

SMS Spam Detection — Detailed Cell-by-Cell Summary

Cell 1 — Code:

This code cell performs operations related to SMS spam detection. It contains:

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

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 2 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
df = pd.read_csv('/content/drive/MyDrive/Colab  
Notebooks/Dataset/sms+spam+collection/SMSSpamCollection.csv')
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 3 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
df.sample(5)
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 4 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
df.shape
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 5 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
# 1. Data Cleaning # 2. Exploratory Data Analysis # 3. Text Preprocessing # 4. Model Building # 5.  
Evaluation # 6. Improvement # 7. streamlit # 8. Deployment
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 6 — Markdown:

This markdown cell provides documentation or explanation. The content introduces or explains:

1. Data Cleaning

Cell 7 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
df.info()
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 8 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
from sklearn.preprocessing import LabelEncoder encoder = LabelEncoder()
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 9 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
df['label'] = encoder.fit_transform(df['label'])
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 10 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
df.head()
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 11 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
# missing values df.isnull().sum()
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 12 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
# check for duplicate values df.duplicated().sum()
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 13 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
# remove duplicates df = df.drop_duplicates(keep='first')
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 14 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
df.duplicated().sum()
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 15 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
df.shape
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 16 — Markdown:

This markdown cell provides documentation or explanation. The content introduces or explains:
EDA

Cell 17 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
df.head()
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 18 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
df['label'].value_counts()
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 19 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
import matplotlib.pyplot as plt
plt.pie(df['label'].value_counts(), labels=['ham', 'spam'], autopct="%0.2f")
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 20 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
# Data is imbalanced
import nltk
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 21 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
!pip install nltk
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 22 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
nltk.download('punkt')
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 23 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
df['num_characters'] = df['message'].apply(len)
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 24 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
df.head()
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 25 — Code:

This code cell performs operations related to SMS spam detection. It contains:
`nltk.download('punkt_tab') df['num_words'] = df['message'].apply(lambda x:len(nltk.word_tokenize(x)))`
The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 26 — Code:

This code cell performs operations related to SMS spam detection. It contains:
`df.head()`
The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 27 — Code:

This code cell performs operations related to SMS spam detection. It contains:
`df['num_sentences'] = df['message'].apply(lambda x:len(nltk.sent_tokenize(x)))`
The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 28 — Code:

This code cell performs operations related to SMS spam detection. It contains:
`df.head()`
The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 29 — Code:

This code cell performs operations related to SMS spam detection. It contains:
`df[['num_characters', 'num_words', 'num_sentences']].describe()`
The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 30 — Code:

This code cell performs operations related to SMS spam detection. It contains:
`# Ham messages df[df['label'] == 0][['num_characters', 'num_words', 'num_sentences']].describe()`
The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 31 — Code:

This code cell performs operations related to SMS spam detection. It contains:
`# Spam df[df['label'] == 1][['num_characters', 'num_words', 'num_sentences']].describe()`
The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 32 — Code:

This code cell performs operations related to SMS spam detection. It contains:
`# Histogram import seaborn as sns`

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 33 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
plt.figure(figsize=(12, 6)) sns.histplot(df[df['label'] == 0]['num_characters']) sns.histplot(df[df['label'] == 1]['num_characters'], color='red')
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 34 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
plt.figure(figsize=(12, 6)) sns.histplot(df[df['label'] == 0]['num_words']) sns.histplot(df[df['label'] == 1]['num_words'], color='red')
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 35 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
sns.pairplot(df, hue='label')
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 36 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
sns.heatmap(df[['label', 'num_characters', 'num_words', 'num_sentences']].corr(), annot=True)
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 37 — Markdown:

This markdown cell provides documentation or explanation. The content introduces or explains:

****3. Data Preprocessing *** 1. Lower case 2. Tokenization 3. Removing special characters 4.*

Removing stop words and punctuation 5. Stemming

Cell 38 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
def transform_text(text): text = text.lower() text = nltk.word_tokenize(text) import string from nltk.corpus import stopwords from nltk.stem.porter import PorterStemmer ps = PorterStemmer() y = [] for i in text: if i.isalnum(): y.append(i) text = y[:] y.clear() for i in text: if i not in stopwords.words('english') and i not in string.punctuation: y.append(i) text = y[:] y.clear() for i in text:
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 39 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
nltk.download('stopwords') transform_text("I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today")
```

The cell executes functions essential for preprocessing, feature extraction, model training, or

evaluation.

Cell 40 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
df['message'][10]
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 41 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
from nltk.stem.porter import PorterStemmer ps = PorterStemmer() ps.stem('loving')
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 42 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
df['transformed_text'] = df['message'].apply(transform_text)
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 43 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
df.head()
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 44 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
from wordcloud import WordCloud wc = WordCloud(width=500,height=500,min_font_size=10,background_color='white')
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 45 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
spam_wc = wc.generate(df[df['label'] == 1]['transformed_text'].str.cat(sep=" "))
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 46 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
plt.figure(figsize=(15,6)) plt.imshow(spam_wc)
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 47 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
ham_wc = wc.generate(df[df['label'] == 0]['transformed_text'].str.cat(sep=" "))
```

The cell executes functions essential for preprocessing, feature extraction, model training, or

evaluation.

Cell 48 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
plt.figure(figsize=(15,6)) plt.imshow(ham_wc)
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 49 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
df.head()
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 50 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
spam_corpus = [] for msg in df[df['label'] == 1]['transformed_text'].tolist(): for word in msg.split(): spam_corpus.append(word)
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 51 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
len(spam_corpus)
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 52 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
from collections import Counter
```

```
sns.barplot(x=pd.DataFrame(Counter(spam_corpus).most_common(30))[0],
```

```
y=pd.DataFrame(Counter(spam_corpus).most_common(30))[1]) plt.xticks(rotation='vertical') plt.show()
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 53 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
ham_corpus = [] for msg in df[df['label'] == 0]['transformed_text'].tolist(): for word in msg.split(): ham_corpus.append(word)
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 54 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
len(ham_corpus)
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 55 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
from collections import Counter  
sns.barplot(x=pd.DataFrame(Counter(ham_corpus).most_common(30))[0],  
y=pd.DataFrame(Counter(ham_corpus).most_common(30))[1]) plt.xticks(rotation='vertical') plt.show()
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 56 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
# Text Vectorization # using Bag of Words df.head()
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 57 — Markdown:

This markdown cell provides documentation or explanation. The content introduces or explains:

Model Building

Cell 58 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer cv = CountVectorizer() tfidf  
= TfidfVectorizer(max_features=3000)
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 59 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
X = tfidf.fit_transform(df['transformed_text']).toarray()
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 60 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
#from sklearn.preprocessing import MinMaxScaler #scaler = MinMaxScaler() #X =  
scaler.fit_transform(X) # appending the num_character col to X #X =  
np.hstack((X,df['num_characters'].values.reshape(-1,1)))
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 61 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
X.shape
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 62 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
y = df['label'].values
```

The cell executes functions essential for preprocessing, feature extraction, model training, or

evaluation.

Cell 63 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
from sklearn.model_selection import train_test_split
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 64 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=2)
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 65 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
from sklearn.naive_bayes import GaussianNB,MultinomialNB,BernoulliNB from sklearn.metrics import accuracy_score,confusion_matrix,precision_score
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 66 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
gnb = GaussianNB() mnb = MultinomialNB() bnb = BernoulliNB()
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 67 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
gnb.fit(X_train,y_train) y_pred1 = gnb.predict(X_test) print(accuracy_score(y_test,y_pred1))
```

```
print(confusion_matrix(y_test,y_pred1)) print(precision_score(y_test,y_pred1))
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 68 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
mnb.fit(X_train,y_train) y_pred2 = mnb.predict(X_test) print(accuracy_score(y_test,y_pred2))
```

```
print(confusion_matrix(y_test,y_pred2)) print(precision_score(y_test,y_pred2))
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 69 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
bnb.fit(X_train,y_train) y_pred3 = bnb.predict(X_test) print(accuracy_score(y_test,y_pred3))
```

```
print(confusion_matrix(y_test,y_pred3)) print(precision_score(y_test,y_pred3))
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 70 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
from sklearn.linear_model import LogisticRegression from sklearn.svm import SVC from
sklearn.naive_bayes import MultinomialNB from sklearn.tree import DecisionTreeClassifier from
sklearn.neighbors import KNeighborsClassifier from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier from sklearn.ensemble import BaggingClassifier from
sklearn.ensemble import ExtraTreesClassifier from sklearn.ensemble import
GradientBoostingClassifier from xgboost import XG
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 71 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
svc = SVC(kernel='sigmoid', gamma=1.0) knc = KNeighborsClassifier() mnb = MultinomialNB() dtc =
DecisionTreeClassifier(max_depth=5) lrc = LogisticRegression(solver='liblinear', penalty='l1') rfc =
RandomForestClassifier(n_estimators=50, random_state=2) abc =
AdaBoostClassifier(n_estimators=50, random_state=2) bc = BaggingClassifier(n_estimators=50,
random_state=2) etc = ExtraTreesClassifier(n_estimators=50, random_state=2) gbdt =
GradientBoostingClassifier(n_estimators=50,random_state=2) xgb = X
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 72 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
clfs = { 'SVC' : svc, 'KN' : knc, 'NB': mnb, 'DT': dtc, 'LR': lrc, 'RF': rfc, 'AdaBoost': abc, 'BgC': bc, 'ETC':
etc, 'GBDT':gbdt, 'xgb':xgb }
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 73 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
def train_classifier(clf,X_train,y_train,X_test,y_test): clf.fit(X_train,y_train) y_pred = clf.predict(X_test)
accuracy = accuracy_score(y_test,y_pred) precision = precision_score(y_test,y_pred) return
accuracy,precision
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 74 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
train_classifier(svc,X_train,y_train,X_test,y_test)
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 75 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
accuracy_scores = [] precision_scores = [] for name,clf in clfs.items():
current_accuracy,current_precision = train_classifier(clf, X_train,y_train,X_test,y_test) print("For
",name) print("Accuracy - ",current_accuracy) print("Precision - ",current_precision)
accuracy_scores.append(current_accuracy) precision_scores.append(current_precision)
```

The cell executes functions essential for preprocessing, feature extraction, model training, or

evaluation.

Cell 76 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
performance_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy':accuracy_scores,'Precision':precision_scores}).sort_values('Precision',ascending=False)
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 77 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
performance_df
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 78 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
performance_df1 = pd.melt(performance_df, id_vars = "Algorithm")
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 79 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
performance_df1
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 80 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
sns.catplot(x = 'Algorithm', y='value', hue = 'variable',data=performance_df1, kind='bar',height=5)  
plt.ylim(0.5,1.0) plt.xticks(rotation='vertical') plt.show()
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 81 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
# model improve # 1. Change the max_features parameter of TfIdf
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 82 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
temp_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy_max_ft_3000':accuracy_scores,'Precision_max_ft_3000':precision_scores}).sort_values('Precision_max_ft_3000',ascending=False)
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 83 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
temp_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy_scaling':accuracy_scores,'Precision_scaling':precision_scores}).sort_values('Precision_scaling',ascending=False)
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 84 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
new_df = performance_df.merge(temp_df,on='Algorithm')
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 85 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
new_df_scaled = new_df.merge(temp_df,on='Algorithm')
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 86 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
temp_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy_num_chars':accuracy_scores,'Precision_num_chars':precision_scores}).sort_values('Precision_num_chars',ascending=False)
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 87 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
new_df_scaled.merge(temp_df,on='Algorithm')
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 88 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
# Voting Classifier
svc = SVC(kernel='sigmoid', gamma=1.0, probability=True)
mnb = MultinomialNB()
etc = ExtraTreesClassifier(n_estimators=50, random_state=2)
from sklearn.ensemble import VotingClassifier
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 89 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
voting = VotingClassifier(estimators=[('svm', svc), ('nb', mnb), ('et', etc)], voting='soft')
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 90 — Code:

This code cell performs operations related to SMS spam detection. It contains:

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

The cell executes functions essential for preprocessing, feature extraction, model training, or

evaluation.

Cell 91 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
y_pred = voting.predict(X_test) print("Accuracy",accuracy_score(y_test,y_pred))
```

```
print("Precision",precision_score(y_test,y_pred))
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 92 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
# Applying stacking estimators=[('svm', svc), ('nb', mnb), ('et', etc)]
```

```
final_estimator=RandomForestClassifier()
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 93 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
from sklearn.ensemble import StackingClassifier
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 94 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
clf = StackingClassifier(estimators=estimators, final_estimator=final_estimator)
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 95 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
clf.fit(X_train,y_train) y_pred = clf.predict(X_test) print("Accuracy",accuracy_score(y_test,y_pred))
```

```
print("Precision",precision_score(y_test,y_pred))
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.

Cell 96 — Code:

This code cell performs operations related to SMS spam detection. It contains:

```
import pickle pickle.dump(tfidf,open('vectorizer.pkl','wb')) pickle.dump(mnb,open('model.pkl','wb'))
```

The cell executes functions essential for preprocessing, feature extraction, model training, or evaluation.