importing all libraries

In [1]:
```
import os
import re
import sys
import nltk
import itertools
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn import tree
from sklearn.svm import SVC
from joblib import dump, load
from wordcloud import WordCloud
import matplotlib.pyplot as plt
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer
from keras.models import Sequential,Model
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.naive_bayes import MultinomialNB
from sklearn.preprocessing import LabelEncoder
from keras.utils.np_utils import to_categorical
from sklearn.ensemble import AdaBoostClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from keras.layers import Dense,LSTM, SpatialDropout1D, Embedding
```

Reading the Dataset

In [2]:
```
df = pd.read_csv('/kaggle/input/resume-dataset/UpdatedResumeDataSet.csv')
df
```

Out[2]:

	Category	Resume
0	Data Science	Skills * Programming Languages: Python (pandas...
1	Data Science	Education Details \r\nMay 2013 to May 2017 B.E...
2	Data Science	Areas of Interest Deep Learning, Control Syste...
3	Data Science	Skills â€¢ R â€¢ Python â€¢ SAP HANA â€¢ Table...
4	Data Science	Education Details \r\n MCA YMCAUST, Faridab...
...	...	...
957	Testing	Computer Skills: â€¢ Proficient in MS office (...
958	Testing	â Willingness to accept the challenges. â ...
959	Testing	PERSONAL SKILLS â€¢ Quick learner, â€¢ Eagerne...
960	Testing	COMPUTER SKILLS & SOFTWARE KNOWLEDGE MS-Power ...
961	Testing	Skill Set OS Windows XP/7/8/8.1/10 Database MY...

962 rows × 2 columns

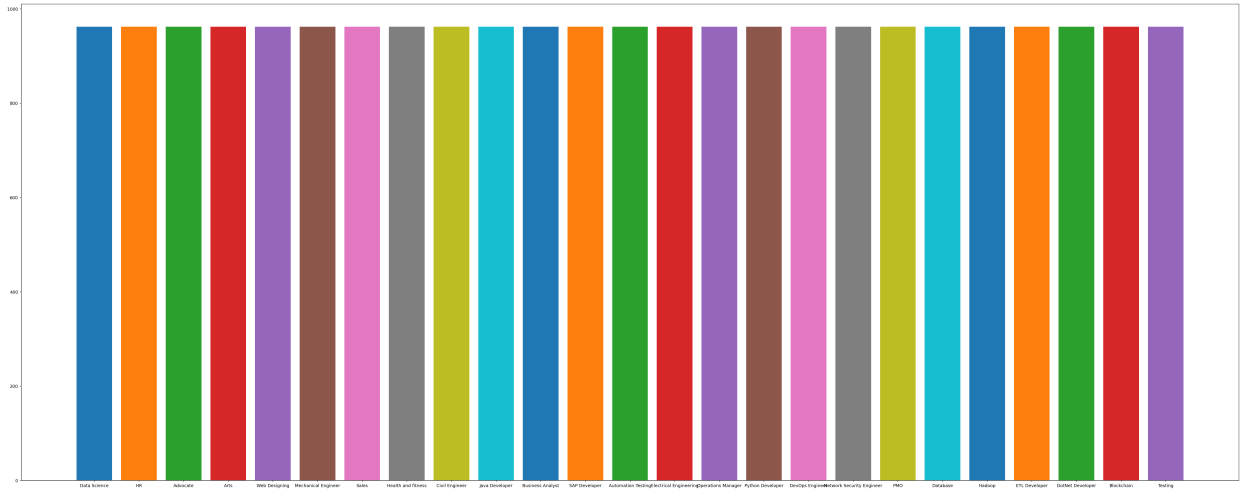
List of all Categories

In [3]:
```
for i in range(len(df['Category'].unique())):
    print(df['Category'].unique()[i])
```

```
Data Science
HR
Advocate
Arts
Web Designing
Mechanical Engineer
Sales
Health and fitness
Civil Engineer
Java Developer
Business Analyst
SAP Developer
Automation Testing
Electrical Engineering
Operations Manager
Python Developer
DevOps Engineer
Network Security Engineer
PMO
Database
Hadoop
ETL Developer
DotNet Developer
Blockchain
Testing
```

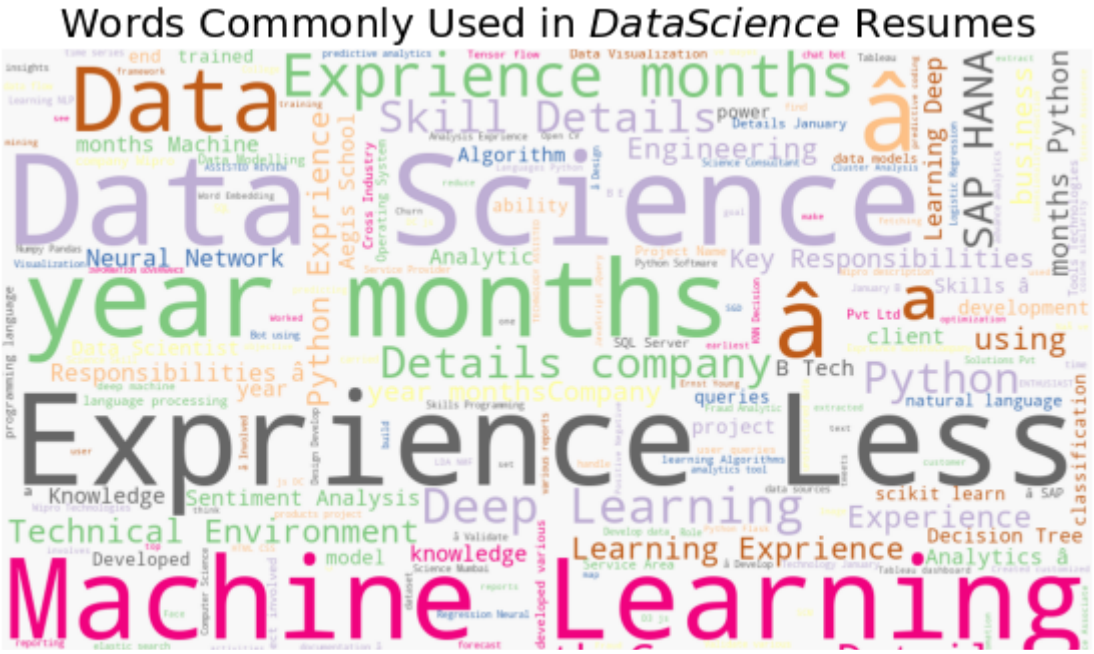
Visualizing the data distribution in each category.

In [4]:
```
plt.figure(figsize=(50, 20), dpi=130)
for i in range(len(df['Category'].unique())):
    plt.bar(df['Category'].unique()[i], len(df['Category']==df['Category'].
plt.show()
```



Visualizing most commonly used words in each type of Resumes

In [5]:
```
a=[ 'Accent_r', 'Blues', 'Blues_r', 'BrBG', 'viridis', 'viridis_r', 'vlag',
for label, cmap in zip(df['Category'].unique(), a):
    text = df.query("Category == @label")["Resume"].str.cat(sep=" ")
    plt.figure(figsize=(10, 6))
    wc = WordCloud(width=1000, height=600, background_color="#f8f8f8", colo
    wc.generate_from_text(text)
    plt.imshow(wc)
    plt.axis("off")
    plt.title(f"Words Commonly Used in ${label}$ Resumes", size=20)
    plt.show()
```

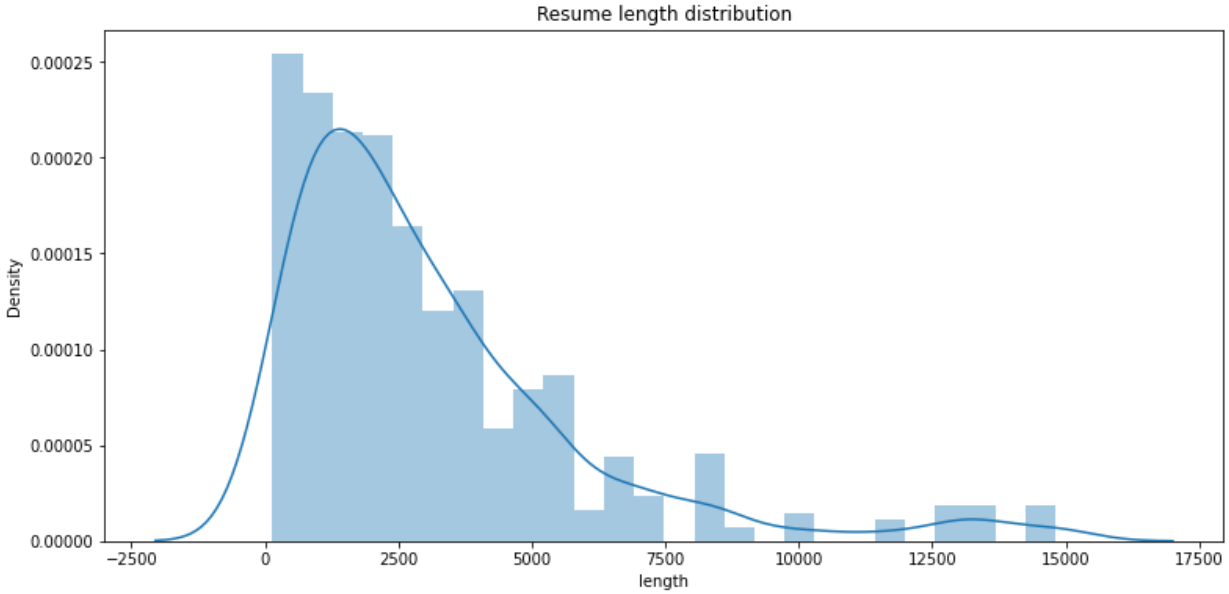


Visualizing the length disstribution of the content in the Resumes.

In [6]:
```
df['length'] = df['Resume'].str.len()
plt.figure(figsize=(12.8,6))
sns.distplot(df['length']).set_title('Resume length distribution')
```

```
/opt/conda/lib/python3.7/site-packages/seaborn/distributions.py:2619: Fut
ureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-
level function with similar flexibility) or `histplot` (an axes-level fun
ction for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[6]:
```
Text(0.5, 1.0, 'Resume length distribution')
```



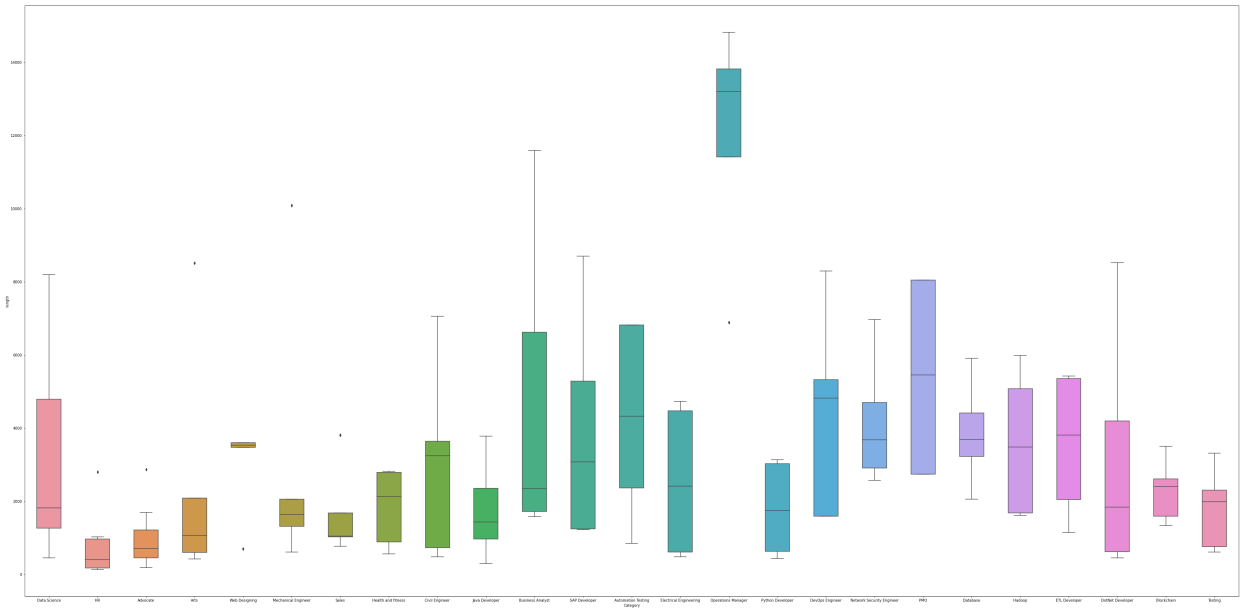
Summary of Length Distribution

In [7]:
```
df['length'].describe()
```

Out[7]:
```
count      962.000000
mean      3160.364865
std       2886.528521
min        142.000000
25%       1217.250000
50%       2355.000000
75%       4073.750000
max      14816.000000
Name: length, dtype: float64
```

Visualizing of number of words in each category of resume using boxplot

In [8]:
```
plt.figure(figsize=(60,30))
sns.boxplot(data=df, x='Category', y='length', width=.5);
```



Pre Processing

Checking for missing data

In [9]:
```
print(df.isnull().sum())
```

```
Category    0
Resume      0
length      0
dtype: int64
```

Converting the data into lower case and removing words with small lengths

In [10]:
```
df['Resume'] = df['Resume'].apply(lambda x:x.lower())
for i in range(len(df)):
    lw=[]
    for j in df['Resume'][i].split():
        if len(j)>=3:
            lw.append(j)
    df['Resume'][i]=" ".join(lw)
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:7: SettingWi
thCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  import sys
```

removing punctuations

In [11]:
```
ps = list(";?.:!,")
df['Resume'] = df['Resume']

for p in ps:
    df['Resume'] = df['Resume'].str.replace(p, '')
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:5: FutureWar
ning: The default value of regex will change from True to False in a futu
re version. In addition, single character regular expressions will *not*
be treated as literal strings when regex=True.
  """
```

Removing '\n' and '\t', extra spaces, quoting text and progressive pronouns

In [12]:
```
df['Resume'] = df['Resume'].str.replace("    ", " ")
df['Resume'] = df['Resume'].str.replace('"', '')
df['Resume'] = df['Resume'].apply(lambda x: x.replace('\t', ' '))
df['Resume'] = df['Resume'].str.replace("'s", "")
df['Resume'] = df['Resume'].apply(lambda x: x.replace('\n', ' '))
```

Applying Lemmatization

In [13]:
```
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package punkt to /usr/share/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to /usr/share/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to /usr/share/nltk_data...
```

Out[13]:
```
True
```

In [14]:
```
wl = WordNetLemmatizer()
nr = len(df)
lis = []
for r in range(0, nr):
    ll = []
    t = df.loc[r]['Resume']
    tw = str(t).split(" ")
    for w in tw:
        ll.append(wl.lemmatize(w, pos="v"))
    lt = " ".join(ll)
    lis.append(lt)
```

In [15]:
```
df['Resume'] = lis
```

Removing Stop-words

In [16]:
```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /usr/share/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

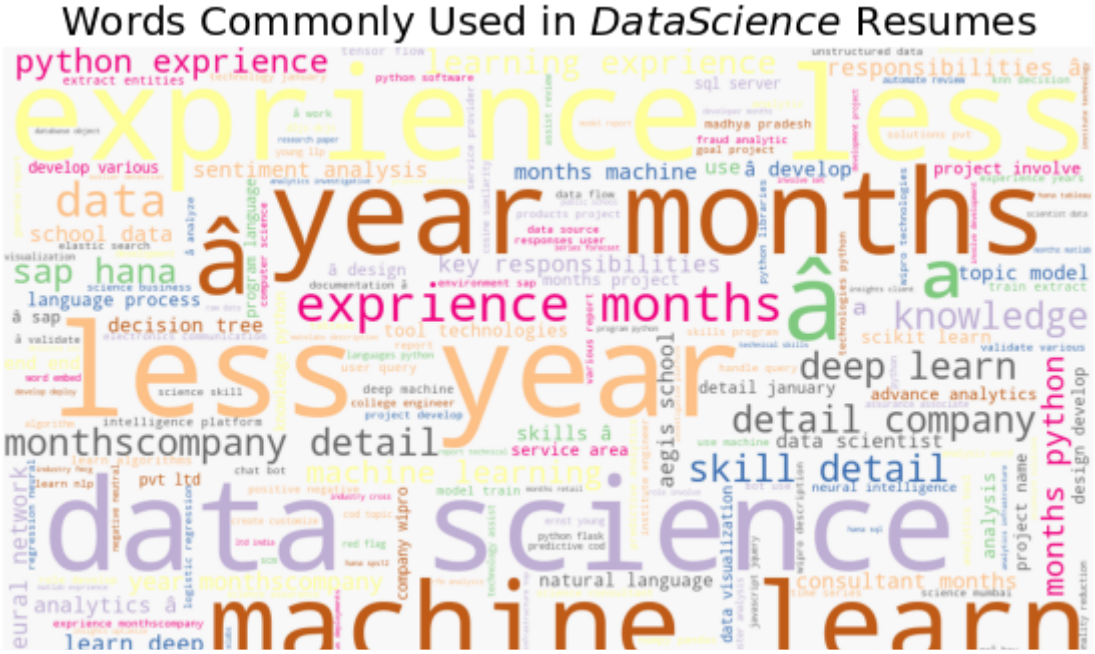
Out[16]:
```
True
```

In [17]:
```
sw = list(stopwords.words('english'))
for s in sw:
    rs = r"\b" + s + r"\b"
    df['Resume'] = df['Resume'].str.replace(rs, '')
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:4: FutureWar
ning: The default value of regex will change from True to False in a futu
re version.
  after removing the cwd from sys.path.
```

Visualizing most commonly used words in Resumes after applying NLP techniques

In [18]:
```
a=[ 'Accent_r', 'Blues', 'Blues_r', 'BrBG', 'viridis', 'viridis_r', 'vlag',
for label, cmap in zip(df['Category'].unique(), a):
    text = df.query("Category == @label")["Resume"].str.cat(sep=" ")
    plt.figure(figsize=(10, 6))
    wc = WordCloud(width=1000, height=600, background_color="#f8f8f8", colo
    wc.generate_from_text(text)
    plt.imshow(wc)
    plt.axis("off")
    plt.title(f"Words Commonly Used in ${label}$ Resumes", size=20)
    plt.show()
```



Data Preparation for Training and Testing

Encoding Labels

In [19]:
```
df['Category'] = LabelEncoder().fit_transform(df['Category'])
```

Using Bag of Words approach for converting the content in resumes into vector form

In [20]:
```
cv = CountVectorizer(max_features = 2500)
X = cv.fit_transform(df['Resume']).toarray()
y = df['Category']
```

Splitting the Data using Stratified split

In [21]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3,
```

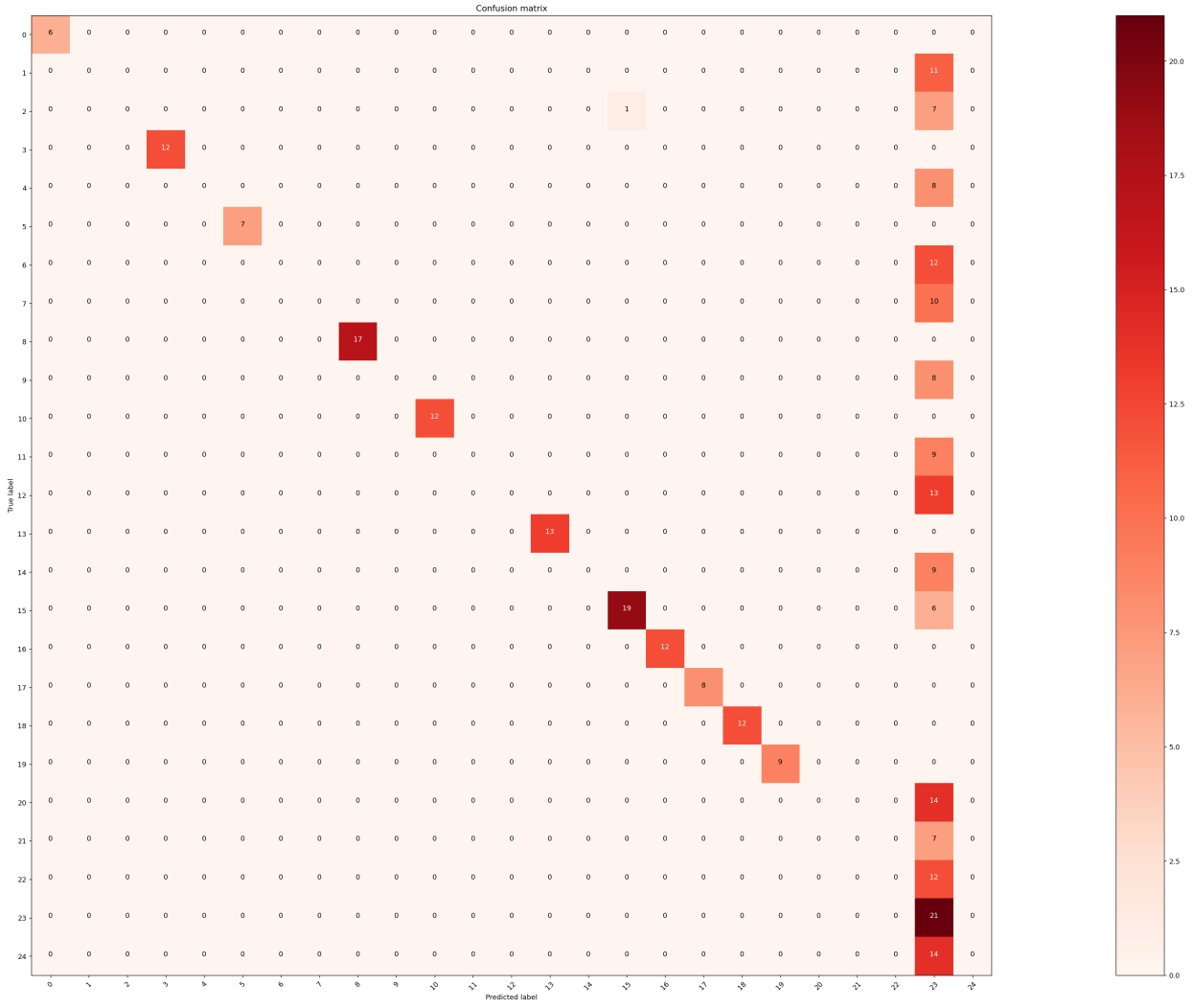
In [22]:
```
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                              cmap=plt.cm.Greens):
    plt.figure(figsize=(50, 20), dpi=130)
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]

    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
```

Using AdaBoost Classifier as the Model and printing evaluating it using confusion matrix

In [23]:
```
clf = AdaBoostClassifier(n_estimators=60)
clf = clf.fit(X_train, y_train)
yp = clf.predict(X_test)
acc = accuracy_score(y_test, yp)
print("accuracy is: ",acc)
CM = confusion_matrix(y_test, yp)
plot_confusion_matrix(CM, classes = range(25),cmap=plt.cm.Reds)
dump(clf, 'ada.joblib')
```

```
accuracy is:  0.5121107266435986
```

Out[23]:
```
['ada.joblib']
```

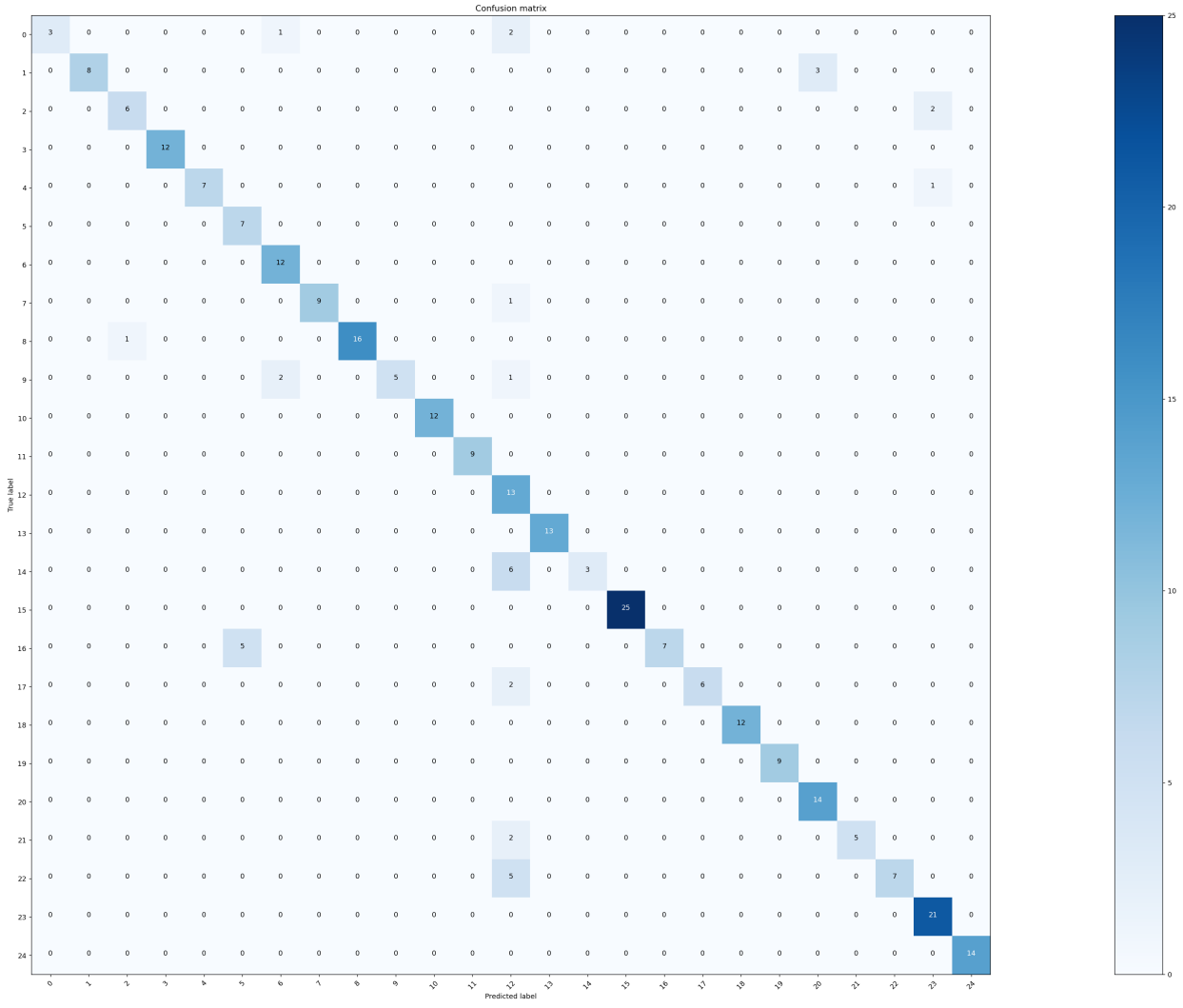


Using KNeighbors Classifier as the Model and printing evaluating it using confusion matrix

In [24]:
```
clf = KNeighborsClassifier(n_neighbors=7)
clf = clf.fit(X_train, y_train)
yp = clf.predict(X_test)
acc = accuracy_score(y_test, yp)
print("accuracy is: ",acc)
CM = confusion_matrix(y_test, yp)
plot_confusion_matrix(CM, classes = range(25),cmap=plt.cm.Blues)
dump(clf, 'knei.joblib')
```

```
accuracy is:  0.8823529411764706
```

Out[24]:
```
['knei.joblib']
```

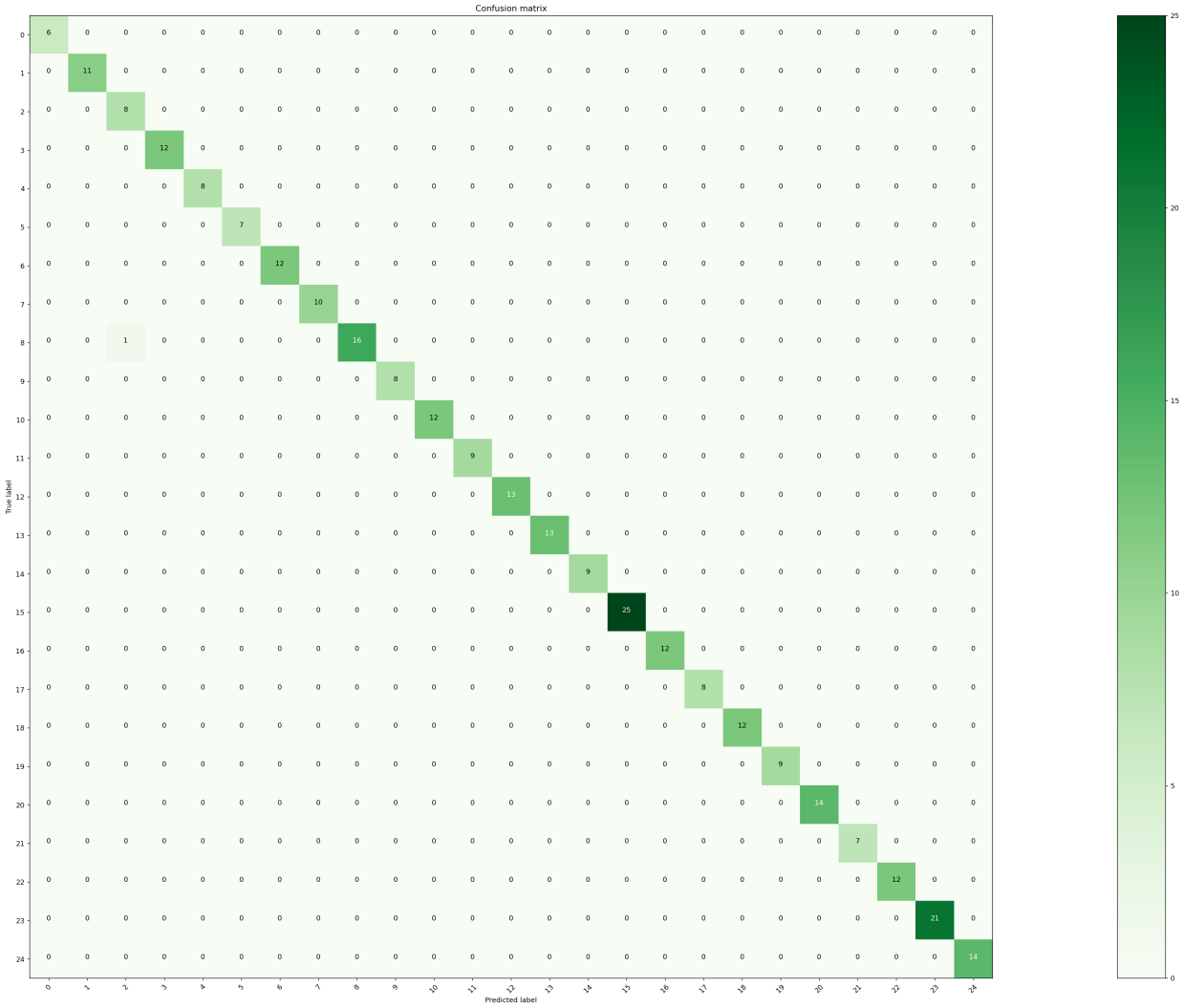


Using Decision tree as the Model and printing evaluating it using confusion matrix

In [25]:
```
clf = tree.DecisionTreeClassifier()
clf = clf.fit(X_train, y_train)
yp = clf.predict(X_test)
acc = accuracy_score(y_test, yp)
print("accuracy is: ",acc)
CM = confusion_matrix(y_test, yp)
plot_confusion_matrix(CM, classes = range(25))
dump(clf, 'DT.joblib')
```

```
accuracy is:  0.9965397923875432
```

Out[25]:
```
['DT.joblib']
```



Turning encoded lables into Categorical format

In [26]:
```
y_train=to_categorical(y_train, num_classes = 25, dtype='float32')
y_test=to_categorical(y_test, num_classes = 25, dtype='float32')
```

In [27]:
```
from keras.models import Sequential,Model
from tensorflow.keras.utils import plot_model
from keras.layers import Dense,LSTM, SpatialDropout1D, Embedding
from keras.layers import Dense, Embedding, GRU, LSTM, Dropout, Bidirectiona
from keras.layers import Conv1D, MaxPool1D, GlobalMaxPooling1D, GlobalAvera
```

Building a 1D CNN model

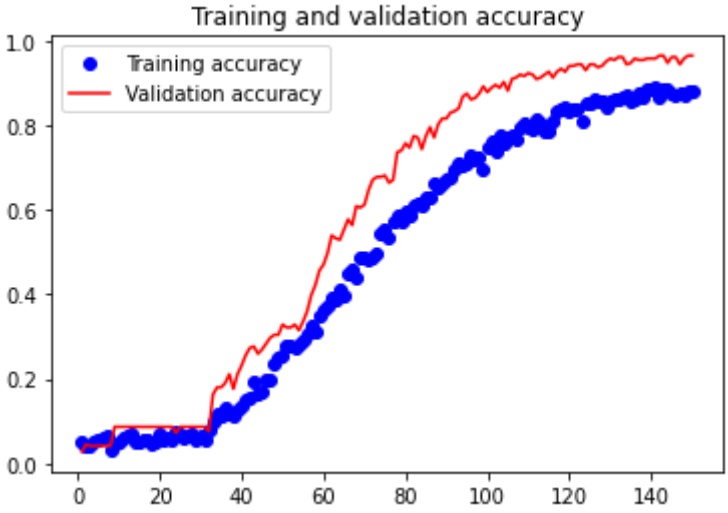
In [28]:
```
model = Sequential()
model.add(Embedding(input_dim=232337, output_dim=100, input_length=X_train.
model.add(Conv1D(128, 3, activation='relu'))
model.add(MaxPool1D(3))
model.add(Dropout(0.2))
model.add(Conv1D(128, 3, activation='relu'))
model.add(GlobalMaxPooling1D())
model.add(Dropout(0.2))
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(32, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(25, activation='softmax'))

model.compile(loss='binary_crossentropy', optimizer="adam", metrics=['accur
```

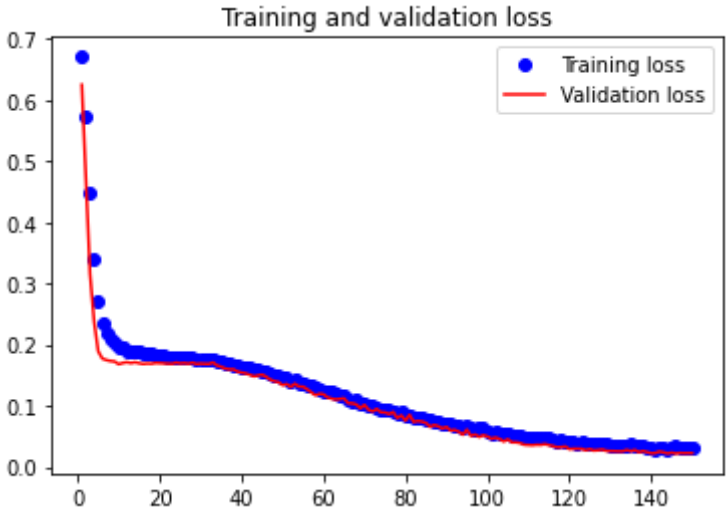
```
2022-10-07 21:50:28.136797: I tensorflow/stream_executor/cuda/cuda_gpu_ex
ecutor.cc:937] successful NUMA node read from SysFS had negative value (-
1), but there must be at least one NUMA node, so returning NUMA node zero
2022-10-07 21:50:28.235594: I tensorflow/stream_executor/cuda/cuda_gpu_ex
ecutor.cc:937] successful NUMA node read from SysFS had negative value (-
1), but there must be at least one NUMA node, so returning NUMA node zero
2022-10-07 21:50:28.236381: I tensorflow/stream_executor/cuda/cuda_gpu_ex
ecutor.cc:937] successful NUMA node read from SysFS had negative value (-
1), but there must be at least one NUMA node, so returning NUMA node zero
2022-10-07 21:50:28.237985: I tensorflow/core/platform/cpu_feature_guard.
cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Netwo
rk Library (oneDNN) to use the following CPU instructions in performance-
critical operations:  AVX2 AVX512F FMA
To enable them in other operations, rebuild TensorFlow with the appropria
te compiler flags.
2022-10-07 21:50:28.238320: I tensorflow/stream_executor/cuda/cuda_gpu_ex
ecutor.cc:937] successful NUMA node read from SysFS had negative value (-
1), but there must be at least one NUMA node, so returning NUMA node zero
2022-10-07 21:50:28.239149: I tensorflow/stream_executor/cuda/cuda_gpu_ex
ecutor.cc:937] successful NUMA node read from SysFS had negative value (-
1), but there must be at least one NUMA node, so returning NUMA node zero
2022-10-07 21:50:28.239849: I tensorflow/stream_executor/cuda/cuda_gpu_ex
ecutor.cc:937] successful NUMA node read from SysFS had negative value (-
1), but there must be at least one NUMA node, so returning NUMA node zero
2022-10-07 21:50:30.567415: I tensorflow/stream_executor/cuda/cuda_gpu_ex
ecutor.cc:937] successful NUMA node read from SysFS had negative value (-
1), but there must be at least one NUMA node, so returning NUMA node zero
2022-10-07 21:50:30.568344: I tensorflow/stream_executor/cuda/cuda_gpu_ex
ecutor.cc:937] successful NUMA node read from SysFS had negative value (-
1), but there must be at least one NUMA node, so returning NUMA node zero
2022-10-07 21:50:30.569033: I tensorflow/stream_executor/cuda/cuda_gpu_ex
ecutor.cc:937] successful NUMA node read from SysFS had negative value (-
1), but there must be at least one NUMA node, so returning NUMA node zero
2022-10-07 21:50:30.569631: I tensorflow/core/common_runtime/gpu/gpu_devi
ce.cc:1510] Created device /job:localhost/replica:0/task:0/device:GPU:0 w
ith 15401 MB memory:  -> device: 0, name: Tesla P100-PCIE-16GB, pci bus i
d: 0000:00:04.0, compute capability: 6.0
```

In [29]:
```
history = model.fit(X_train, y_train, batch_size = 64, epochs=150, validati
```

```
2022-10-07 21:50:31.455512: I tensorflow/compiler/mlir/mlir_graph_optimiz
ation_pass.cc:185] None of the MLIR Optimization Passes are enabled (regi
stered 2)

Epoch 1/150

2022-10-07 21:50:33.007257: I tensorflow/stream_executor/cuda/cuda_dnn.c
c:369] Loaded cuDNN version 8005
```

In [30]:
```
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(1, len(acc) + 1)
plt.plot(epochs, acc, 'bo', label='Training accuracy')
plt.plot(epochs, val_acc, 'r', label='Validation accuracy')
plt.title('Training and validation accuracy')
plt.legend()
plt.figure()
plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'r', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()
plt.show()
```





In [31]:
```
a= model.evaluate(X_test,y_test)
print("accuracy is: ",a[1])
```

```
10/10 [==============================] - 0s 6ms/step - loss: 0.0227 - acc
uracy: 0.9654
accuracy is:  0.9653978943824768
```

In [32]:
```
model.save('cnn.h5')
```