SENTIMENTAL ANALYISI

IPHONE 14 release

BANA 6620 FINAL PROJECT

Presented by:
SAI DURGESH MUKKAMALA
UDAY KIRAN GADDE
GAURAV SAI MALLARAM

OVERVIEW

- This data about the classification of analysis
- Analyzing the consumer sentiment towards the new iPhone serises in world markets
- We used different type python tools like Pandas ,Numpy ,Word cloud,Bigram Model, Build logistic Regression Model,Hyper para meter
- We taken the data from kaggle to create this machine learning model

REQUIRMENTS

FUNCTIONAL
NON FUNCTIONAL

- Data set loading
- Data set preprocessing
- Model Building
- Plotting graphs
- Safety and security
- Availability

Reading the Dataset and Analyzing the dataset

```
df = pd.read_csv('iphone14-query-tweets.csv')
```

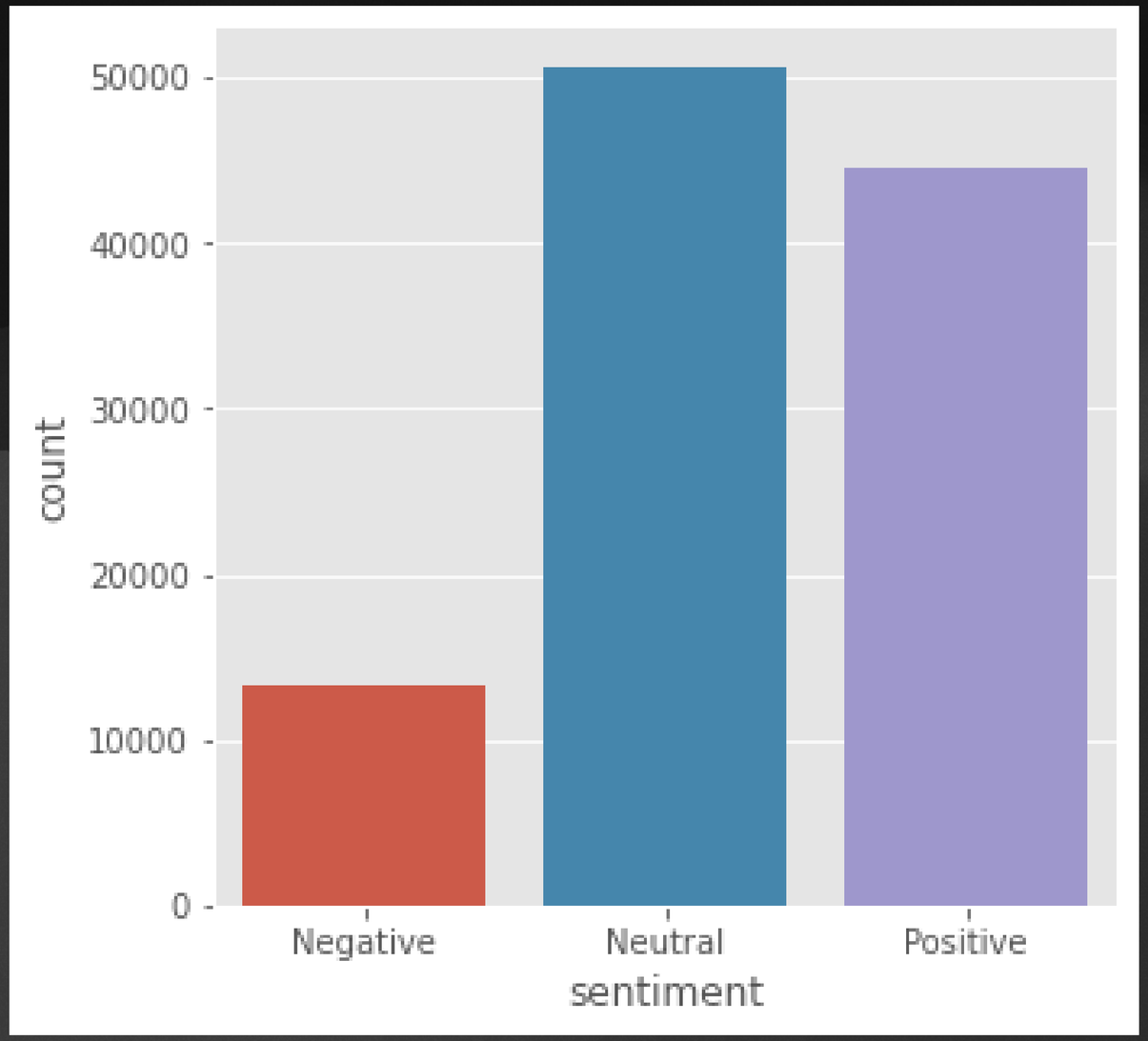
```
df.head()
```

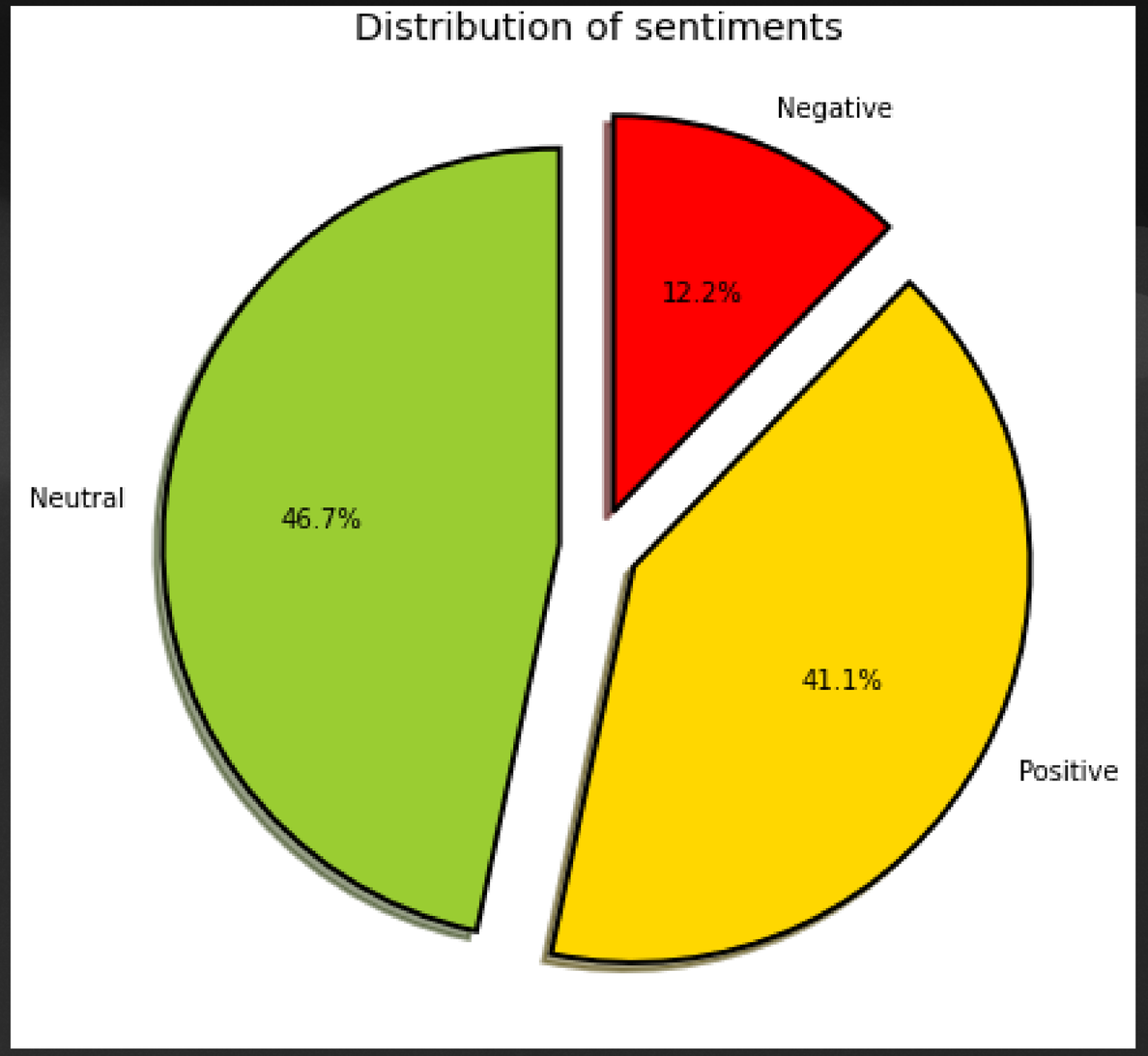
	date_time	username	user_location	user_description	verified	followers_count	following_count	tw
0	2022-09-08 22:49:29+00:00	TheAppleGang101	NaN	A new account dedicated to all of the latest l...	False	10	28	
1	2022-09-08 22:49:27+00:00	TheJessicats	1999	Tweet like nobody's reading • standup comedian...	False	1642	1444	
2	2022-09-08 22:49:16+00:00	itschefnotjeff	📌 mom's basement	👨🏽💳🪪🛂✈️	False	77	87	
3	2022-09-08 22:49:09+00:00	HalfRonin	Between the darkness and light	Preferring to be the dumbest person in the roo...	False	549	717	
4	2022-09-08 22:49:09+00:00	Deejayrayman	Texas	Father of 3. Follower of Christ. Lover of all ...	False	48	153	

Caption

GRAPH AND PIE CHART

Positive ,Negative and Neutral





Review

- The above graph and pie chart shows about the negative and postive reviews of new iPhone and as well as neutral reviews
- When compared with neutral reviews The negative reviews has about 12.2 %
- And the positive reviews has about 41.1% and neutral reviews has $46.7%

Bigram model

Create a Bigram Model

In [43]:
```
vect = CountVectorizer(ngram_range=(1,2)).fit(text_df['tweet_text'])
```

In [44]:
```
feature_names = vect.get_feature_names()
print("Number of features: {}\n".format(len(feature_names)))
print("First 20 features:\n {}".format(feature_names[:20]))
```

```
C:\Users\HP\anaconda3\lib\site-packages\sklearn\utils\deprecation.py:87: FutureWarning: Fun
ction get_feature_names is deprecated; get_feature_names is deprecated in 1.0 and will be r
emoved in 1.2. Please use get_feature_names_out instead.
  warnings.warn(msg, category=FutureWarning)
```

```
Number of features: 499743

First 20 features:
 ['00', '00 bottom', '00 iphone', '000', '000 6k', '000 appleevent', '000 entrylevel', '000
good', '000 help', '000 retweet', '000 travel', '000 tree', '000 usd', '000 weekend', '0000
', '0000 0200', '000000', '000000 units', '000000000010', '000001s']
```

In [45]:
```
X = text_df['tweet_text']
Y = text_df['sentiment']
X = vect.transform(X)
```

In [46]:
```
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
```

In [47]:
```
print("Size of x_train:", (x_train.shape))
print("Size of y_train:", (y_train.shape))
print("Size of x_test:", (x_test.shape))
print("Size of y_test:", (y_test.shape))
```

```
Size of x_train: (86830, 499743)
Size of y_train: (86830,)
Size of x_test: (21708, 499743)
Size of y_test: (21708,)
```

We use this model to find any sequence of words which is given in the tweets of data by using conditional probability of one preceding order

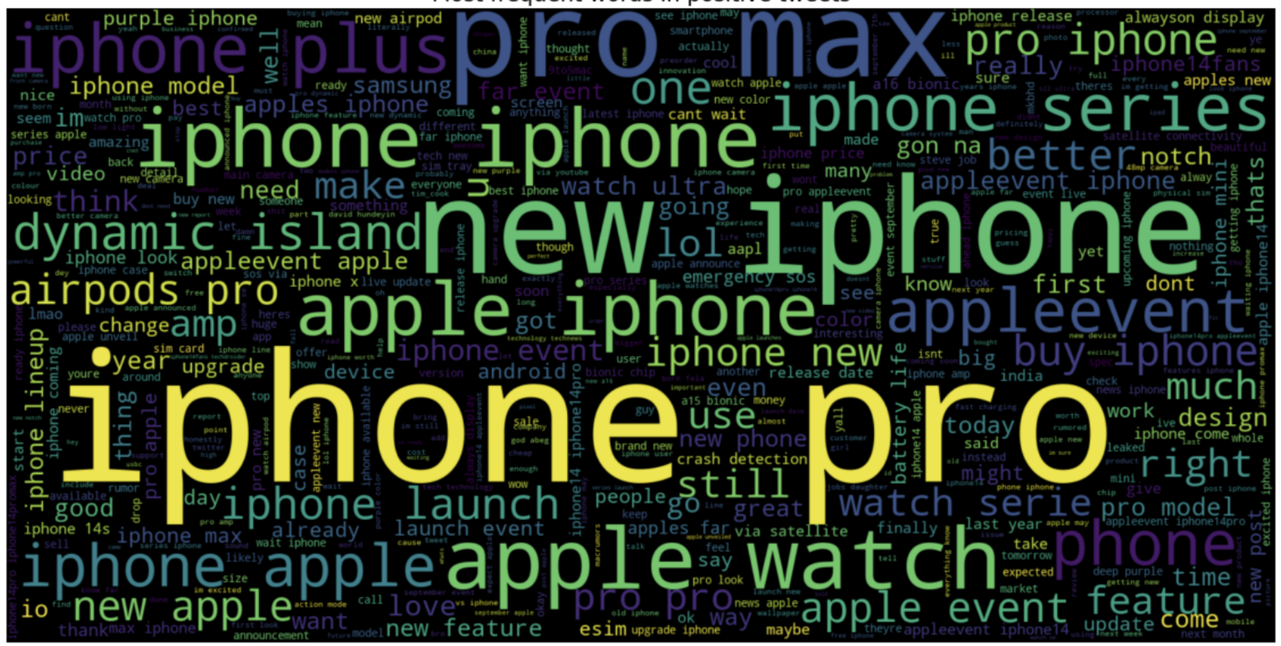
Wordcloud for the Positive Reviews

The images show that the word's which are used frequently in the positive reviews with the font size of 19 and figure size is like 20,15

In this the word cloud "iPhone pro" and "new iPhone" has frequently used by the consumers

Wordcloud for the positive tweets

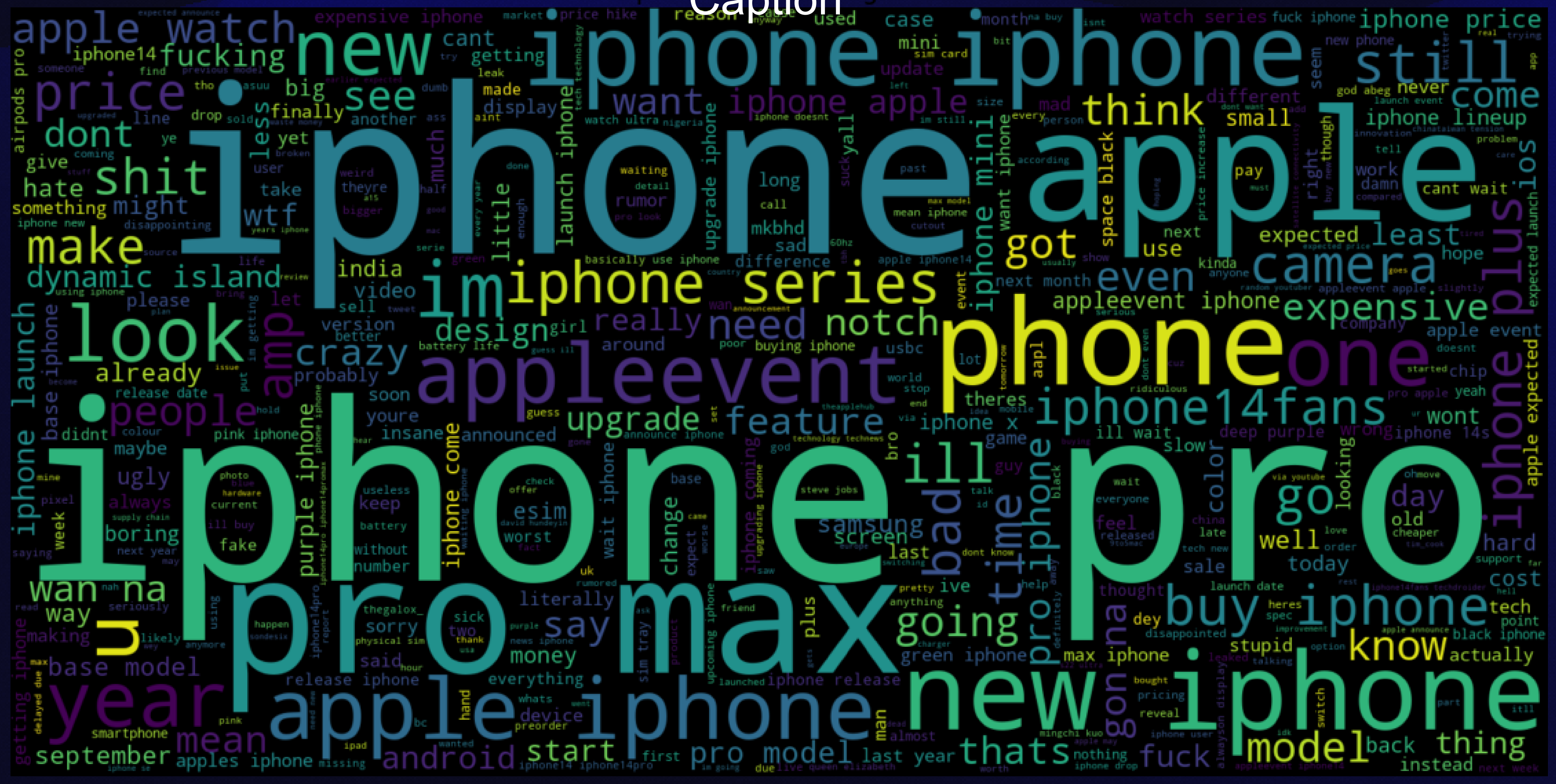
In [38]:
```
text = ' '.join([word for word in pos_tweets['tweet_text']])
plt.figure(figsize=(20,15), facecolor='None')
wordcloud = WordCloud(max_words=500, width=1600, height=800).generate(text)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title('Most frequent words in positive tweets', fontsize=19)
plt.show()
```



Wordcloud for the negative tweets

```
In [40]: text = ' '.join([word for word in neg_tweets['tweet_text']])
         plt.figure(figsize=(20,15), facecolor='None')
         wordcloud = WordCloud(max_words=500, width=1600, height=800).generate(text)
         plt.imshow(wordcloud, interpolation='bilinear')
         plt.axis("off")
         plt.title('Most frequent words in negative tweets', fontsize=19)
         plt.show()
```

Wordcloud for the Negative Reviews



Caption

The images show that the word's which are used frequently in the negative . reviews with the font size of 19 and figure size is like 20,15

In this the word cloud "iPhone pro" and "iPhone apple" has frequently used by the consumers

Wordcloud for the Neutral Reviews

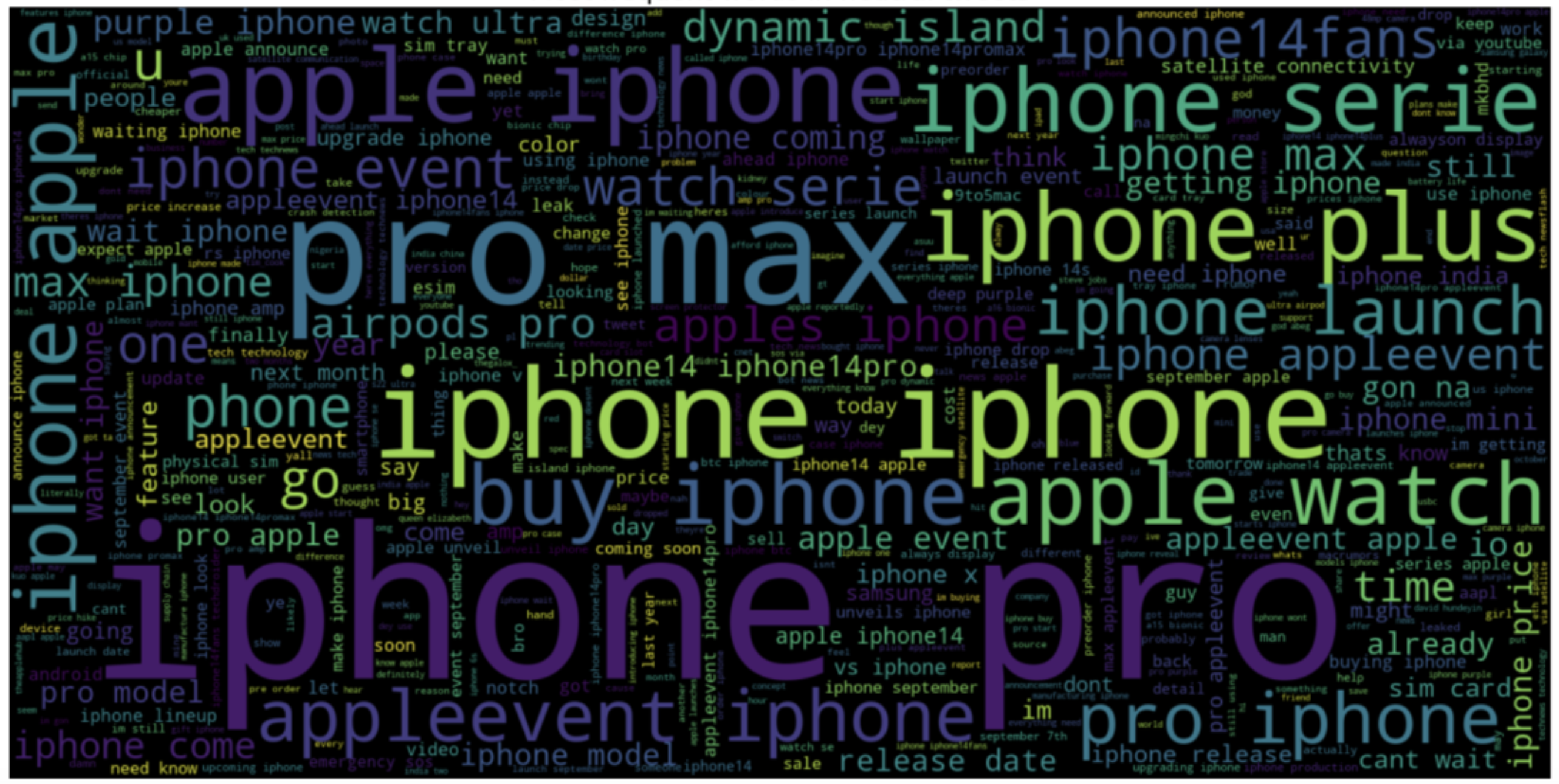
The images show that the word's which are used frequently in the neural reviews with the font size of 19 and figure size is like 20,15

In this the word cloud "iPhone pro" has frequently used by the consumers

Wordcloud for the neutral tweets

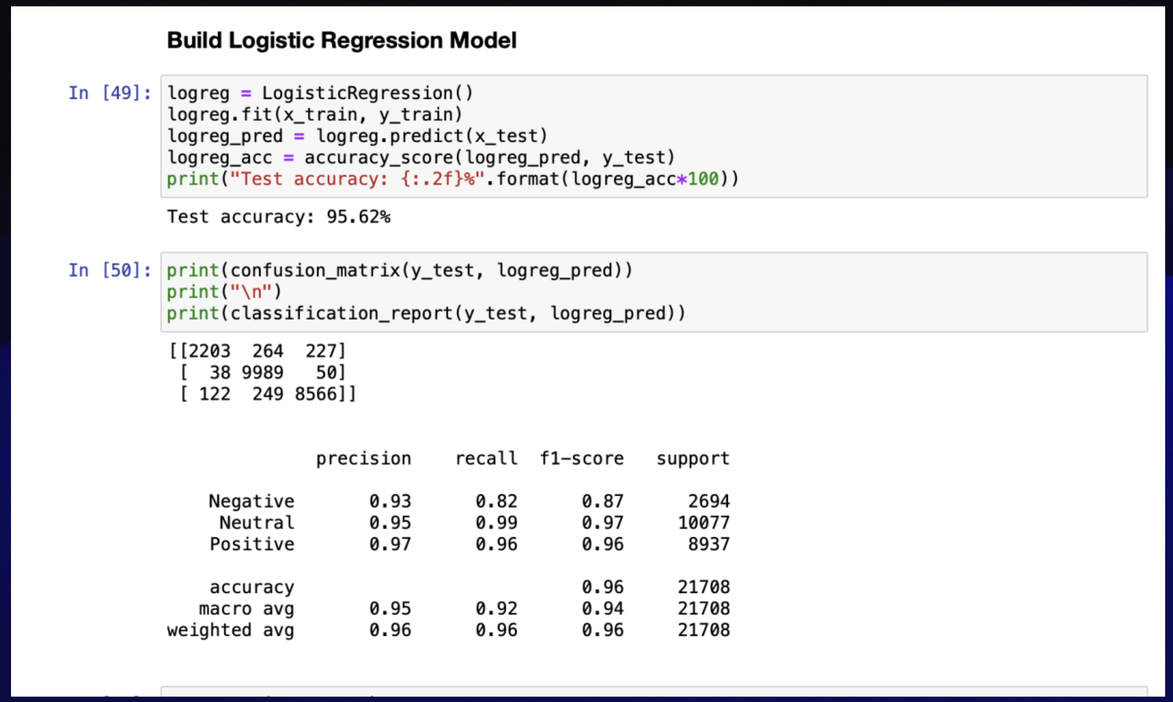
```
In [42]: text = ' '.join([word for word in neutral_tweets['tweet_text']])
         plt.figure(figsize=(20,15), facecolor='None')
         wordcloud = WordCloud(max_words=500, width=1600, height=800).generate(text)
         plt.imshow(wordcloud, interpolation='bilinear')
         plt.axis("off")
         plt.title('Most frequent words in neutral tweets', fontsize=19)
         plt.show()
```

Most frequent words in neutral tweets



Build Logistic Regression Model

In this model we categorized The outcomes like precision ,recall, f1 score and support of new iPhone series tweets
And also we accurate the test value



Build Logistic Regression Model

In [49]:
```
logreg = LogisticRegression()
logreg.fit(x_train, y_train)
logreg_pred = logreg.predict(x_test)
logreg_acc = accuracy_score(logreg_pred, y_test)
print("Test accuracy: {:.2f}%".format(logreg_acc*100))
```

```
Test accuracy: 95.62%
```

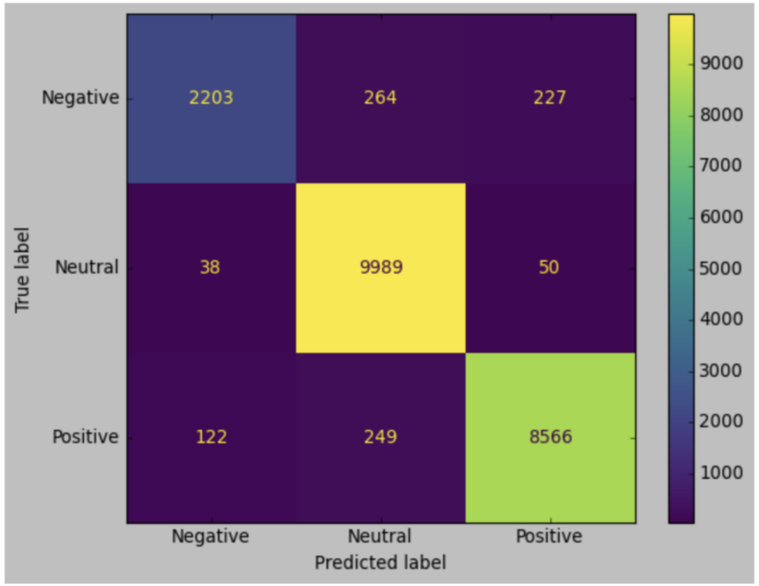
In [50]:
```
print(confusion_matrix(y_test, logreg_pred))
print("\n")
print(classification_report(y_test, logreg_pred))
```

```
[[2203  264  227]
 [  38 9989   50]
 [ 122  249 8566]]


              precision    recall  f1-score   support

    Negative       0.93      0.82      0.87      2694
     Neutral       0.95      0.99      0.97     10077
    Positive       0.97      0.96      0.96      8937

    accuracy                           0.96     21708
   macro avg       0.95      0.92      0.94     21708
weighted avg       0.96      0.96      0.96     21708
```

Hyper parameter

Hyperparameter tuning consists of finding a set of optimal hyperparameter values for a learning algorithm while apply ing the optimized algorithm

```
In [51]: style.use('classic')
         cm = confusion_matrix(y_test, logreg_pred, labels=logreg.classes_)
         disp = ConfusionMatrixDisplay(confusion_matrix = cm, display_labels=logreg.classes_)
         disp.plot()

Out[51]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x1978948a670>
```



CONCLUSION

In proposed system, a model is built to generate sentiments for iPhone 14 tweets by the customers. The model consists of four parts: data loading from kaggle, data preprocessing, generating sentiments, model training, computing accuracy and confusion matrix.

This application will predict the sentiments for the iPhone 14 tweets which are posted by the customer on twitter. Sentiments can be classified into positive, negative or neutral. Firstly, we have performed data preprocessing steps in order to refine the data. Secondly, we have generated the sentiments for the tweets. Then we have built the model using logistic regression and vector machine. We have also computed the respective accuracies and confusion matrix.

Accuracy obtained for logistic regression is 95.62%. On tuning it with GridSearchCV, the accuracy obtained is 95.97%. Accuracy obtained for SVM is 96.84%. On tuning it with GridSearchCV, the accuracy obtained is 95.97%.

THANK YOU