0,3	"Saunderscock, Mr. William Henry"	male	20,0,0,A/5	2151,8.05,,S							
14	14,0,3	"Andersson, Mr. Anders Johan"	male	39,1,5,347882	31.275,,S							
15	15,0,3	"Vesterlund, Miss. Hulda Amanda Adolfina"	female	14,0,0,358466	7.8542,,S							
16	16,1,2	"Hewlett, Mrs. (Mary D Kingcome)"	female	55,0,0,248786	16,,S							
17	17,0,3	"Rice, Master. Eugene"	male	2,4,1,382852	29.125,,Q							
18	18,1,2	"Williams, Mr. Charles Eugene"	male	0,0,244373	13,,S							
19	19,0,3	"Vander Planke, Mrs. Julius (Emelia Maria Vanderplanke)"	female	31,1,0,345763	18,,S							
20	20,1,3	"Masselmani, Mrs. Fatima"	female	0,0,2649	7.225,,C							
21	21,0,2	"Finnay, Mr. Joseph J"	male	35,0,0,239865	26,,S							
22	22,1,2	"Bresley, Mr. Lawrence"	male	14,0,0,248698	11.056,S							
23	23,1,3	"McGowan, Miss. Anna "Annie""	female	15,0,0,338923	8.0292,,Q							
24	24,1,1	"Gloger, Mr. William Thompson"	male	20,0,0,113788	35.5,A6,S							
25	25,0,3	"Palsson, Miss. Torborg Danira"	female	8,3,1,349989	21.075,,S							
26	26,1,3	"Asplund, Mrs. Carl Oscar (Selma Augusta Emilia Johansson)"	female	38,1,5,347877	31.3875,,S							
27	27,0,3	"Emir, Mr. Farred Chehab"	male	0,0,2631	7.225,,C							
28	28,0,1	"Fortune, Mr. Charles Alexander"	male	19,3,2,19958	263,C25 C25 C27,S							
29	29,1,3	"O'Dwyer, Miss. Ellen "Nellie""	female	0,0,338959	7.8792,,Q							

The `test.csv` dataset contains similar information but does not disclose the “ground truth” for each passenger.

Using the patterns we find in the train.csv data, we have to predict whether the other 418 passengers on board (found in test.csv) survived.



	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	892,3	"Kelly, Mr. James"	male	34.5,0,0,338911	7.8292,,Q						
2	893,3	"Wilkes, Mrs. James (Ellen Needs)"	female	47,1,0,363272	7,,S						
3	894,2	"Hyles, Mr. Thomas Francis"	male	62,0,0,248276	9.6875,,Q						
4	895,3	"Wicz, Mr. Albert"	male	27,0,0,315154	8.6625,,S						
5	896,3	"Hirvonen, Mrs. Alexander (Helga E Lindqvist)"	female	22,1,1,3181298	12.2875,,S						
6	897,3	"Svensson, Mr. Johan Cervin"	male	14,0,0,7538	9.225,,S						
7	898,3	"Connolly, Miss. Kate"	female	30,0,0,338972	7.6292,,Q						
8	899,2	"Caldwell, Mr. Albert Francis"	male	26,1,1,248738	29,,S						
9	900,3	"Abraham, Mrs. Joseph (Sophie Halaut Easu)"	female	18,0,0,2657	7.2292,,C						
10	901,3	"Davies, Mr. John Samuel"	male	21,2,0,A/4	48871,24.15,,S						
11	902,3	"Ilieff, Mr. Ylio"	male	0,0,349228	7.8958,,S						
12	903,1	"Jones, Mr. Charles Cresson"	male	46,0,0,694	26,,S						
13	904,1	"Snyder, Mrs. John Pillsbury (Nelle Stevenson)"	female	23,1,0,21228	82.2667,845,S						
14	905,2	"Howard, Mr. Benjamin"	male	63,1,0,24865	26,,S						
15	906,1	"Chaffee, Mrs. Herbert Fuller (Carrie Constance Toogood)"	female	47,1,0,W.E.P.	5734,61.175,E31,S						
16	907,2	"del Carlo, Mrs. Sebastiano (Argenia Genovesi)"	female	24,1,0,SC/PARIS	2167,27.7288,,C						
17	908,2	"Keane, Mr. Daniel"	male	35,0,0,233734	12.35,,Q						
18	909,3	"Assaf, Mr. Gerios"	male	21,0,0,2692	7.225,,C						
19	910,3	"Ilmakangas, Miss. Ida Livija"	female	27,1,0,STON/O2	3181270,7.925,,S						
20	911,3	"Assaf Khalil, Mrs. Mariana (Miriam)"	female	45,0,0,2696	7.225,,C						
21	912,1	"Rothschild, Mr. Martin"	male	55,1,0,PC	17683,59.4,,C						
22	913,3	"Olsen, Master. Artur Karl"	male	9,0,1,C	17368,3.1788,,S						
23	914,1	"Elegenheim, Mrs. Alfred (Antoinette)"	female	0,0,PC	17598,31.6833,,S						
24	915,1	"Williams, Mr. Richard Norris II"	male	21,0,1,PC	17597,61.3792,,C						
25	916,1	"Ryerson, Mrs. Arthur Larned (Emily Maria Borie)"	female	48,1,3,PC	17688,262.375,857 859 863 866,C						
26	917,3	"Robins, Mr. Alexander A"	male	50,1,0,A/5	3337,14.5,,S						
27	918,1	"Ostby, Miss. Helene Ragnhild"	female	22,0,1,113589	61.9792,836,C						
28	919,3	"Daher, Mr. Shedid"	male	22,5,0,0,2698	7.225,,C						
29	920,1	"Brady, Mr. John Bertram"	male	41,0,0,113854	38.5,A21,S						

Ps. You can download this data set from [Kaggle.com](https://www.kaggle.com/datasets/u2csy2/titanic)

Methodology

K-nearest neighbors (kNN) is a supervised machine learning algorithm that can be used to solve both classification and regression tasks. KNN as an algorithm that comes from real life. The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other (sometimes called distance, proximity, or closeness).

Advantages of KNN

- Quick calculation time
- Simple algorithm – to interpret
- Versatile – useful for regression and classification
- High accuracy – you do not need to compare with better-supervised learning models
- No assumptions about data – no need to make additional assumptions, tune several parameters, or build a model. This makes it crucial in nonlinear data case.

Disadvantages of KNN

- Accuracy depends on the quality of the data
- With large data, the prediction stage might be slow
- Sensitive to the scale of the data and irrelevant features
- Require high memory – need to store all of the training data

For the algorithm to work best on a particular dataset we need to choose the most appropriate distance metric accordingly. There are a lot of different distance metrics available, but I am only going to talk about a few widely used ones.

So here are some of the distances used:

Chebyshev Distance, Minkowski Distance, Manhattan Distance, Euclidean Distance

Chebyshev Distance

The Chebyshev distance between two points is defined as the maximum value of the absolute value of the coordinate value difference. From the mathematical point of view, Chebyshev distance is a measure derived from uniform norm, and also a kind of hyperconvex measure.

The formula for calculating Chebyshev Distance is:

$$d(x, y) = \max_i |x_i - y_i|$$

Minkowski Distance

It is a metric intended for real-valued vector spaces. We can calculate Minkowski distance only in a normed vector space, which means in a space where distances can be represented as a vector that has a length and the lengths cannot be negative.

There are a few conditions that the distance metric must satisfy:

1. *Non-negativity*: $d(x, y) \geq 0$
2. *Identity*: $d(x, y) = 0$ if and only if $x = y$
3. *Symmetry*: $d(x, y) = d(y, x)$
4. *Triangle Inequality*: $d(x, y) + d(y, z) \geq d(x, z)$

$$\left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}$$

Manhattan Distance

This distance is also known as taxicab distance or city block distance, that is because the way this distance is calculated. The distance between two points is the sum of the absolute differences of their Cartesian coordinates.

As we know we get the formula for Manhattan distance by substituting $p=1$ in the Minkowski distance formula.

$$d = \sum_{i=1}^n |x_i - y_i|$$

Euclidean Distance

This distance is the most widely used one as it is the default metric that SKlearn library of Python uses for K-Nearest Neighbour. It is a measure of the true straight line distance between two points in Euclidean space.

It can be used by setting the value of p equal to 2 in Minkowski distance metric.

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

The KNN Algorithm Steps

1. Load the data
2. Initialize K to your chosen number of neighbors
3. For each example in the data
 - 3.1 Calculate the distance between the query example and the current example from the data. Add the distance and the index of the example to an ordered collection
4. Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances
5. Pick the first K entries from the sorted collection
6. Get the labels of the selected K entries
7. If regression, return the mean of the K labels, If classification, return the mode of the K labels

Implementation

```
miniprojecttemmuz.py x
1 import pandas as pd
2 from sklearn.preprocessing import StandardScaler
3 from sklearn.neighbors import KNeighborsClassifier
4 from sklearn.metrics import confusion_matrix
5 from sklearn.decomposition import PCA
```

Before starting the main coding part, we have to understand which libraries we should import for our project. Pandas library is basically used for manipulation of the titanic dataset. Scikit-learn is a machine learning library in Python. It features various algorithms like support vector machine, random forests, and k-neighbours which we will use. Especially for KNN's classifier and confusion matrix to find the accuracy, precision, recall and fscore. My main idea to use PCA which is short version of Principal Component Analysis, used for to speed it up.

```
titanic_train = pd.read_csv("train.csv")
titanic_test = pd.read_csv("test.csv")
test_given_predict = pd.read_csv("gender_submission.csv")
```

This section is for importing the datasets that we have got from the Kaggle.com. All of those three files is provided via Kaggle.

```
survived_people = titanic_train["Survived"].values
titanic_train = titanic_train.drop(["Survived"], axis=1)
all_datas = pd.concat([titanic_train, titanic_test], ignore_index=True, axis=0)
all_datas = all_datas.drop(["PassengerId", "Name", "Ticket", "Fare", "Cabin"], axis=1)
all_datas["Embarked"].fillna(method='ffill', inplace=True)
all_datas["Age"].fillna(all_datas["Age"].median(), inplace=True)
```

Getting the survived people from the dataset. Taking of the survived people then add to the set together. Take off the useless parts then fill the empty values for valid and better results.

```
gender_temp = pd.get_dummies(all_datas['Sex'], prefix='Sex')
gender_temp = gender_temp.iloc[:, 0]
aboard_temp = pd.get_dummies(all_datas['Embarked'], prefix='Embarked')
aboard_temp = aboard_temp.iloc[:, :2]
```

Encoding the data into temporary variable. Second line is used for avoiding the temporary variable multicollinearity. Ps. Embarked means that If the passenger is in the ship or miss the voyage.

```
all_datos = all_datos.drop(["Sex", "Embarked"], axis=1)
all_datos = pd.concat([gender_temp, aboard_temp, all_datos], axis=1)
indept_var = all_datos.iloc[:].values
```

Taking of the column from the dataset then, add the new ones.

```
indept_var_train = indept_var[:891, :]
indept_var_test = indept_var[891:, :]
survived_people_train = survived_people[:]
survived_people_test = test_given_predict.iloc[:, 1].values
```

Dividing the data set into for training the model and for the testing the model for our project and letting survived_people_test to the end. I have manually set to 891 because of our datasets property, you could choose different value.

```
scaling = StandardScaler()
indept_var_train = scaling.fit_transform(indept_var_train)
indept_var_test = scaling.transform(indept_var_test)
```

Scaling the characteristics, features for that it will transform your data such that its distribution will have a mean value 0 and standard deviation of 1.

```
classifier_chebyshev = KNeighborsClassifier(n_neighbors=13, metric='chebyshev', p=7, n_jobs=-1)
classifier_chebyshev.fit(indept_var_train, survived_people_train)
survived_people_predict = classifier_chebyshev.predict(indept_var_test)
matrix = confusion_matrix(survived_people_test, survived_people_predict)
```

Fitting the classifier to the training set for Chebyshev distance that we have talked out at the methodology chapter. After that we are trying to predict the test dataset results. And also we have to create confusion matrix.

```
False_Positive = matrix[1][0]
False_Negative = matrix[0][1]
True_Positive = matrix[0][0]
True_Negative = matrix[1][1]
```

Required values for calculations.

```
accuracy = ((True_Positive + True_Negative) / (True_Positive + True_Negative + False_Positive + False_Negative))
precision = True_Positive / (True_Positive + False_Positive)
recall = True_Positive / (True_Positive + False_Negative)
f_score = (2 * precision * recall) / (precision + recall)
```


Using the FalsePositive, FalseNegative, TruePositive, TrueNegative values, we are calculating the accuracy, precision, recall, and fscore.

```
print("Chebyshev Results")
print("Accuracy is {:.6f}.".format(accuracy))
print("Precision is {:.6f}.".format(precision))
print("Recall is {:.6f}.".format(recall))
print("Fscore is {:.6f}.\n".format(f_score))
```

Printing the values of the calculations.

```
isSurvived = {'Survived':survived_people_predict}
idPeople = {'PassengerId': titanic_test.iloc[:, 0].values}
alldata_predicts = pd.concat([pd.DataFrame(idPeople), pd.DataFrame(isSurvived)], axis=1)
alldata_predicts.to_csv('chebyshev_predictions.csv', encoding='utf-8', index=False)
```

Producing a new dataset to see the result for each person for each distance.

```
pca = PCA(n_components=4)
indept_var_train = pca.fit_transform(indept_var_train)
indept_var_test = pca.transform(indept_var_test)
variance = pca.explained_variance_ratio_
```

My main idea to use PCA which is short version of Principal Common Analysis, used for to speed it up.

```
classifier_minkowski = KNeighborsClassifier(n_neighbors=13, metric='minkowski', p=3, n_jobs=-1)
classifier_minkowski.fit(indept_var_train, survived_people_train)
survived_people_predict = classifier_minkowski.predict(indept_var_test)
matrix = confusion_matrix(survived_people_test, survived_people_predict)
```

```
classifier_manhattan = KNeighborsClassifier(n_neighbors=13, metric='manhattan', p=1, n_jobs=-1)
classifier_manhattan.fit(indept_var_train, survived_people_train)
survived_people_predict = classifier_manhattan.predict(indept_var_test)
matrix = confusion_matrix(survived_people_test, survived_people_predict)
```

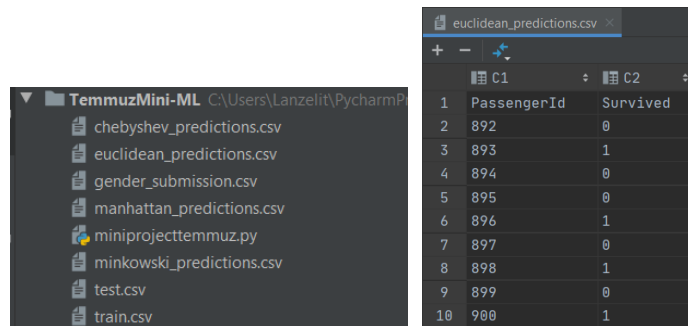
```
classifier_euclidean = KNeighborsClassifier(n_neighbors=13, metric='euclidean', p=2, n_jobs=-1)
classifier_euclidean.fit(indept_var_train, survived_people_train)
survived_people_predict = classifier_euclidean.predict(indept_var_test)
matrix = confusion_matrix(survived_people_test, survived_people_predict)
```

Same structure used for those experiments too.

Experiments and Results

```
Chebyshev Results      Minkowski Results
Accuracy is 0.899522.   Accuracy is 0.913876
Precision is 0.927481.  Precision is 0.929104
Recall is 0.913534.    Recall is 0.936090
Fscore is 0.920455.    Fscore is 0.932584

Manhattan Results      Euclidean Results
Accuracy is 0.899522    Accuracy is 0.918660
Precision is 0.911765   Precision is 0.942748
Recall is 0.932331     Recall is 0.928571
Fscore is 0.921933     Fscore is 0.935606
```



The image shows a file explorer on the left with a folder named 'TemmuzMini-ML' containing several CSV files: 'chebyshev_predictions.csv', 'euclidean_predictions.csv', 'gender_submission.csv', 'manhattan_predictions.csv', 'minkowski_predictions.csv', 'test.csv', and 'train.csv'. To the right, a preview of the 'euclidean_predictions.csv' file is shown, displaying two columns: 'PassengerId' and 'Survived'.

	C1	C2
1	PassengerId	Survived
2	892	0
3	893	1
4	894	0
5	895	0
6	896	1
7	897	0
8	898	1
9	899	0
10	900	1

In this experiment main problem was use machine learning to create a model that predicts which passengers survived the Titanic shipwreck. When we look at the euclidean_predictions file we can easily see the predicted result and and compare with the main dataset(each distance methods has prediction file too).At the top of this sections you are able to see each distance method's accuracy, precision,recall and fscores. When we look at the accuracies best result is at the Euclidean method because of the we can easily understand why the Euclidean method is the most used one.On the other hand Chebyshec's and Manhattan's accuracies are same but other values aren't. The biggest precision values is belong to Euclidean because, precision means that how precise/accurate our model is out of those predicted positives , how many of them actually are positive.If we look at the recall values,Chebyshev has the highest value because recall actually calculates how many of the actual positives our model capture through labelling it as positive.When we look at the Fscores, Euclidean has the highest one.This is because of the Fscore means that when you want to seek a balance between precision and recall.

Full Project Code can be downloaded via this link

<https://drive.google.com/drive/folders/1jfTPRXmheXW6pPESJE5vBTFe5GeO2JaS?usp=sharing>

Source

<https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>

http://bcmi.sjtu.edu.cn/home/niuli/teaching/2020_2_3.pdf

<https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9>

<https://medium.com/@arslanov/makine-%C3%B6%C4%9Frenmesi-knn-k-nearest-neighbors-algoritmas%C4%B1-bdfb688d7c5f>