Akshay Sadalgekar – 1001944713

Aaditya Damle – 1001955625

Aishwarya Muchandi – 1001879673

Report NN

1. Introduction:

Titanic dataset contains real data from the titanic accident On April 15, 1912, the largest passenger liner ever made collided with an iceberg during her maiden voyage. When the Titanic sank it killed 1502 out of 2224 passengers and crew.

This dataset contains 891 records of the passengers and the dataset columns are Name, Pclass, Age, Sex, Sibsp, Parch, Ticket, Fare, cabin, Embarked.

Nearest Neighbors Classification is used and scikit learn implement KNeighborsClassifier which implements learning based on K nearest neighbors at each point

This dataset is analysed using python pandas matplotlib, seaborn.

Python:

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects

NumPy:

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

Pandas:

Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

Matplotlib:

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

Seaborn:

Seaborn is a library for making statistical graphics in Python. It builds on top of [matplotlib](https://matplotlib.org/) and integrates closely with [pandas](https://pandas.pydata.org/) data structures.

Seaborn helps you explore and understand your data. Its plotting functions operate on data frames and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots.

Scikit-learn:

Scikit-learn is an open source machine learning library that supports supervised and unsupervised learning. It also provides various tools for model fitting, data pre-processing, model selection and evaluation, and many other utilities.

the main features that scikit-learn provides. It assumes a very basic working knowledge of machine learning practices (model fitting, predicting, cross-validation, etc.)

1. Nearest Neighbor Method:

Nearest Neighbor is includes functionality for supervised and unsupervised neighbors-based learning.

The nearest neighbor method is used to find number of training samples closest to the new point and predict the label from these points.

These number of sample can be user-defined constant which is called k-nearest learning or based on local density of point which is known as radius based learning.

The distance between points can be any distance measuring metric and among those Euclidean is most commonly used.

Syntax:

class sklearn.neighbors.NearestNeighbors(\*, n\_neighbors=5, radius=1.0, algorithm='auto', leaf\_size=30, metric='minkowski', p=2, metric\_params=None, n\_jobs=None)[[source]](https://github.com/scikit-learn/scikit-learn/blob/844b4be24/sklearn/neighbors/_unsupervised.py#L7)

Parameters:

**n\_neighbors:** int, default=5

* Number of neighbors to use by default for [kneighbors](https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.NearestNeighbors.html" \l "sklearn.neighbors.NearestNeighbors.kneighbors" \o "sklearn.neighbors.NearestNeighbors.kneighbors)

**Radius** float, default=1.0

* Range of parameter space to use by default for [radius\_neighbors](https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.NearestNeighbors.html" \l "sklearn.neighbors.NearestNeighbors.radius_neighbors" \o "sklearn.neighbors.NearestNeighbors.radius_neighbors)

**algorithm**{‘auto’, ‘ball\_tree’, ‘kd\_tree’, ‘brute’}, default=’auto’

* Algorithm used to compute the nearest neighbors

**leaf\_size:** int, default=30

**metric** str or callable, default=’minkowski’

* The distance metric to use for the tree. The default metric is minkowski, and with p=2 is equivalent to the standard Euclidean metric

**p:** int, default=2

**n\_job: s**int, default=None

* The number of parallel jobs to run for neighbors search

Methods:

fit(X[, y])

* Fit the nearest neighbors estimator from the training dataset.

get\_params([deep])

* Get parameters for this estimator.

kneighbors([X, n\_neighbors, return\_distance])

* Find the K-neighbors of a point.

kneighbors\_graph([X, n\_neighbors, mode])

* Compute the (weighted) graph of k-Neighbors for points in X.

radius\_neighbors([X, radius, …])

* Find the neighbors within a given radius of a point or points.

radius\_neighbors\_graph([X, radius, mode, …])

* Compute the (weighted) graph of Neighbors for points in X.

set\_params(\*\*params)

* Set the parameters of this estimator.

1. Criteria for selecting the three attributes:

There are 7 columns in the dataset ‘ Survived', 'Pclass', 'Age', 'SibSp', 'Parch',

'Embarked', 'Sex'.

We are choosing y label as ‘Survived’ column as it tells how many passenger survived in the titanic accident. Out of remaining 6 columns we decided to choose

1 Feature: Pclass

2 Feature: Embarked

3 Feature: Sex

The criteria to choose these columns was, in order to determine how many survived in

the crash these best 3 features plays significant role in contributing the target variable .

We used sklearn.feature\_selection.mutual\_info\_classif which estimatedsmutual

information for target. It is calculates entropy for each features and feature with higher

values indicate high dependency on the target “Survived”.

This features will give best results for the classification model.

Also in real life scenario Sex will determine the gender of the passenger which will be

useful for keeping count, Embarked will be useful for local authorities to trace their family members and Pclass can give additional information how high-class passengers used their powers to save each other compare to poor.

The other 3 attributes we can choose is Age, Parch, Sibsp which come in top ten list of

Features who has high dependency on target and would give better results when used.

Text

Description automatically generated

Chart, bar chart

Description automatically generated

1. Interpreting and Comparing result:

4.1 With k = 3

Table

Description automatically generated

Text

Description automatically generated

Graphical user interface

Description automatically generated with low confidence

In the confusion matrix with k = 3, more actually negative values are correctly predicted as negative which shows the good accuracy.

We have reduce the values in diagonal from bottom left to top right in the confusion matrix in order to get a good result of the classifier.

* 1. With k = 5

Table

Description automatically generated

Text

Description automatically generated

A picture containing graphical user interface

Description automatically generated

In the classification report with k = 5, the values corresponding to precision, f1 score and recall have increased in comparison with the classifier with k = 3.

Here in the confusion matrix, more values both negative and positive are predicted correctly.

* 1. With k=15

Table

Description automatically generated

Text

Description automatically generated

A picture containing graphical user interface

Description automatically generated

In this case the confusion matrix displays maximum values with correct prediction as compared to the previous two outputs.

The values corresponding to precision, f1 score and recall are highest in this classification report with k = 15, as opposed to the classifiers with k = 3 and k=5.

External Sources:

<https://pandas.pydata.org/docs/user_guide/10min.html#grouping>

<https://numpy.org/doc/stable/>

<https://matplotlib.org/stable/tutorials/index.html>

<https://matplotlib.org/stable/gallery/lines_bars_and_markers/barchart.html>

<https://seaborn.pydata.org/generated/seaborn.catplot.html>

<http://seaborn.pydata.org/generated/seaborn.countplot.html#seaborn.countplot>

<https://datascience.stackexchange.com/questions/77534/how-to-use-df-groupby-to-select-and-sum-specific-columns-w-o-pandas-trimming-t>

<https://stackoverflow.com/questions/61667274/plotting-multiple-countplots-using-seaborn>

<https://www.datasciencemadesimple.com/pie-chart-in-python-with-legends/>

<https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>

<https://www.youtube.com/watch?v=wTF6vzS9fy4&t=757s>

<https://www.youtube.com/watch?v=81JSbXZ26Ls>

<https://towardsdatascience.com/predicting-the-survival-of-titanic-passengers-30870ccc7e8>

https://medium.com/@moussadoumbia\_90919/elbow-method-in-supervised-learning-optimal-k-value-99d425f229e7