

TITANIC DATASET

EXAMPLE DECK

March 27, 2023

DAMINI TREHAN CONSULTING LLC

NEAREST NEIGHBOR
(KEIGHBORS CLASSIFIER)

SUPPORT VECTOR
MACHINE

DECISION TREE

LOGISTIC REGRESSION

RANDOM FOREST

- 1.OBJECTIVE / PROBLEM STATEMENT
- 2.LITERATURE REVIEW / MARKET RESEARCH /
METHODS RESEARCH
- 3.DATA SET / DATA
- 4.METHODS / PROPOSED METHODS
- 5.PROJECT TIMELINE
- 6.FINDINGS

TABLE OF CONTENTS

OBJECTIVE :
{INSERT TEXT HERE}

{Insert Text here}

PROBLEM STATEMENT

LITERATURE REVIEW

train_data

✓ 0.1s

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	
0	1	0	3	Braund, Mr. Owen Harris	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	6.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	2.0	0	0	370376	17.7500	NaN	Q

891 rows × 12 columns

DATA

test_data

✓ 0.0s

PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S
...
413	1305	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	S
414	1306	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C
415	1307	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	S
416	1308	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	S
417	1309	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	C

418 rows x 11 columns

A. Random Forest

```
from sklearn.ensemble import RandomForestClassifier
y = train_data["Survived"]

features = ["Pclass", "Sex", "SibSp", "Parch"]
X = pd.get_dummies(train_data[features])
X_test = pd.get_dummies(test_data[features])

model = RandomForestClassifier(n_estimators=100, max_depth=5, random_state=1)
model.fit(X, y)
predictions = model.predict(X_test)

output = pd.DataFrame({'PassengerId': test_data.PassengerId, 'Survived': predictions})
output
```

D. Decision Tree

```
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier(max_depth=3, random_state=42)
clf.fit(X_train, y_train)
clf.predict(X_test)
predictions = clf.predict(X_final_test)

output = pd.DataFrame({'PassengerId': test_data.PassengerId, 'Survived': predictions})
output
```

B. K Neighbors

```
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier

y = train_data["Survived"]

features = ["Pclass", "Sex", "SibSp", "Parch"]
X = pd.get_dummies(train_data[features])
X_final_test = pd.get_dummies(test_data[features])

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=42, stratify=y)

k = 7 #arbitrarily choose a k for now
knn = KNeighborsClassifier(n_neighbors=k)
knn.fit(X_train, y_train)
predictions = knn.predict(X_final_test)

output = pd.DataFrame({'PassengerId': test_data.PassengerId, 'Survived': predictions})
output
```

C. Logistic Regression

```
from sklearn.linear_model import LogisticRegression
from sklearn import metrics

logisticRegr = LogisticRegression(solver='lbfgs', max_iter=1000)
logisticRegr.fit(X_train, y_train)
logisticRegr.predict(X_test)
predictions = logisticRegr.predict(X_final_test)

output = pd.DataFrame({'PassengerId': test_data.PassengerId, 'Survived': predictions})
```

E. SVM

```
from sklearn.svm import SVC

svm = SVC()
svm.fit(X_train, y_train)
svm.predict(X_test)
predictions = svm.predict(X_final_test)

output = pd.DataFrame({'PassengerId': test_data.PassengerId, 'Survived': predictions})
output
```

PROPOSED METHODS

PROJECT TIMELINE

FINDINGS

DAMINI TREHAN CONSULTING LLC