In [ ]:

```
Name: Ishan Chaskar
URK21CS1181
```

In [ ]:

```
Aim:
To perform performance analysis K-Means Clustering technique on loan.csv
dataset
```

In [ ]:

```
Description:
K-means clustering algorithm computes the centroids and iterates until we
find the optimal centroid. The number of clusters identified from data by
the algorithm is represented by ‘K’ in K-means. In this algorithm, the data
points are assigned to a cluster in such a manner that the sum of the
squared distance between the data points and the centroid would be minimum.
It is to be understood that less variation within the clusters will lead to
more similar data points within same cluster.

Step 1: First, we need to specify the number of clusters, K, that need to
be generatedby this algorithm.

Step 2: Next, randomly select K data points and assign each data point to a
cluster. In simple words, classify the data based on the number of data
points.

Step 3: Now it will compute the cluster centroids.

Step4: Next, keep iterating the following until we find the optimal centroid
which is the asignment of data points to the clusters that are not changing
any more
entroid=N1 i=1Nxi

Where: Centroid Centroid is the centroid of the cluster.
N is the number of data points in the cluster. xi represents the individual
data points in the cluster.
Silhouette Score=1/N i=1Ns(i)

Where: coreSilhouette Score is the average silhouette score for the dataset.
N is he total number of data points. s(i) is the silhouette score for
data point i DBI=1/K( i=1K maxj!=iRij)

Where: DBI is the Davies-Bouldin Index. K is
the number of clusters. Rij is the similarity index between clusters ii and
jj.
```

In [4]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score, davies_bouldin_score
df = pd.read_csv('Loan.csv')
df
```

Out[4]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Co
0	LP001003	Male	Yes	1	Graduate	No	4583	
1	LP001005	Male	Yes	0	Graduate	Yes	3000	
2	LP001006	Male	Yes	0	Not Graduate	No	2583	
3	LP001008	Male	No	0	Graduate	No	6000	
4	LP001013	Male	Yes	0	Not Graduate	No	2333	
...	...	...	...	...	...	...	...	
376	LP002953	Male	Yes	3+	Graduate	No	5703	
377	LP002974	Male	Yes	0	Graduate	No	3232	
378	LP002978	Female	No	0	Graduate	No	2900	
379	LP002979	Male	Yes	3+	Graduate	No	4106	
380	LP002990	Female	No	0	Graduate	Yes	4583	

381 rows × 13 columns

In [ ]:

```
1. Develop a K-means clustering model for the Loan dataset using the
scikit-learn
a. Use the columns: 'ApplicantIncome', 'LoanAmount' as the input variables.
```

In [6]:

```
print('URK21CS1181')
df2=df.loc[:,['ApplicantIncome','LoanAmount']]
df2
```

```
URK21CS1181
```

Out[6]:

	ApplicantIncome	LoanAmount
0	4583	128
1	3000	66
2	2583	120
3	6000	141
4	2333	95
...	...	...
376	5703	128
377	3232	108
378	2900	71
379	4106	40
380	4583	133

381 rows × 2 columns

In [ ]:

```
b. Compute the optimal number of cluster ‘K’ from 1-10 using the Elbow method
```

In [7]:

```
print('URK21CS1181')
wcss=[]
for i in range(1,11):
    kmeans=KMeans(n_clusters=i,init='k-means++',random_state=4)
    kmeans.fit_transform(df2)
    wcss.append(kmeans.inertia_)
```

```
URK21CS1181
```

```
/home/urk21cs1181/.local/lib/python3.9/site-packages/sklearn/cluster/_kmeans.p
y:1412: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/urk21cs1181/.local/lib/python3.9/site-packages/sklearn/cluster/_kmeans.p
y:1412: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/urk21cs1181/.local/lib/python3.9/site-packages/sklearn/cluster/_kmeans.p
y:1412: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/urk21cs1181/.local/lib/python3.9/site-packages/sklearn/cluster/_kmeans.p
y:1412: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/urk21cs1181/.local/lib/python3.9/site-packages/sklearn/cluster/_kmeans.p
y:1412: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/urk21cs1181/.local/lib/python3.9/site-packages/sklearn/cluster/_kmeans.p
y:1412: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/urk21cs1181/.local/lib/python3.9/site-packages/sklearn/cluster/_kmeans.p
y:1412: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/urk21cs1181/.local/lib/python3.9/site-packages/sklearn/cluster/_kmeans.p
y:1412: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/urk21cs1181/.local/lib/python3.9/site-packages/sklearn/cluster/_kmeans.p
y:1412: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
/home/urk21cs1181/.local/lib/python3.9/site-packages/sklearn/cluster/_kmeans.p
y:1412: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```

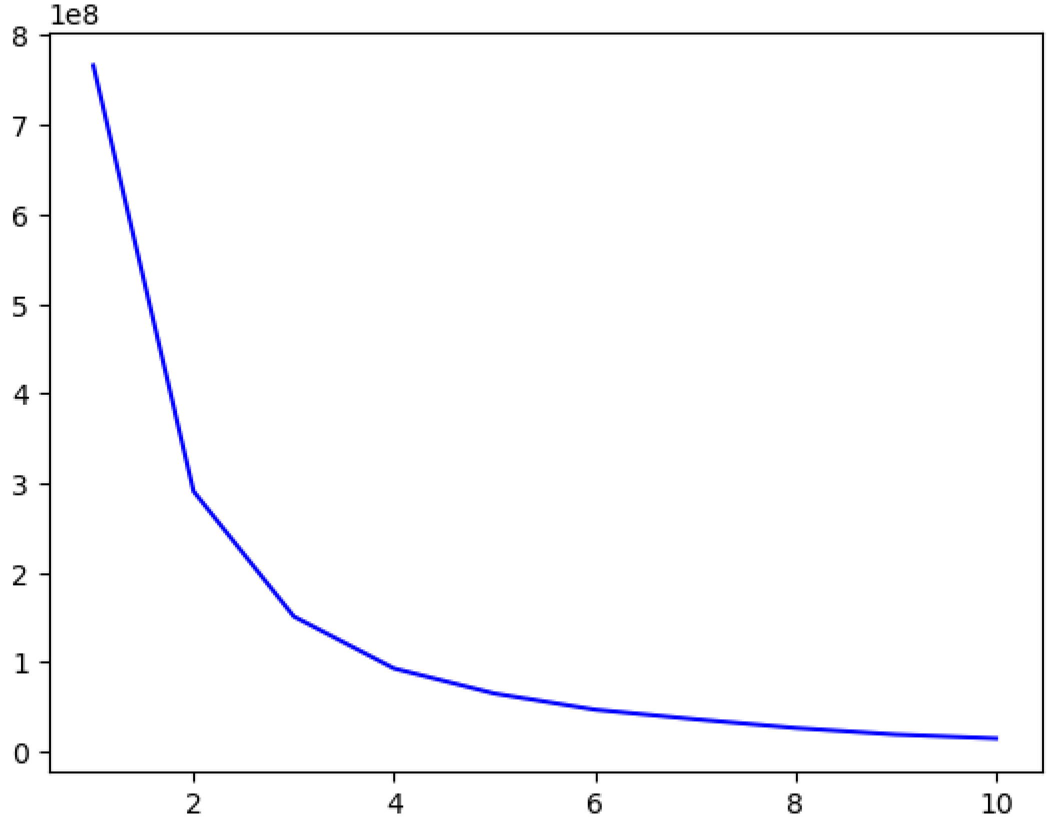
In [ ]:

```
c. Plot the graph between number of cluster K and within-cluster sum of squares
value.
```

In [8]:

```
print("URK21CS1181")
print(wcss)
plt.plot(range(1,11),wcss,c='b')
plt.show()
```

```
URK21CS1181
[766336682.7979002, 291148680.6268268, 151307581.0588933, 93198687.75918044, 65
244928.260725856, 47146700.2008873, 36453562.21664511, 26710430.509071264, 1941
6359.90912394, 15052501.537626004]
```



In [ ]:

```
d. Perform the K-means clustering with the selected optimal K.
```

In [9]:

```
print('URK21CS1181')
km=KMeans(n_clusters=3,init='k-means++',random_state=0)
y_predict=km.fit_predict(df2)
km.cluster_centers_
```

```
URK21CS1181
/home/urk21cs1181/.local/lib/python3.9/site-packages/sklearn/cluster/_kmeans.p
y:1412: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```

Out[9]:

```
array([[2506.41304348,   97.58695652],
       [4043.45454545,  109.77922078],
       [6512.76744186,  119.48837209]])
```

In [ ]:

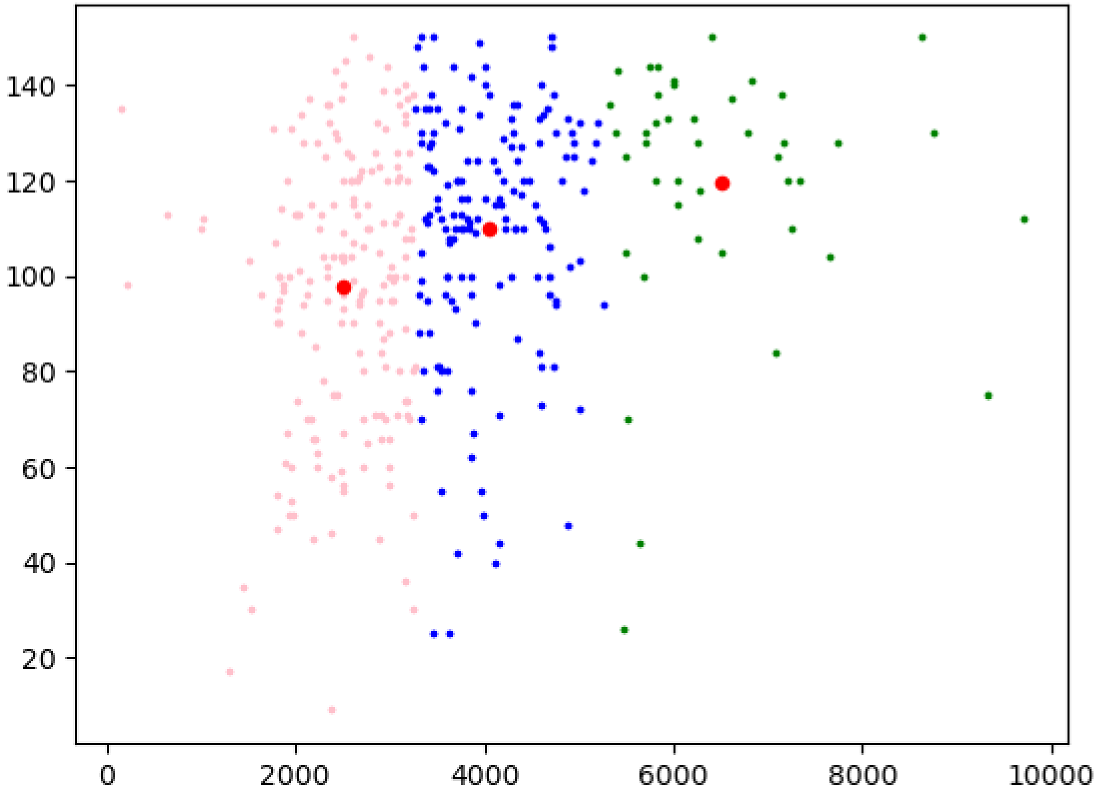
```
e. Display the cluster centroids.
f. Visualize the data representation of K-means clustering.
g. Change the value of K in K-means with different values and tabulate the
   performance metrics such as silhouette_score and davies_bouldin_score
   obtained.
```

In [10]:

```
 print('URK21CS1181')
plt.scatter(df2.iloc[:, 0][y_predict == 0], df2.iloc[:, 1][y_predict == 0],
color='pink', s=3)

plt.scatter(df2.iloc[:, 0][y_predict == 1], df2.iloc[:, 1][y_predict == 1],
color='blue', s=3)
```

```
plt.scatter(df2.iloc[:, 0][y_predict == 2], df2.iloc[:, 1][y_predict == 2],
color='green', s=3)

plt.scatter(km.cluster_centers_[0][0], km.cluster_centers_[0][1], c='r', s=20)
plt.scatter(km.cluster_centers_[1][0], km.cluster_centers_[1][1], c='r', s=20)
plt.scatter(km.cluster_centers_[2][0], km.cluster_centers_[2][1], c='r', s=20)

print("For 3 Clusters:")
print("silhouette_score: ", silhouette_score(df2, km.labels_))
print("davies_bouldin_score: ", davies_bouldin_score(df2, km.labels_))
```

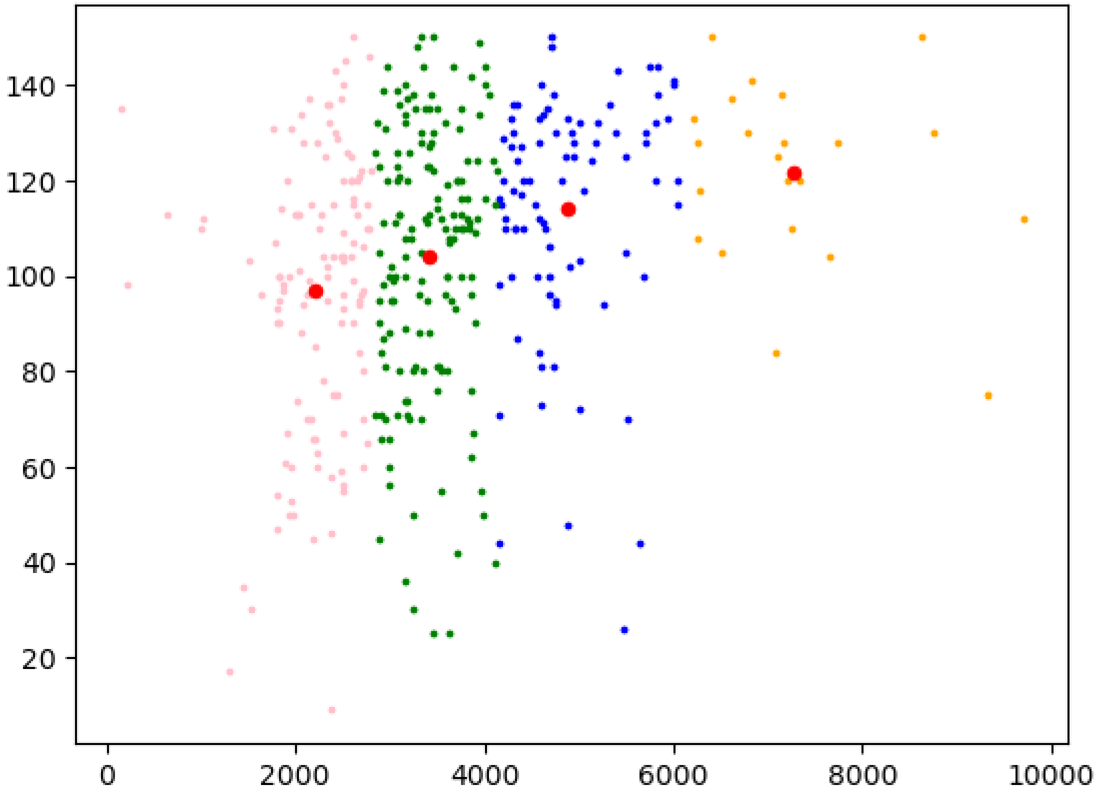
```
URK21CS1181
For 3 Clusters:
silhouette_score:  0.5415113631813214
davies_bouldin_score:  0.5695527663558264
```



In [11]:

```
print('URK21CS1181')
k = 4
km = KMeans(n_clusters=k)
y_predict = km.fit_predict(df2)
plt.scatter(df2.iloc[:, 0][y_predict == 0], df2.iloc[:, 1][y_predict == 0],
color='pink', s=3)

plt.scatter(df2.iloc[:, 0][y_predict == 1], df2.iloc[:, 1][y_predict == 1],
color='blue', s=3)

plt.scatter(df2.iloc[:, 0][y_predict == 2], df2.iloc[:, 1][y_predict == 2],
color='green', s=3)

plt.scatter(df2.iloc[:, 0][y_predict == 3], df2.iloc[:, 1][y_predict == 3],
color='orange', s=3)

plt.scatter(km.cluster_centers_[:, 0], km.cluster_centers_[:, 1], c='r', s=20)

print("For 4 Clusters:")
print("silhouette_score: ", silhouette_score(df2, km.labels_))
print("davies_bouldin_score: ", davies_bouldin_score(df2, km.labels_))
```

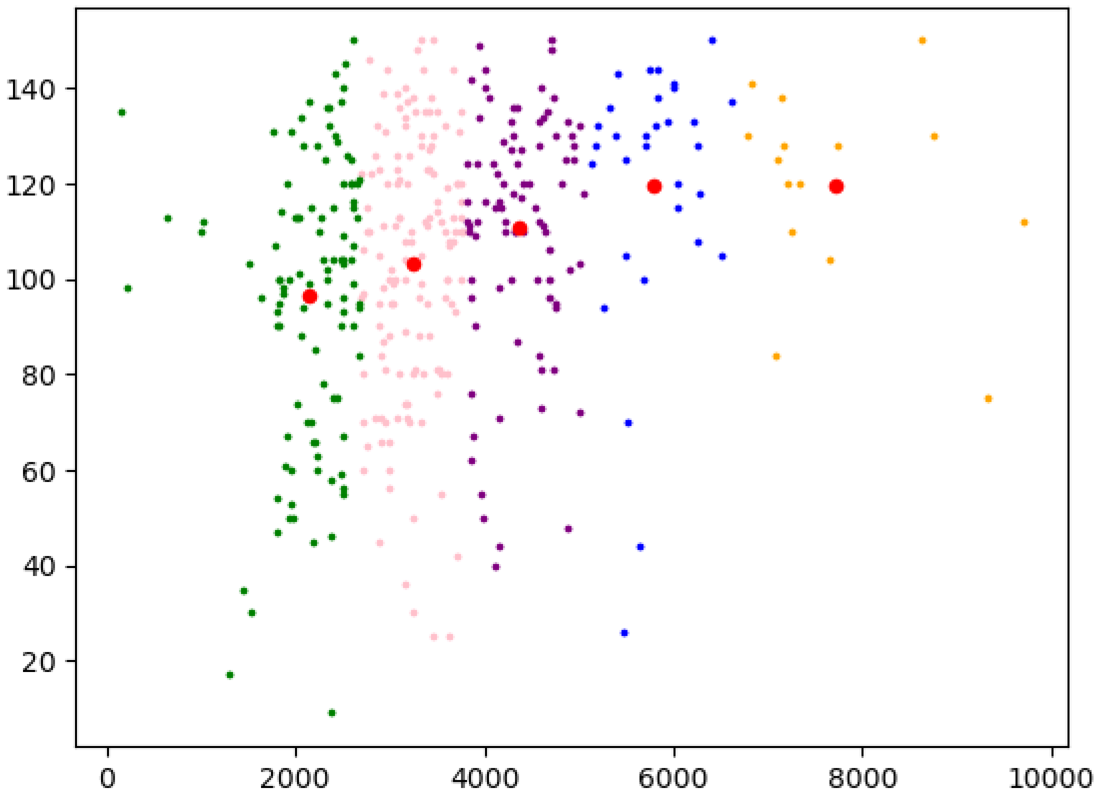
```
URK21CS1181
/home/urk21cs1181/.local/lib/python3.9/site-packages/sklearn/cluster/_kmeans.p
y:1412: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
For 4 Clusters:
silhouette_score:  0.5418116085741493
davies_bouldin_score:  0.5312993347397801
```



In [12]:
```
print('URK21CS1181')
k = 5
km = KMeans(n_clusters=k)
y_predict = km.fit_predict(df2)
plt.scatter(df2.iloc[:, 0][y_predict == 0], df2.iloc[:, 1][y_predict == 0],
color='pink', s=3)
plt.scatter(df2.iloc[:, 0][y_predict == 1], df2.iloc[:, 1][y_predict == 1],
color='blue', s=3)
plt.scatter(df2.iloc[:, 0][y_predict == 2], df2.iloc[:, 1][y_predict == 2],
color='green', s=3)
plt.scatter(df2.iloc[:, 0][y_predict == 3], df2.iloc[:, 1][y_predict == 3],
color='orange', s=3)
plt.scatter(df2.iloc[:, 0][y_predict == 4], df2.iloc[:, 1][y_predict == 4],
color='purple', s=3)
plt.scatter(km.cluster_centers_[:, 0], km.cluster_centers_[:, 1], c='r', s=20)
print("For 5 Clusters:")
print("silhouette_score: ", silhouette_score(df2, km.labels_))
print("davies_bouldin_score: ", davies_bouldin_score(df2, km.labels_))
```

```
URK21CS1181
/home/urk21cs1181/.local/lib/python3.9/site-packages/sklearn/cluster/_kmeans.p
y:1412: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```

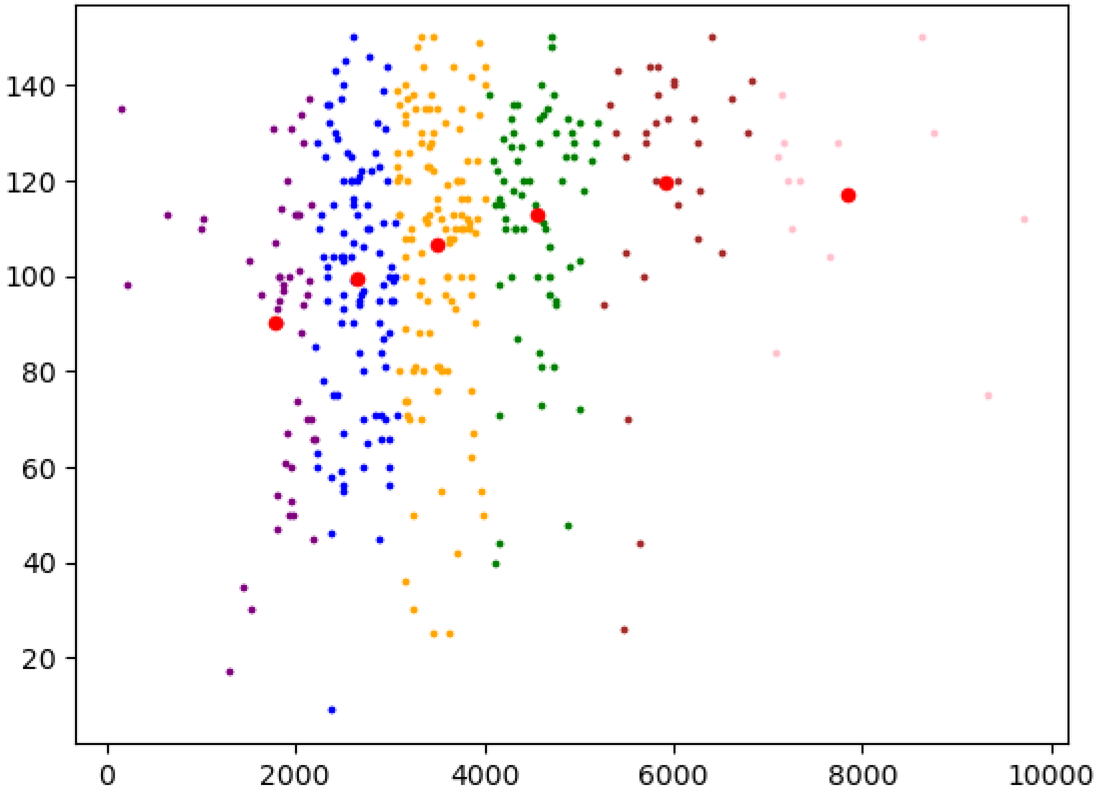
```
For 5 Clusters:
silhouette_score:  0.5341427351221556
davies_bouldin_score:  0.5475355494628509
```



In [16]:

```python
print('URK21CS1181')
k = 6
km = KMeans(n_clusters=k)
y_predict = km.fit_predict(df2)
plt.scatter(df2.iloc[:, 0][y_predict == 0], df2.iloc[:, 1][y_predict == 0], 
color='pink', s=3)
plt.scatter(df2.iloc[:, 0][y_predict == 1], df2.iloc[:, 1][y_predict == 1], 
color='blue', s=3)
plt.scatter(df2.iloc[:, 0][y_predict == 2], df2.iloc[:, 1][y_predict == 2], 
color='green', s=3)
plt.scatter(df2.iloc[:, 0][y_predict == 3], df2.iloc[:, 1][y_predict == 3], 
color='orange', s=3)
plt.scatter(df2.iloc[:, 0][y_predict == 4], df2.iloc[:, 1][y_predict == 4], 
color='purple', s=3)
plt.scatter(df2.iloc[:, 0][y_predict == 5], df2.iloc[:, 1][y_predict == 5], 
color='brown', s=3)
plt.scatter(km.cluster_centers_[:, 0], km.cluster_centers_[:, 1], c='r', s=20)
print("For 6 Clusters:")
print("silhouette_score: ", silhouette_score(df2, km.labels_))
print("davies_bouldin_score: ", davies_bouldin_score(df2, km.labels_))
```

```
URK21CS1181
/home/urk21cs1181/.local/lib/python3.9/site-packages/sklearn/cluster/_kmeans.p
y:1412: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
For 6 Clusters:
silhouette_score:  0.5225239222596462
davies_bouldin_score:  0.5588992782206206
```



In [ ]:

```
Result: Hence the python code to create KMeans cluster model for Loan dataset ha
        been coded and executed successfully.
```