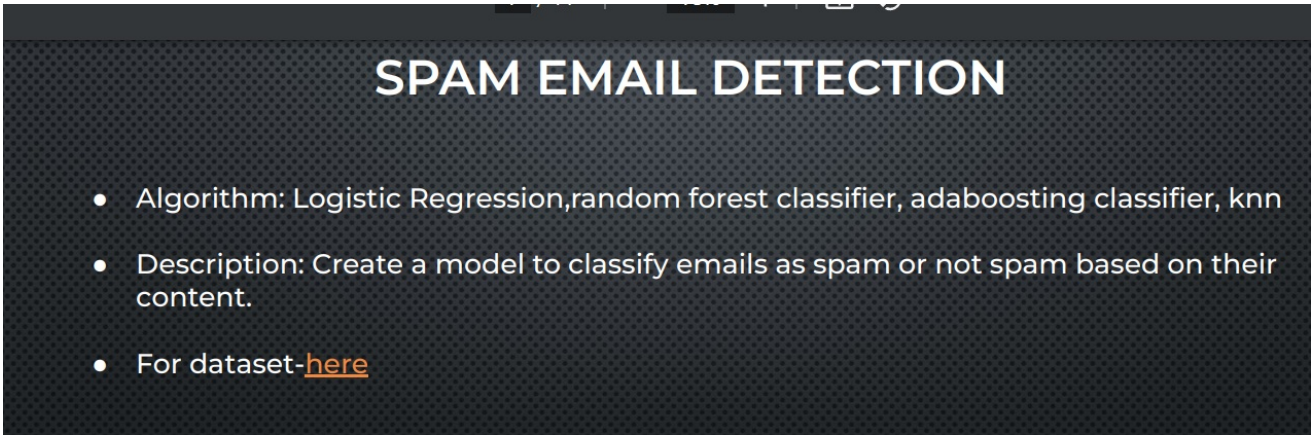


In [63]:
```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [64]:
```
df=pd.read_csv('spam.csv')
df
```

Out[64]:

	Category	Message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...
...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...
5568	ham	Will ü b going to esplanade fr home?
5569	ham	Pity, * was in mood for that. So...any other s...
5570	ham	The guy did some bitching but I acted like i'd...
5571	ham	Rofl. Its true to its name

5572 rows × 2 columns

In [96]:
```
print("The Number of rows =",df.shape[0])
print("The Number of columns =",df.shape[1])
```

```
The Number of rows = 5572
The Number of columns = 2
```

In [66]:
```
# Display the first few rows of the dataset to understand its structure
df.head()
```

Out[66]:

	Category	Message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

In [67]:
```
# Display the last few rows of the dataset to understand its structure
df.tail()
```

Out[67]:

	Category	Message
5567	spam	This is the 2nd time we have tried 2 contact u...
5568	ham	Will ü b going to esplanade fr home?
5569	ham	Pity, * was in mood for that. So...any other s...
5570	ham	The guy did some bitching but I acted like i'd...
5571	ham	Rofl. Its true to its name

In [68]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Category  5572 non-null   object
 1   Message   5572 non-null   object
dtypes: object(2)
memory usage: 87.2+ KB
```

In [69]:
```
pd.isnull(df).sum()
```

Out[69]:
```
Category    0
Message     0
dtype: int64
```

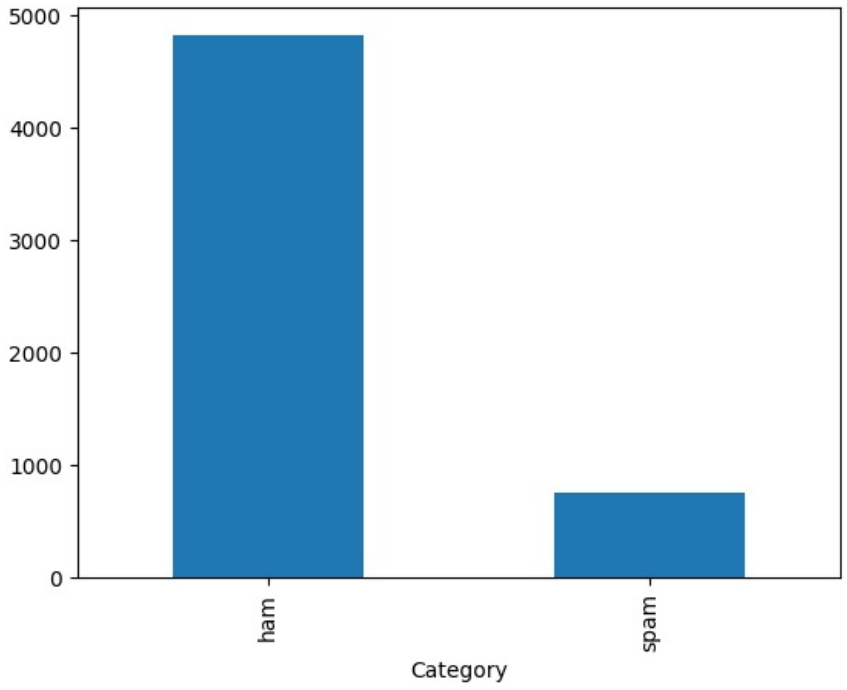
In [70]:
```
df.describe()
```

Out[70]:

	Category	Message
count	5572	5572
unique	2	5157
top	ham	Sorry, I'll call later
freq	4825	30

In [71]:
```
df['Category'].value_counts().plot(kind='bar')
```

Out[71]:
```
<Axes: xlabel='Category'>
```



In [72]:
```
# Encode the labels (ham = 0, spam = 1)
label_encoder = LabelEncoder()
df['Category'] = label_encoder.fit_transform(df['Category'])
```

In [73]:
```
# Split the data into features (X) and target (y)
x = df['Message']#independent variable
y = df['Category']#dependent variable
```

In [74]:
```
x
```

Out[74]:
```
0       Go until jurong point, crazy.. Available only ...
1                           Ok lar... Joking wif u oni...
2       Free entry in 2 a wkly comp to win FA Cup fina...
3       U dun say so early hor... U c already then say...
4       Nah I don't think he goes to usf, he lives aro...
                              ...
5567    This is the 2nd time we have tried 2 contact u...
5568                 Will ü b going to esplanade fr home?
5569    Pity, * was in mood for that. So...any other s...
5570    The guy did some bitching but I acted like i'd...
5571                          Rofl. Its true to its name
Name: Message, Length: 5572, dtype: object
```

In [75]:
```
y
```

Out[75]:
```
0       0
1       0
2       1
3       0
4       0
       ..
5567    1
5568    0
5569    0
5570    0
5571    0
Name: Category, Length: 5572, dtype: int64
```

In [76]:
```
# Convert the text data to numerical data using TF-IDF
vectorizer_trans = TfidfVectorizer(stop_words='english', max_df=0.7)
x_trans = vectorizer_trans.fit_transform(x)
```

In [77]:
```
# Split the data into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(x_trans, y, test_size=0.2, random_state=42)
```

In [78]:
```
x_train.shape, x_test.shape, y_train.shape, y_test.shape
```

Out[78]:
```
((4457, 8440), (1115, 8440), (4457,), (1115,))
```

In [79]:
```
# Initialize the models
Logis_model = LogisticRegression(max_iter=1000)
Rando_model = RandomForestClassifier(n_estimators=100)
AdaBoos_model = AdaBoostClassifier(n_estimators=100)
KNe_model = KNeighborsClassifier()
```

In [80]:
```
# Train the models
Logis_model.fit(x_train, y_train)
Rando_model.fit(x_train, y_train)
AdaBoos_model.fit(x_train, y_train)
KNe_model.fit(x_train, y_train)
```

Out[80]:
```
KNeighborsClassifier
KNeighborsClassifier()
```

In [81]:
```
# Predict the test set results
y_predict1 = Logis_model.predict(x_test)
y_predict2 = Rando_model.predict(x_test)
y_predict3 = AdaBoos_model.predict(x_test)
y_predict4 = KNe_model.predict(x_test)
```

In [82]:
```
#checking accuracy of the model
print('Logistin_Regression_Accuracy',accuracy_score(y_test, y_predict1))
print('Report',classification_report(y_test, y_predict1))
print('Confusion_matrix',confusion_matrix(y_test, y_predict1))
```

```
Logistin_Regression_Accuracy 0.95695067264574
Report               precision    recall  f1-score   support

           0       0.95      1.00      0.98       966
           1       1.00      0.68      0.81       149

    accuracy                           0.96      1115
   macro avg       0.98      0.84      0.89      1115
weighted avg       0.96      0.96      0.95      1115

Confusion_matrix [[966   0]
 [ 48 101]]
```

In [83]:
```
#checking accuracy of the model
print('Random_Forest_Model_Accuracy',accuracy_score(y_test, y_predict2))
print('Report',classification_report(y_test, y_predict2))
print('Confusion_matrix',confusion_matrix(y_test, y_predict2))
```

```
Random_Forest_Model_Accuracy 0.9829596412556054
Report               precision    recall  f1-score   support

           0       0.98      1.00      0.99       966
           1       1.00      0.87      0.93       149

    accuracy                           0.98      1115
   macro avg       0.99      0.94      0.96      1115
weighted avg       0.98      0.98      0.98      1115

Confusion_matrix [[966   0]
 [ 19 130]]
```

In [84]:
```
#checking accuracy of the model
print('AdaBoost_model',accuracy_score(y_test, y_predict3))
print('Report',classification_report(y_test, y_predict3))
print('Confusion_matrix',confusion_matrix(y_test, y_predict3))
```

```
AdaBoost_model 0.9739910313901345
Report               precision    recall  f1-score   support

           0       0.98      0.99      0.99       966
           1       0.96      0.84      0.90       149

    accuracy                           0.97      1115
   macro avg       0.97      0.92      0.94      1115
weighted avg       0.97      0.97      0.97      1115

Confusion_matrix [[961   5]
 [ 24 125]]
```

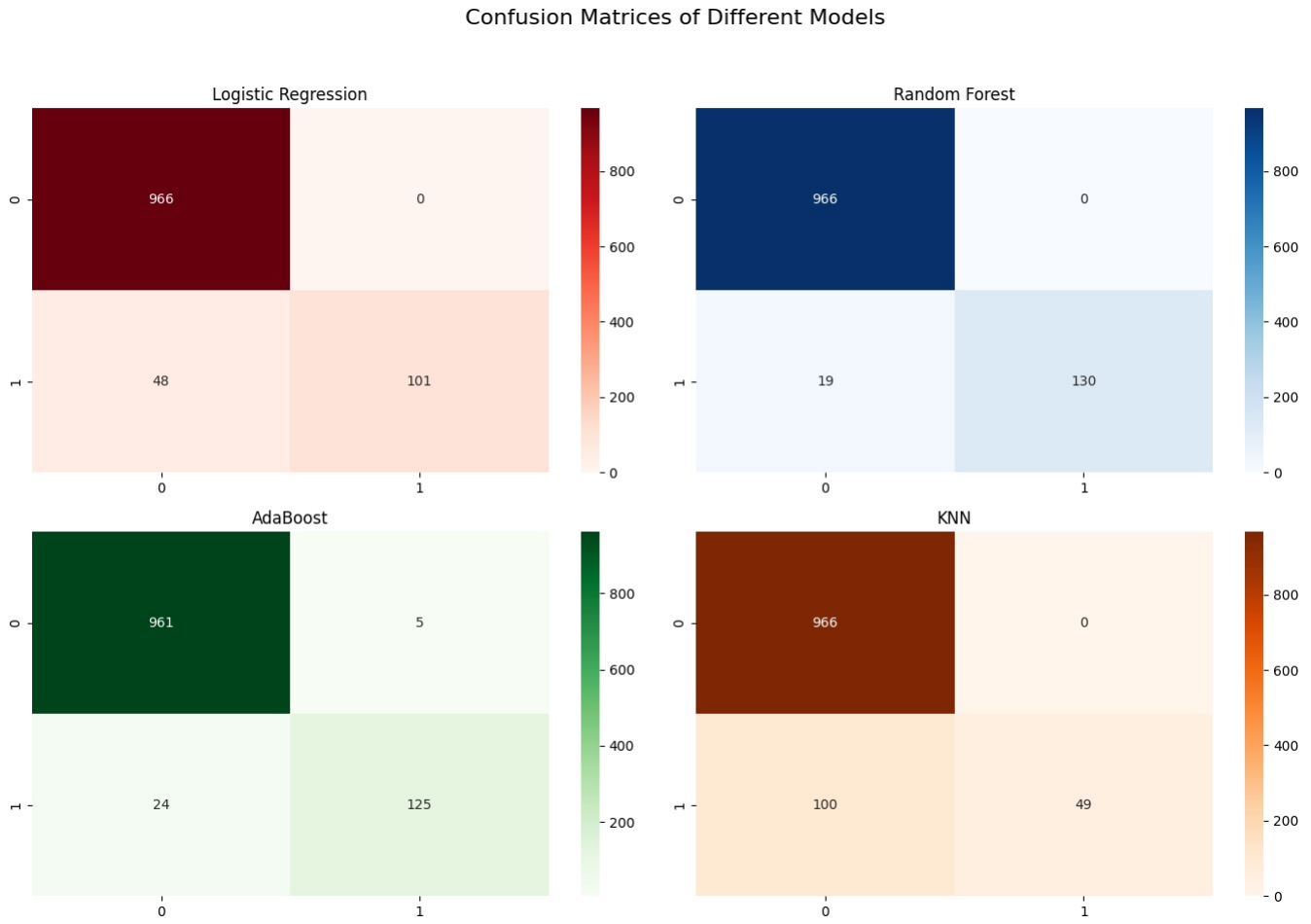
In [85]:
```
#checking accuracy of the model
print('KNN_model',accuracy_score(y_test, y_predict4))
print('Report',classification_report(y_test, y_predict4))
print('Confusion_matrix',confusion_matrix(y_test, y_predict4))
```

```
KNN_model 0.9103139013452914
Report               precision    recall  f1-score   support

           0       0.91      1.00      0.95       966
           1       1.00      0.33      0.49       149

    accuracy                           0.91      1115
   macro avg       0.95      0.66      0.72      1115
weighted avg       0.92      0.91      0.89      1115

Confusion_matrix [[966   0]
 [100  49]]
```

In [86]:
```
# Evaluate the models
results = {
    'Logistic Regression': {
        'Accuracy': accuracy_score(y_test, y_predict1),
        'Report': classification_report(y_test, y_predict1),
        'Confusion Matrix': confusion_matrix(y_test, y_predict1)
    },
    'Random Forest': {
        'Accuracy': accuracy_score(y_test, y_predict2),
        'Report': classification_report(y_test, y_predict2),
        'Confusion Matrix': confusion_matrix(y_test, y_predict2)
    },
    'AdaBoost': {
        'Accuracy': accuracy_score(y_test, y_predict3),
        'Report': classification_report(y_test, y_predict3),
        'Confusion Matrix': confusion_matrix(y_test, y_predict3)
    },
    'KNN': {
        'Accuracy': accuracy_score(y_test, y_predict4),
        'Report': classification_report(y_test, y_predict4),
        'Confusion Matrix': confusion_matrix(y_test, y_predict4)
    }
}
```

In [87]:
```
# Plot the confusion matrices with different colors
fig, axes = plt.subplots(2, 2, figsize=(14, 10))
fig.suptitle('Confusion Matrices of Different Models', fontsize=16)

sns.heatmap(results['Logistic Regression']['Confusion Matrix'], annot=True, fmt='d', cmap='Reds', ax=axes[0, 0]
axes[0, 0].set_title('Logistic Regression')

sns.heatmap(results['Random Forest']['Confusion Matrix'], annot=True, fmt='d', cmap='Blues', ax=axes[0, 1])
axes[0, 1].set_title('Random Forest')

sns.heatmap(results['AdaBoost']['Confusion Matrix'], annot=True, fmt='d', cmap='Greens', ax=axes[1, 0])
axes[1, 0].set_title('AdaBoost')

sns.heatmap(results['KNN']['Confusion Matrix'], annot=True, fmt='d', cmap='Oranges', ax=axes[1, 1])
axes[1, 1].set_title('KNN')
```

```
plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```



In [88]:
```
# Define a small new dataset containing both ham and spam to test how our model predicts when new test data is
new_data = [
    "Win a $1000 Walmart gift card. Go to http://bit.ly/123456 to claim now.",
    "Hey, I saw your profile on LinkedIn, let's connect!",
    "URGENT! Your account has been compromised. Reply with your password to secure it.",
    "Can we reschedule our meeting to next week?",
    "Congratulations, you've been selected for a free cruise to the Bahamas!",
    "Don't forget to pick up the groceries on your way home.",
    "Get the best rates for your car insurance now.",
    "Hey, just wanted to check if you're coming to the party tonight?",
    "Exclusive offer! Buy one get one free at our store!",
    "Hi, can you send me the files by EOD?"
]
```

In [89]:
```
# Convert the new data to the same format (TF-IDF) as the training data
new_data_trans = vectorizer_trans.transform(new_data)

# Predict using the trained models
pred_logis = Logis_model.predict(new_data_trans)
pred_rando = Rando_model.predict(new_data_trans)
pred_adaboost = AdaBoos_model.predict(new_data_trans)
pred_knn = KNe_model.predict(new_data_trans)
```

In [90]:
```
# Combine the results into a DataFrame for easy comparison
results_df = pd.DataFrame({
    'Message': new_data,
    'Logistic Regression': label_encoder.inverse_transform(pred_logis),
    'Random Forest': label_encoder.inverse_transform(pred_rando),
    'AdaBoost': label_encoder.inverse_transform(pred_adaboost),
    'KNN': label_encoder.inverse_transform(pred_knn)
})

results_df
```

Out[90]:

	Message	Logistic Regression	Random Forest	AdaBoost	KNN
0	Win a $1000 Walmart gift card. Go to http://bi...	spam	spam	spam	ham
1	Hey, I saw your profile on LinkedIn, let's con...	ham	ham	ham	ham
2	URGENT! Your account has been compromised. Rep...	ham	ham	spam	ham
3	Can we reschedule our meeting to next week?	ham	ham	ham	ham
4	Congratulations, you've been selected for a fr...	ham	ham	ham	ham
5	Don't forget to pick up the groceries on your ...	ham	ham	ham	ham
6	Get the best rates for your car insurance now.	ham	ham	ham	ham
7	Hey, just wanted to check if you're coming to ...	ham	ham	ham	ham
8	Exclusive offer! Buy one get one free at our s...	ham	ham	ham	ham
9	Hi, can you send me the files by EOD?	ham	ham	ham	ham

In [ ]:

In [ ]: