```
import numpy as np
import pandas as pd
```

```
master_titanic = pd.read_csv("/content/Titanic.csv")
```

```
master_titanic.head()
```

```
df = master_titanic.iloc[:, [2,4,5,6,7,9]]
```

```
df
```

```
x = df.join(pd.get_dummies(df.Sex))
```

```
del x['male']
del x['Sex']
```

```
x['Survived'] =master_titanic['Survived']
```

```
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn.model_selection import train_test_split # Import train_test_split funct
from sklearn import metrics
import matplotlib.pyplot as plt
```

```
x['Age'].replace(to_replace=np.nan, value=x.Age.mean(), inplace=True, limit=None, regex=False, metho
```

```
x.isna().value_counts()

     Pclass  Age    SibSp  Parch  Fare   female  Survived
     False   False  False  False  False  False   False       891
     dtype: int64
```

```
X_train, X_test, y_train, y_test = train_test_split(x.loc[:,x.columns != 'Survived'], x['Survived'],
```

### Build a decision tree and Make a prediction with a decision tree.

```
dt_classifier = DecisionTreeClassifier().fit(X_train, y_train)
```

```
y_pred = dt_classifier.predict(X_test)
```

```
y=y_test
```

```
X=y_pred
```

```
**Estimate the accuracy scores to best analyse the predictions
for each case.**
```

### Estimate the accuracy scores to best analyse the predictions for each case.

```
from sklearn import metrics
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

     Accuracy: 0.7541899441340782
```

### Evaluate using gridsearch CV

```
from sklearn.model_selection import GridSearchCV

gd= GridSearchCV(dt_classifier,{'max_depth':[x for x in range (10)],'criterion':['gini','entropy',]}
```

**Entropy is the measurement of the impurity or randomness in the data points.**

**Gini index calculates the amount of probability of a specific feature that is classified incorrectly when selected randomly**

```
gd.fit(X_train,y_train)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/model_selection/_validation.py:372: FitFailedWar
20 fits failed out of a total of 200.
The score on these train-test partitions for these parameters will be set to nan.
If these failures are not expected, you can try to debug them by setting error_score='raise'.

Below are more details about the failures:
--------------------------------------------------------------------------------
20 fits failed with the following error:
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/sklearn/model_selection/_validation.py", line 68
    estimator.fit(X_train, y_train, **fit_params)
  File "/usr/local/lib/python3.7/dist-packages/sklearn/tree/_classes.py", line 942, in fit
    X_idx_sorted=X_idx_sorted,
  File "/usr/local/lib/python3.7/dist-packages/sklearn/tree/_classes.py", line 306, in fit
    raise ValueError("max_depth must be greater than zero. ")
ValueError: max_depth must be greater than zero.

  warnings.warn(some_fits_failed_message, FitFailedWarning)
/usr/local/lib/python3.7/dist-packages/sklearn/model_selection/_search.py:972: UserWarning: One
 0.79489437 0.80199531 0.79493349 0.79219484        nan 0.7879108
 0.77668232 0.82445227 0.81883803 0.80479264 0.79217527 0.79211659
 0.79493349 0.79350548]
  category=UserWarning,
GridSearchCV(cv=10, estimator=DecisionTreeClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]})
```

```
print(gd.best_params_)
print(gd.best_score_)
```

```
{'criterion': 'entropy', 'max_depth': 3}
0.8244522691705791
```

## Calculating Bias and variance.

```
# %pip install mlxtend --upgrade
```

```
from mlxtend.evaluate import bias_variance_decomp
mse, bias, var = bias_variance_decomp(dt_classifier, X_train.values, y_train.values,
                                      X_test.values, y_test.values,
                                      loss='mse', random_seed=1)# summarize results
print('MSE: %.3f' % mse)
print('Bias: %.3f' % bias)
print('Variance: %.3f' % var)
```

```
MSE: 0.243
Bias: 0.150
Variance: 0.093
```

## Justify the bias and variance

Bias is value which tells us how good the model has fit to the training set, that is the amount of assumptions a model has taken to better fit the validation data.

Variance is the amount by which model underperforms on test dataset after training on test dataset. This results majorly due to overfitting.