# Mental Health Prediction

DATA-DRIVEN PREDICTION OF MENTAL HEALTH STATUS USING AI MODELS

# Logistic Regression

Home | Mental | Home | Mental | Home | Mental | Home | Mental | Home | Mental
localhost:8888/notebooks/Mental%20Health%20Prediction%20by%20Logistic%20Regression.ipynb

jupyter Mental Health Prediction by Logistic Regression Last Checkpoint: 11 days ago

File Edit View Run Kernel Settings Help Trusted

Markdown Open in... Python 3 (ipykernel)

# Importing Important Libraries

[1]:
```
import pandas as pd
import numpy as np
```

# Importing the dataset from a CSV file into Jupyter Notebook

[2]:
```
df = pd.read_csv('survey.csv')
df.head()
```

[2]:

| | Timestamp | Age | Gender | Country | state | self_employed | family_history | treatment | work_interfere | no_employees | ... | leave | men |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 27-08-2014 11:29 | 37 | Female | United States | IL | NaN | No | Yes | Often | Jun-25 | ... | Somewhat easy | |
| **1** | 27-08-2014 | 44 | M | United | IN | NaN | No | No | Rarely | More than | | Don't | |

Search ENG IN 02:22 PM 25-10-2025

Home | Mental | Home | Mental | Home | Mental | Home | Mental | Home | Mental
localhost:8888/notebooks/Mental%20Health%20Prediction%20by%20Logistic%20Regression.ipynb
jupyter Mental Health Prediction by Logistic Regression Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Dropping unnecessary columns from the dataset

[3]:
```
df = df.drop(['Timestamp', 'comments'], axis=1)

df.head()
```

[3]:

| | Age | Gender | Country | state | self_employed | family_history | treatment | work_interfere | no_employees | remote_work | ... | anonymity | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 37 | Female | United States | IL | NaN | No | Yes | Often | Jun-25 | No | ... | Yes | S |
| **1** | 44 | M | United States | IN | NaN | No | No | Rarely | More than 1000 | No | ... | Don't know | |
| **2** | 32 | Male | Canada | NaN | NaN | No | No | Rarely | Jun-25 | No | ... | Don't know | S |
| **3** | 31 | Male | United Kingdom | NaN | NaN | Yes | Yes | Often | 26-100 | No | ... | No | S |

Search ENG IN 02:23 PM 25-10-2025

Home | Mental | Home | Mental | Home | Mental | Home | Mental | Home | Mental
localhost:8888/notebooks/Mental%20Health%20Prediction%20by%20Logistic%20Regression.ipynb

Mental Health Prediction by Logistic Regression Last Checkpoint: 11 days ago

File Edit View Run Kernel Settings Help Trusted

Markdown Open in... Python 3 (ipykernel)

# Checking dataset columns for required data preprocessing

[4]:
```
df['no_employees'].unique()
```

[4]:
```
array(['Jun-25', 'More than 1000', '26-100', '100-500', '01-May',
       '500-1000'], dtype=object)
```

[5]:
```
# Keep only valid employee range values
valid_values = ['1-5', '6-25', '26-100', '100-500', '500-1000', 'More than 1000']

# Replace invalid ones with NaN
df['no_employees'] = df['no_employees'].apply(lambda x: x if x in valid_values else None)

# Then, you can fill missing with mode (most common value)
df['no_employees'].fillna(df['no_employees'].mode()[0], inplace=True)
```

[6]:
```
df['no_employees'].unique()
```

Search ENG IN 02:23 PM 25-10-2025

Home | Mental | Home | Mental | Home | Mental | Home | Mental | Home | Mental
localhost:8888/notebooks/Mental%20Health%20Prediction%20by%20Logistic%20Regression.ipynb



# Applying Label Encoding to convert binary categorical data into numeric form

[10]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
for column in df.columns:
    if df[column].dtype == 'object' and df[column].nunique() == 2:
        df[column] = le.fit_transform(df[column])
    else:
        pass
```

[11]:
```
df.head()
```

[11]:

| | Age | Gender | Country | state | self_employed | family_history | treatment | work_interfere | no_employees | remote_work | ... | anonymity | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 37 | Female | United States | IL | 2 | 0 | 1 | Often | 26-100 | 0 | ... | Yes | S |
| | | | United | | | | | | More than | | | Don't | |

Search ENG IN 02:23 PM 25-10-2025

Home | Mental | Home | Mental | Home | Mental | Home | Mental | Home | Mental
localhost:8888/notebooks/Mental%20Health%20Prediction%20by%20Logistic%20Regression.ipynb
jupyter Mental Health Prediction by Logistic Regression Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Saving the modified dataset to a CSV file for future use

[12]:
```
df.to_csv('survey_modifies.csv', index = False)
```

# Applying One Hot Encoding to convert multi-categorical data into numeric form

[13]:
```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder

# Creat a list of Multi-category categorical Columns
mult_cat_col = [col for col in df.columns if df[col].dtype == 'object' and df[col].nunique() > 2]

# Creat a list of Binary-category categorical Columns
Bin_cat_col_names = [col for col in df.columns if col not in mult_cat_col]
```

Search ENG IN 02:23 PM 25-10-2025

Home | Mental | Home | Mental | Home | Mental | Home | Mental | Home | Mental
localhost:8888/notebooks/Mental%20Health%20Prediction%20by%20Logistic%20Regression.ipynb



# Applying SimpleImputer to fill missing values in the dataset

[15]:
```
from sklearn.impute import SimpleImputer
si = SimpleImputer(missing_values=np.nan, strategy='mean')
si.fit(df)
```

[15]:
```
▾ SimpleImputer
SimpleImputer()
```

# Dividing the dataset into features (X) and target (y)

[16]:
```
X = df.drop('treatment', axis=1)
y = df['treatment']
```

Search ENG IN 02:23 PM 25-10-2025

Home | Mental | Home | Mental | Home | Mental | Home | Mental | Home | Mental
localhost:8888/notebooks/Mental%20Health%20Prediction%20by%20Logistic%20Regression.ipynb
jupyter Mental Health Prediction by Logistic Regression Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Dividing the dataset into features (X) and target (y)

[16]:
```
X = df.drop('treatment', axis=1)
y = df['treatment']
```

# Splitting the dataset into training and testing sets: X_train, X_test, y_train, y_test

[17]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.25, random_state=1)
```

# Applying feature scaling to normalize the dataset

Search ENG IN 02:23 PM 25-10-2025

Home | Mental | Home | Mental | Home | Mental | Home | Mental | Home | Mental
localhost:8888/notebooks/Mental%20Health%20Prediction%20by%20Logistic%20Regression.ipynb
jupyter Mental Health Prediction by Logistic Regression Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Applying Logistic Regression algorithm to the dataset

[19]:
```
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state=0)
classifier.fit(X_train, y_train)
```

[19]:
```
LogisticRegression
LogisticRegression(random_state=0)
```

# Predicting target values for X_test and comparing with actual values before applying K-Fold cross-validation

[20]:
```
y_pred = classifier.predict(X_test)
comparison = pd.DataFrame({'predictied ': y_pred, 'actual ': y_test})
```

Search ENG IN 02:23 PM 25-10-2025

Home | Mental | Home | Mental | Home | Mental | Home | Mental | Home | Mental
localhost:8888/notebooks/Mental%20Health%20Prediction%20by%20Logistic%20Regression.ipynb
jupyter Mental Health Prediction by Logistic Regression Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Predicting target values for X_test and comparing with actual values before applying K-Fold cross-validation

[20]:
```
y_pred = classifier.predict(X_test)
comparison = pd.DataFrame({'predictied ': y_pred, 'actual ': y_test})
comparison.head(10)
```

[20]:

| | predictied | actual |
|---|---|---|
| **972** | 1.0 | 1.0 |
| **719** | 0.0 | 0.0 |
| **979** | 1.0 | 1.0 |
| **550** | 0.0 | 0.0 |
| **512** | 0.0 | 0.0 |

Search ENG IN 02:23 PM 25-10-2025

Home | Mental | Home | Mental | Home | Mental | Home | Mental | Home | Mental
localhost:8888/notebooks/Mental%20Health%20Prediction%20by%20Logistic%20Regression.ipynb
jupyter Mental Health Prediction by Logistic Regression Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Calculating correct and incorrect predictions using the confusion matrix before applying K-Fold cross-validation

[21]:
```
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
cm
```

[21]:
```
array([[117,  40],
       [ 22, 136]], dtype=int64)
```

# Calculating the accuracy of the model's predictions before applying K-Fold cross-validation

[22]:
```
accuracy_score = accuracy_score(y_test, y_pred)
print('accuracy score {: 2f}%' format((accuracy_score)* 100))
```

Search ENG IN 02:23 PM 25-10-2025

Home | Mental | Home | Mental | Home | Mental | Home | Mental | Home | Mental
localhost:8888/notebooks/Mental%20Health%20Prediction%20by%20Logistic%20Regression.ipynb
jupyter Mental Health Prediction by Logistic Regression Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Calculating the accuracy of the model's predictions before applying K-Fold cross-validation

[22]:
```
accuracy_score = accuracy_score(y_test, y_pred)
print('accuracy_score {:.2f}%'.format((accuracy_score)* 100))
```

```
accuracy_score 80.32%
```

# Applying K-Fold cross-validation for more robust evaluation of the model

[23]:
```
from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator=classifier, X = X_train, y = y_train, cv = 10)
print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Standard Deviation: {:.2}%".format(accuracies.std()*100))
```

Search ENG IN 02:23 PM 25-10-2025

Home | Mental | Home | Mental | Home | Mental | Home | Mental | Home | Mental
localhost:8888/notebooks/Mental%20Health%20Prediction%20by%20Logistic%20Regression.ipynb
jupyter Mental Health Prediction by Logistic Regression Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Applying K-Fold cross-validation for more robust evaluation of the model

[23]:
```
from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator=classifier, X = X_train, y = y_train, cv = 10)
print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Standard Deviation: {:.2}%".format(accuracies.std()*100))
```

```
Accuracy: 82.10 %
Standard Deviation: 3.1%
```

# Predicting the target values for X_test and comparing them with actual values after applying K-Fold cross-validation

[24]:
```
y_pred = classifier.predict(X_test)
```

Search ENG IN 02:24 PM 25-10-2025

Home | Mental | Home | Mental | Home | Mental | Home | Mental | Home | Mental
localhost:8888/notebooks/Mental%20Health%20Prediction%20by%20Logistic%20Regression.ipynb
jupyter Mental Health Prediction by Logistic Regression Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Predicting the target values for X_test and comparing them with actual values after applying K-Fold cross-validation

[24]:
```
y_pred = classifier.predict(X_test)
comparison = pd.DataFrame({'predictied ': y_pred, 'actual ': y_test})
comparison.head(10)
```

[24]:

| | predictied | actual |
|---|---|---|
| **972** | 1.0 | 1.0 |
| **719** | 0.0 | 0.0 |
| **979** | 1.0 | 1.0 |
| **550** | 0.0 | 0.0 |
| **512** | 0.0 | 0.0 |

Search ENG IN 02:24 PM 25-10-2025

Home | Mental | Home | Mental | Home | Mental | Home | Mental | Home | Mental
localhost:8888/notebooks/Mental%20Health%20Prediction%20by%20Logistic%20Regression.ipynb
jupyter Mental Health Prediction by Logistic Regression Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Calculating correct and incorrect predictions using the confusion matrix after applying K-Fold cross-validation

[25]:
```
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
cm
```

[25]:
```
array([[117,  40],
       [ 22, 136]], dtype=int64)
```

# Calculating the accuracy of the model's predictions after applying K-Fold cross-validation

[26]:
```
accuracy_score = accuracy_score(y_test, y_pred)
```

Search
ENG IN
02:24 PM 25-10-2025

Home | Me | Home | Mental | Home | Mental | Home | Mental | Home | Mental | Home | Mental

localhost:8888/notebooks/Mental%20Health%20Prediction%20by%20Logistic%20Regression.ipynb

jupyter Mental Health Prediction by Logistic Regression Last Checkpoint: 11 days ago

File Edit View Run Kernel Settings Help Trusted

Markdown Open in... Python 3 (ipykernel)

# Calculating the accuracy of the model's predictions after applying K-Fold cross-validation

[26]:
```
accuracy_score = accuracy_score(y_test, y_pred)
print('accuracy_score {:.2f}%'.format((accuracy_score)* 100))
```

```
accuracy_score 80.32%
```

[ ]:

Search ENG IN 02:24 PM 25-10-2025

# K-Nearest Neighbour Classification

Home | Mental | Home | Me | Home | Mental | Home | Mental | Home | Mental | Home | Mental
localhost:8889/notebooks/Mental%20Health%20Prediction%20by%20K-NN.ipynb

jupyter Mental Health Prediction by K-NN Last Checkpoint: 11 days ago

File Edit View Run Kernel Settings Help Trusted

Markdown Open in... Python 3 (ipykernel)

# Applying K-NN algorithm to the dataset

[19]:
```
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=5, metric = 'minkowski', p = 2)
classifier.fit(X_train, y_train)
```

[19]:
```
▾ KNeighborsClassifier
KNeighborsClassifier()
```

[20]:
```
classifier.get_params()
```

[20]:
```
{'algorithm': 'auto',
 'leaf_size': 30,
 'metric': 'minkowski',
 'metric_params': None,
 'n_jobs': None,
 'n_neighbors': 5,
```

31°C Sunny Search ENG IN 02:52 PM 25-10-2025

Home | Mental | Home | Me | Home | Mental | Home | Mental | Home | Mental | Home | Mental
localhost:8889/notebooks/Mental%20Health%20Prediction%20by%20K-NN.ipynb
jupyter Mental Health Prediction by K-NN Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Predicting target values for X_test and comparing with actual values before applying K-Fold cross-validation

[21]:
```
y_pred = classifier.predict(X_test)
comparison = pd.DataFrame({'predictied ': y_pred, 'actual ': y_test})
comparison.head(10)
```

[21]:

| | predictied | actual |
|---|---|---|
| **972** | 1.0 | 1.0 |
| **719** | 0.0 | 0.0 |
| **979** | 0.0 | 1.0 |
| **550** | 0.0 | 0.0 |
| **512** | 0.0 | 0.0 |
| **119** | 1.0 | 0.0 |

31°C Sunny
Search
ENG IN
02:52 PM 25-10-2025

Home | Mental | Home | Me | Home | Mental | Home | Mental | Home | Mental | Home | Mental
localhost:8889/notebooks/Mental%20Health%20Prediction%20by%20K-NN.ipynb
jupyter Mental Health Prediction by K-NN Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Calculating correct and incorrect predictions using the confusion matrix before applying K-Fold cross-validation

[22]:
```
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
cm
```

[22]:
```
array([[110,  47],
       [ 64,  94]], dtype=int64)
```

# Calculating the accuracy of the model's predictions before applying K-Fold cross-validation

[23]:
```
accuracy_score = accuracy_score(y_test, y_pred)
print('accuracy_score {:.2f}%'.format((accuracy_score)* 100))
```

31°C Sunny
Search
ENG IN
02:52 PM 25-10-2025

Home | Mental | Home | Me | Home | Mental | Home | Mental | Home | Mental | Home | Mental
localhost:8889/notebooks/Mental%20Health%20Prediction%20by%20K-NN.ipynb
jupyter Mental Health Prediction by K-NN Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Calculating the accuracy of the model's predictions before applying K-Fold cross-validation

[23]:
```
accuracy_score = accuracy_score(y_test, y_pred)
print('accuracy_score {:.2f}%'.format((accuracy_score)* 100))
```

```
accuracy_score 64.76%
```

# Applying K-Fold cross-validation for more robust evaluation of the model

[24]:
```
from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator=classifier, X = X_train, y = y_train, cv = 10)
print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Standard Deviation: {:.2}%".format(accuracies.std()*100))
```

31°C Sunny
Search
ENG IN
02:52 PM 25-10-2025

Home | Mental | Home | Me | Home | Mental | Home | Mental | Home | Mental | Home | Mental
localhost:8889/notebooks/Mental%20Health%20Prediction%20by%20K-NN.ipynb
jupyter Mental Health Prediction by K-NN Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Predicting the target values for X_test and comparing them with actual values after applying K-Fold cross-validation

[25]:
```
y_pred = classifier.predict(X_test)
comparison = pd.DataFrame({'predictied ': y_pred, 'actual ': y_test})
comparison.head(10)
```

[25]:

| | predictied | actual |
|---|---|---|
| **972** | 1.0 | 1.0 |
| **719** | 0.0 | 0.0 |
| **979** | 0.0 | 1.0 |
| **550** | 0.0 | 0.0 |
| **512** | 0.0 | 0.0 |
| **119** | 1.0 | 0.0 |

31°C Sunny
Search
ENG IN
02:52 PM 25-10-2025

Home | Mental | Home | Me | Home | Mental | Home | Mental | Home | Mental | Home | Mental
localhost:8889/notebooks/Mental%20Health%20Prediction%20by%20K-NN.ipynb
jupyter Mental Health Prediction by K-NN Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Calculating correct and incorrect predictions using the confusion matrix after applying K-Fold cross-validation

[26]:
```
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
cm
```

[26]:
```
array([[110,  47],
       [ 64,  94]], dtype=int64)
```

# Calculating the accuracy of the model's predictions after applying K-Fold cross-validation

[27]:
```
accuracy_score = accuracy_score(y_test, y_pred)
print('accuracy_score {:.2f}%'.format((accuracy_score)* 100))
```

31°C Sunny
Search
ENG IN
02:52 PM 25-10-2025

Home | Mental | Home | Me | Home | Mental | Home | Mental | Home | Mental | Home | Mental
localhost:8889/notebooks/Mental%20Health%20Prediction%20by%20K-NN.ipynb

Mental Health Prediction by K-NN Last Checkpoint: 11 days ago

File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Calculating the accuracy of the model's predictions after applying K-Fold cross-validation

[27]:
```
accuracy_score = accuracy_score(y_test, y_pred)
print('accuracy_score {:.2f}%'.format((accuracy_score)* 100))
```

```
accuracy_score 64.76%
```

[ ]:

31°C Sunny
Search
ENG IN
02:52 PM 25-10-2025

# Support Vector Machine Classification

Home Mental Home Mental Home Me Home Mental Home Mental Home Mental
localhost:8890/notebooks/Mental%20Health%20Prediction%20by%20SVM.ipynb
jupyter Mental Health Prediction by SVM Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Applying SVM algorithm to the dataset

[19]:
```
from sklearn.svm import SVC
classifier = SVC(kernel='rbf', random_state=0)
classifier.fit(X_train, y_train)
```

[19]:
```
SVC
SVC(random_state=0)
```

# Predicting target values for X_test and comparing with actual values before applying K-Fold cross-validation

[20]:
```
y_pred = classifier.predict(X_test)
comparison = pd.DataFrame({'predictied ': y_pred, 'actual ': y_test})
comparison.head(10)
```

31°C Sunny Search ENG IN 02:59 PM 25-10-2025

Home Mental Home Mental Home Me Home Mental Home Mental Home Mental
localhost:8890/notebooks/Mental%20Health%20Prediction%20by%20SVM.ipynb
jupyter Mental Health Prediction by SVM Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Predicting target values for X_test and comparing with actual values before applying K-Fold cross-validation

[20]:
```
y_pred = classifier.predict(X_test)
comparison = pd.DataFrame({'predictied ': y_pred, 'actual ': y_test})
comparison.head(10)
```

[20]:

| | predictied | actual |
|---|---|---|
| **972** | 1.0 | 1.0 |
| **719** | 0.0 | 0.0 |
| **979** | 1.0 | 1.0 |
| **550** | 0.0 | 0.0 |
| **512** | 0.0 | 0.0 |
| **119** | 0.0 | 0.0 |

31°C Sunny Search ENG IN 02:59 PM 25-10-2025

Home Mental Home Mental Home Me Home Mental Home Mental Home Mental
localhost:8890/notebooks/Mental%20Health%20Prediction%20by%20SVM.ipynb
jupyter Mental Health Prediction by SVM Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Calculating correct and incorrect predictions using the confusion matrix before applying K-Fold cross-validation

[21]:
```
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
cm
```

[21]:
```
array([[107,  50],
       [ 17, 141]], dtype=int64)
```

# Calculating the accuracy of the model's predictions before applying K-Fold cross-validation

[22]:
```
accuracy_score = accuracy_score(y_test, y_pred)
print('accuracy_score {:.2f}%'.format((accuracy_score)* 100))
```

```
accuracy_score 78.73%
```

31°C Sunny Search ENG IN 02:59 PM 25-10-2025

Home Mental Home Mental Home Me Home Mental Home Mental Home Mental
localhost:8890/notebooks/Mental%20Health%20Prediction%20by%20SVM.ipynb
jupyter Mental Health Prediction by SVM Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Applying K-Fold cross-validation for more robust evaluation of the model

[23]:
```
from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator=classifier, X = X_train, y = y_train, cv = 10)
print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Standard Deviation: {:.2}%".format(accuracies.std()*100))
```

```
Accuracy: 80.42 %
Standard Deviation: 4.6%
```

[24]:
```
classifier.get_params()
```

[24]:
```
{'C': 1.0,
 'break_ties': False,
 'cache_size': 200,
 'class_weight': None,
 'coef0': 0.0,
 'decision_function_shape': 'ovr',
```

31°C Sunny Search ENG IN 02:59 PM 25-10-2025

Home Mental Home Mental Home Me Home Mental Home Mental Home Mental
localhost:8890/notebooks/Mental%20Health%20Prediction%20by%20SVM.ipynb
jupyter Mental Health Prediction by SVM Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Applying Grid Search algorithm to find the optimal parameters

[25]:
```
from sklearn.model_selection import GridSearchCV
parameters = [{'C': [0.25, 0.5, 0.75, 1], 'kernel': ['linear']},
              {'C': [0.25, 0.5, 0.75, 1], 'kernel': ['rbf'], 'gamma': [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9] }]
grid_search = GridSearchCV(estimator = classifier,
                           param_grid = parameters,
                           scoring = 'accuracy',
                           cv = 10,
                           n_jobs = -1)
grid_search.fit(X_train, y_train)
best_accuracy = grid_search.best_score_
best_parameters = grid_search.best_params_
print("Best Accuracy: {:.2f}%".format(best_accuracy*100))
print("Best Parameters: ", best_parameters)
```

```
Best Accuracy: 82.41%
Best Parameters:  {'C': 0.75, 'kernel': 'linear'}
```

31°C Sunny Search ENG IN 02:59 PM 25-10-2025

Home Mental Home Mental Home Me Home Mental Home Mental Home Mental
localhost:8890/notebooks/Mental%20Health%20Prediction%20by%20SVM.ipynb
jupyter Mental Health Prediction by SVM Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Predicting the target values for X_test and comparing them with actual values after applying K-Fold cross-validation

[26]:
```
y_pred = classifier.predict(X_test)
comparison = pd.DataFrame({'predictied ': y_pred, 'actual ': y_test})
comparison.head(10)
```

[26]:

| | predictied | actual |
|---|---|---|
| **972** | 1.0 | 1.0 |
| **719** | 0.0 | 0.0 |
| **979** | 1.0 | 1.0 |
| **550** | 0.0 | 0.0 |
| **512** | 0.0 | 0.0 |
| **119** | 0.0 | 0.0 |

31°C Sunny Search ENG IN 02:59 PM 25-10-2025

Home Mental Home Mental Home Me Home Mental Home Mental Home Mental
localhost:8890/notebooks/Mental%20Health%20Prediction%20by%20SVM.ipynb
jupyter Mental Health Prediction by SVM Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Calculating correct and incorrect predictions using the confusion matrix after applying K-Fold cross-validation

[27]:
```
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
cm
```

[27]:
```
array([[107,  50],
       [ 17, 141]], dtype=int64)
```

# Calculating the accuracy of the model's predictions after applying K-Fold cross-validation

[28]:
```
accuracy_score = accuracy_score(y_test, y_pred)
print('accuracy_score {:.2f}%'.format((accuracy_score)* 100))
```

31°C Sunny Search ENG IN 02:59 PM 25-10-2025

Home Mental Home Mental Home Me Home Mental Home Mental Home Mental
localhost:8890/notebooks/Mental%20Health%20Prediction%20by%20SVM.ipynb
jupyter Mental Health Prediction by SVM Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Calculating the accuracy of the model's predictions after applying K-Fold cross-validation

[28]:
```
accuracy_score = accuracy_score(y_test, y_pred)
print('accuracy_score {:.2f}%'.format((accuracy_score)* 100))
```

```
accuracy_score 78.73%
```

[ ]:

31°C Sunny Search ENG IN 02:59 PM 25-10-2025

# Naïve Bayes Classification

Home Mental Home Mental Home Mental Home Me Home Mental Home Mental
localhost:8891/notebooks/Mental%20Health%20Prediction%20by%20Naive%20Bayes.ipynb

jupyter Mental Health Prediction by Naive Bayes Last Checkpoint: 2 days ago

File Edit View Run Kernel Settings Help Trusted

Markdown Open in... Python 3 (ipykernel)

# Applying Naive Bayes algorithm to the dataset

[19]:
```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)
```

[19]:
```
▾ GaussianNB
GaussianNB()
```

[20]:
```
classifier.get_params()
```

[20]:
```
{'priors': None, 'var_smoothing': 1e-09}
```

# Predicting target values for X_test and comparing with actual values before applying K-Fold cross-validation

31°C Sunny Search ENG IN 03:04 PM 25-10-2025

Home Mental Home Mental Home Mental Home Me Home Mental Home Mental
localhost:8891/notebooks/Mental%20Health%20Prediction%20by%20Naive%20Bayes.ipynb
jupyter Mental Health Prediction by Naive Bayes Last Checkpoint: 2 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Predicting target values for X_test and comparing with actual values before applying K-Fold cross-validation

[21]:
```
y_pred = classifier.predict(X_test)
comparison = pd.DataFrame({'predictied ': y_pred, 'actual ': y_test})
comparison.head(10)
```

[21]:

| | predictied | actual |
|---|---|---|
| **972** | 1.0 | 1.0 |
| **719** | 1.0 | 0.0 |
| **979** | 1.0 | 1.0 |
| **550** | 1.0 | 0.0 |
| **512** | 1.0 | 0.0 |
| **119** | 1.0 | 0.0 |

31°C Sunny Search ENG IN 03:04 PM 25-10-2025

Home Mental Home Mental Home Mental Home Me Home Mental Home Mental
localhost:8891/notebooks/Mental%20Health%20Prediction%20by%20Naive%20Bayes.ipynb
jupyter Mental Health Prediction by Naive Bayes Last Checkpoint: 2 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Calculating correct and incorrect predictions using the confusion matrix before applying K-Fold cross-validation

[22]:
```
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
cm
```

[22]:
```
array([[ 14, 143],
       [  5, 153]], dtype=int64)
```

# Calculating the accuracy of the model's predictions before applying K-Fold cross-validation

[23]:
```
accuracy_score = accuracy_score(y_test, y_pred)
print('accuracy_score {:.2f}%'.format((accuracy_score)* 100))
```

31°C Sunny Search ENG IN 03:04 PM 25-10-2025

Home | Mental | Home | Mental | Home | Mental | Home | Me | Home | Mental | Home | Mental
localhost:8891/notebooks/Mental%20Health%20Prediction%20by%20Naive%20Bayes.ipynb
jupyter Mental Health Prediction by Naive Bayes Last Checkpoint: 2 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Calculating the accuracy of the model's predictions before applying K-Fold cross-validation

[23]:
```
accuracy_score = accuracy_score(y_test, y_pred)
print('accuracy_score {:.2f}%'.format((accuracy_score)* 100))
```

```
accuracy_score 53.02%
```

# Applying K-Fold cross-validation for more robust evaluation of the model

[24]:
```
from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator=classifier, X = X_train, y = y_train, cv = 10)
print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Standard Deviation: {:.2}%".format(accuracies.std()*100))
```

```
Accuracy: 54.03 %
```

31°C Sunny Search ENG IN 03:04 PM 25-10-2025

Home Mental Home Mental Home Mental Home Me Home Mental Home Mental
localhost:8891/notebooks/Mental%20Health%20Prediction%20by%20Naive%20Bayes.ipynb
jupyter Mental Health Prediction by Naive Bayes Last Checkpoint: 2 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Applying K-Fold cross-validation for more robust evaluation of the model

[24]:
```
from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator=classifier, X = X_train, y = y_train, cv = 10)
print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Standard Deviation: {:.2}%".format(accuracies.std()*100))
```

```
Accuracy: 54.03 %
Standard Deviation: 3.8%
```

# Predicting the target values for X_test and comparing them with actual values after applying K-Fold cross-validation

[25]:
```
y_pred = classifier.predict(X_test)
comparison = pd.DataFrame({'predictied ': y_pred, 'actual ': y_test})
comparison.head(10)
```

31°C Sunny Search ENG IN 03:04 PM 25-10-2025

Home Mental Home Mental Home Mental Home Me Home Mental Home Mental
localhost:8891/notebooks/Mental%20Health%20Prediction%20by%20Naive%20Bayes.ipynb
jupyter Mental Health Prediction by Naive Bayes Last Checkpoint: 2 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Predicting the target values for X_test and comparing them with actual values after applying K-Fold cross-validation

[25]:
```
y_pred = classifier.predict(X_test)
comparison = pd.DataFrame({'predictied ': y_pred, 'actual ': y_test})
comparison.head(10)
```

[25]:

| | predictied | actual |
|---|---|---|
| **972** | 1.0 | 1.0 |
| **719** | 1.0 | 0.0 |
| **979** | 1.0 | 1.0 |
| **550** | 1.0 | 0.0 |
| **512** | 1.0 | 0.0 |
| **119** | 1.0 | 0.0 |

31°C Sunny Search ENG IN 03:04 PM 25-10-2025

Home Mental Home Mental Home Mental Home Me Home Mental Home Mental
localhost:8891/notebooks/Mental%20Health%20Prediction%20by%20Naive%20Bayes.ipynb
jupyter Mental Health Prediction by Naive Bayes Last Checkpoint: 2 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Calculating correct and incorrect predictions using the confusion matrix after applying K-Fold cross-validation

[26]:
```
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
cm
```

[26]:
```
array([[ 14, 143],
       [  5, 153]], dtype=int64)
```

# Calculating the accuracy of the model's predictions after applying K-Fold cross-validation

[27]:
```
accuracy_score = accuracy_score(y_test, y_pred)
print('accuracy_score {:.2f}%'.format((accuracy_score)* 100))
```

```
accuracy_score 53.02%
```

31°C Sunny Search ENG IN 03:04 PM 25-10-2025

Home Mental Home Mental Home Mental Home Me Home Mental Home Mental
localhost:8891/notebooks/Mental%20Health%20Prediction%20by%20Naive%20Bayes.ipynb
jupyter Mental Health Prediction by Naive Bayes Last Checkpoint: 2 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Calculating the accuracy of the model's predictions after applying K-Fold cross-validation

[27]:
```
accuracy_score = accuracy_score(y_test, y_pred)
print('accuracy_score {:.2f}%'.format((accuracy_score)* 100))
```

```
accuracy_score 53.02%
```

[ ]:

31°C Sunny Search ENG IN 03:04 PM 25-10-2025

# Decision Tree Classification

Home Mental Home Mental Home Mental Home Mental Home Me Home Mental
localhost:8892/notebooks/Mental%20Health%20Prediction%20by%20Decision%20Tree.ipynb
jupyter Mental Health Prediction by Decision Tree Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Applying Decision Tree algorithm to the dataset

[19]:
```
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion = 'gini', random_state = 0)
classifier.fit(X_train, y_train)
```

[19]:
```
DecisionTreeClassifier
DecisionTreeClassifier(random_state=0)
```

[20]:
```
classifier.get_params()
```

[20]:
```
{'ccp_alpha': 0.0,
 'class_weight': None,
 'criterion': 'gini',
 'max_depth': None,
 'max_features': None,
 'max_leaf_nodes': None,
```

News for you SEBI Bars Mutua... Search ENG IN 03:59 PM 25-10-2025

Home Mental Home Mental Home Mental Home Mental Home Me Home Mental
localhost:8892/notebooks/Mental%20Health%20Prediction%20by%20Decision%20Tree.ipynb
jupyter Mental Health Prediction by Decision Tree Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Predicting target values for X_test and comparing with actual values before applying K-Fold cross-validation

[21]:
```
y_pred = classifier.predict(X_test)
comparison = pd.DataFrame({'predictied ': y_pred, 'actual ': y_test})
comparison.head(10)
```

[21]:

| | predictied | actual |
|---|---|---|
| **972** | 1.0 | 1.0 |
| **719** | 0.0 | 0.0 |
| **979** | 1.0 | 1.0 |
| **550** | 0.0 | 0.0 |
| **512** | 0.0 | 0.0 |
| **119** | 0.0 | 0.0 |

News for you SEBI Bars Mutua... Search ENG IN 03:59 PM 25-10-2025

Home Mental Home Mental Home Mental Home Mental Home Me Home Mental
localhost:8892/notebooks/Mental%20Health%20Prediction%20by%20Decision%20Tree.ipynb
jupyter Mental Health Prediction by Decision Tree Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Calculating correct and incorrect predictions using the confusion matrix before applying K-Fold cross-validation

[22]:
```
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
cm
```

[22]:
```
array([[121,  36],
       [ 38, 120]], dtype=int64)
```

# Calculating the accuracy of the model's predictions before applying K-Fold cross-validation

[23]:
```
accuracy_score = accuracy_score(y_test, y_pred)
print('accuracy_score {:.2f}%'.format((accuracy_score)* 100))
```

News for you SEBI Bars Mutua... Search ENG IN 03:59 PM 25-10-2025

Home Mental Home Mental Home Mental Home Mental Home Me Home Mental
localhost:8892/notebooks/Mental%20Health%20Prediction%20by%20Decision%20Tree.ipynb
jupyter Mental Health Prediction by Decision Tree Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Calculating the accuracy of the model's predictions before applying K-Fold cross-validation

[23]:
```
accuracy_score = accuracy_score(y_test, y_pred)
print('accuracy_score {:.2f}%'.format((accuracy_score)* 100))
```

```
accuracy_score 76.51%
```

# Applying K-Fold cross-validation for more robust evaluation of the model

[24]:
```
from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator=classifier, X = X_train, y = y_train, cv = 10)
print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Standard Deviation: {:.2}%".format(accuracies.std()*100))
```

News for you SEBI Bars Mutua... Search ENG IN 03:59 PM 25-10-2025

Home Mental Home Mental Home Mental Home Mental Home Me Home Mental
localhost:8892/notebooks/Mental%20Health%20Prediction%20by%20Decision%20Tree.ipynb
jupyter Mental Health Prediction by Decision Tree Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Applying K-Fold cross-validation for more robust evaluation of the model

[24]:
```
from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator=classifier, X = X_train, y = y_train, cv = 10)
print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Standard Deviation: {:.2}%".format(accuracies.std()*100))
```

```
Accuracy: 76.06 %
Standard Deviation: 3.6%
```

# Predicting the target values for X_test and comparing them with actual values after applying K-Fold cross-validation

[25]:
```
y_pred = classifier.predict(X_test)
comparison = pd.DataFrame({'predictied ': y_pred, 'actual ': y_test})
```

News for you SEBI Bars Mutua... Search ENG IN 03:59 PM 25-10-2025

Home Mental Home Mental Home Mental Home Mental Home Me Home Mental
localhost:8892/notebooks/Mental%20Health%20Prediction%20by%20Decision%20Tree.ipynb
jupyter Mental Health Prediction by Decision Tree Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Predicting the target values for X_test and comparing them with actual values after applying K-Fold cross-validation

[25]:
```
y_pred = classifier.predict(X_test)
comparison = pd.DataFrame({'predictied ': y_pred, 'actual ': y_test})
comparison.head(10)
```

[25]:

| | predictied | actual |
|---|---|---|
| **972** | 1.0 | 1.0 |
| **719** | 0.0 | 0.0 |
| **979** | 1.0 | 1.0 |
| **550** | 0.0 | 0.0 |
| **512** | 0.0 | 0.0 |
| **119** | 0.0 | 0.0 |

News for you SEBI Bars Mutua... Search ENG IN 03:59 PM 25-10-2025

Home Mental Home Mental Home Mental Home Mental Home Me Home Mental
localhost:8892/notebooks/Mental%20Health%20Prediction%20by%20Decision%20Tree.ipynb
jupyter Mental Health Prediction by Decision Tree Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Calculating correct and incorrect predictions using the confusion matrix after applying K-Fold cross-validation

[26]:
```
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
cm
```

[26]:
```
array([[121,  36],
       [ 38, 120]], dtype=int64)
```

# Calculating the accuracy of the model's predictions after applying K-Fold cross-validation

[27]:
```
accuracy_score = accuracy_score(y_test, y_pred)
print('accuracy_score {:.2f}%'.format((accuracy_score)* 100))
```

```
accuracy_score 76.51%
```

News for you SEBI Bars Mutua... Search ENG IN 03:59 PM 25-10-2025

Home Mental Home Mental Home Mental Home Mental Home Me Home Mental
localhost:8892/notebooks/Mental%20Health%20Prediction%20by%20Decision%20Tree.ipynb
jupyter Mental Health Prediction by Decision Tree Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Calculating the accuracy of the model's predictions after applying K-Fold cross-validation

[27]:
```
accuracy_score = accuracy_score(y_test, y_pred)
print('accuracy_score {:.2f}%'.format((accuracy_score)* 100))
```

```
accuracy_score 76.51%
```

[ ]:

News for you SEBI Bars Mutua... Search ENG IN 03:59 PM 25-10-2025

# Random Forest Classification

Home Mental Home Mental Home Mental Home Mental Home Mental Home Me
localhost:8897/notebooks/Mental%20Health%20Prediction%20by%20Random%20Forest.ipynb
jupyter Mental Health Prediction by Random Forest Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Applying Random Forest algorithm to the dataset

[19]:
```
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators=10, criterion = 'gini', random_state = 0)
classifier.fit(X_train, y_train)
```

[19]:
```
▾ RandomForestClassifier
RandomForestClassifier(n_estimators=10, random_state=0)
```

[20]:
```
classifier.get_params()
```

[20]:
```
{'bootstrap': True,
 'ccp_alpha': 0.0,
 'class_weight': None,
 'criterion': 'gini',
 'max_depth': None,
 'max_features': 'sqrt',
```

Trending videos Avatar: Fire and... Search ENG IN 04:02 PM 25-10-2025

Home Mental Home Mental Home Mental Home Mental Home Mental Home Me
localhost:8897/notebooks/Mental%20Health%20Prediction%20by%20Random%20Forest.ipynb
jupyter Mental Health Prediction by Random Forest Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Predicting target values for X_test and comparing with actual values before applying K-Fold cross-validation

[21]:
```
y_pred = classifier.predict(X_test)
comparison = pd.DataFrame({'predictied ': y_pred, 'actual ': y_test})
comparison.head(10)
```

[21]:

| | predictied | actual |
|---|---|---|
| **972** | 1.0 | 1.0 |
| **719** | 0.0 | 0.0 |
| **979** | 1.0 | 1.0 |
| **550** | 0.0 | 0.0 |
| **512** | 0.0 | 0.0 |
| **119** | 0.0 | 0.0 |

Trending videos Avatar: Fire and... Search ENG IN 04:02 PM 25-10-2025

Home Mental Home Mental Home Mental Home Mental Home Mental Home Me
localhost:8897/notebooks/Mental%20Health%20Prediction%20by%20Random%20Forest.ipynb

jupyter Mental Health Prediction by Random Forest Last Checkpoint: 11 days ago

File Edit View Run Kernel Settings Help Trusted

Markdown Open in... Python 3 (ipykernel)

# Calculating correct and incorrect predictions using the confusion matrix before applying K-Fold cross-validation

[22]:
```
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
cm
```

[22]:
```
array([[125,  32],
       [ 35, 123]], dtype=int64)
```

# Calculating the accuracy of the model's predictions before applying K-Fold cross-validation

[23]:
```
accuracy_score = accuracy_score(y_test, y_pred)
print('accuracy_score {:.2f}%'.format((accuracy_score)* 100))
```

```
accuracy_score 78.73%
```

Trending videos Avatar: Fire and... Search ENG IN 04:02 PM 25-10-2025

Home Mental Home Mental Home Mental Home Mental Home Mental Home Me
localhost:8897/notebooks/Mental%20Health%20Prediction%20by%20Random%20Forest.ipynb
jupyter Mental Health Prediction by Random Forest Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Calculating the accuracy of the model's predictions before applying K-Fold cross-validation

[23]:
```
accuracy_score = accuracy_score(y_test, y_pred)
print('accuracy_score {:.2f}%'.format((accuracy_score)* 100))
```

```
accuracy_score 78.73%
```

# Applying K-Fold cross-validation for more robust evaluation of the model

[24]:
```
from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator=classifier, X = X_train, y = y_train, cv = 10)
print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Standard Deviation: {:.2}%".format(accuracies.std()*100))
```

```
Accuracy: 79.13 %
```

Trending videos Avatar: Fire and... Search ENG IN 04:02 PM 25-10-2025

Home Mental Home Mental Home Mental Home Mental Home Mental Home Me
localhost:8897/notebooks/Mental%20Health%20Prediction%20by%20Random%20Forest.ipynb
jupyter Mental Health Prediction by Random Forest Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Applying K-Fold cross-validation for more robust evaluation of the model

[24]:
```
from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator=classifier, X = X_train, y = y_train, cv = 10)
print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Standard Deviation: {:.2}%".format(accuracies.std()*100))
```

```
Accuracy: 79.13 %
Standard Deviation: 3.1%
```

# Predicting the target values for X_test and comparing them with actual values after applying K-Fold cross-validation

[25]:
```
y_pred = classifier.predict(X_test)
comparison = pd.DataFrame({'predictied ': y_pred, 'actual ': y_test})
```

Trending videos Avatar: Fire and... Search ENG IN 04:02 PM 25-10-2025

Home Mental Home Mental Home Mental Home Mental Home Mental Home Me
localhost:8897/notebooks/Mental%20Health%20Prediction%20by%20Random%20Forest.ipynb
jupyter Mental Health Prediction by Random Forest Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Predicting the target values for X_test and comparing them with actual values after applying K-Fold cross-validation

[25]:
```
y_pred = classifier.predict(X_test)
comparison = pd.DataFrame({'predictied ': y_pred, 'actual ': y_test})
comparison.head(10)
```

[25]:

| | predictied | actual |
|---|---|---|
| **972** | 1.0 | 1.0 |
| **719** | 0.0 | 0.0 |
| **979** | 1.0 | 1.0 |
| **550** | 0.0 | 0.0 |
| **512** | 0.0 | 0.0 |
| **119** | 0.0 | 0.0 |

Trending videos Avatar: Fire and... Search ENG IN 04:02 PM 25-10-2025

Home Mental Home Mental Home Mental Home Mental Home Mental Home Me
localhost:8897/notebooks/Mental%20Health%20Prediction%20by%20Random%20Forest.ipynb
jupyter Mental Health Prediction by Random Forest Last Checkpoint: 11 days ago
File Edit View Run Kernel Settings Help Trusted
Markdown Open in... Python 3 (ipykernel)

# Calculating correct and incorrect predictions using the confusion matrix after applying K-Fold cross-validation

[26]:
```
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
cm
```

[26]:
```
array([[125,  32],
       [ 35, 123]], dtype=int64)
```

# Calculating the accuracy of the model's predictions after applying K-Fold cross-validation

[27]:
```
accuracy_score = accuracy_score(y_test, y_pred)
print('accuracy_score {:.2f}%'.format((accuracy_score)* 100))
```

Trending videos Avatar: Fire and... Search ENG IN 04:02 PM 25-10-2025

Home Mental Home Mental Home Mental Home Mental Home Mental Home Me
localhost:8897/notebooks/Mental%20Health%20Prediction%20by%20Random%20Forest.ipynb

jupyter Mental Health Prediction by Random Forest Last Checkpoint: 11 days ago

File Edit View Run Kernel Settings Help Trusted

Markdown Open in... Python 3 (ipykernel)

# Calculating the accuracy of the model's predictions after applying K-Fold cross-validation

[27]:
```
accuracy_score = accuracy_score(y_test, y_pred)
print('accuracy_score {:.2f}%'.format((accuracy_score)* 100))
```

```
accuracy_score 78.73%
```

[ ]:

Trending videos Avatar: Fire and... Search ENG IN 04:02 PM 25-10-2025

# Thank you