Visualization

```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
match= pd.read_csv("/content/drive/MyDrive/4th sem Christ/ Machine Learning/matches.csv")
ball= pd.read_csv("/content/drive/MyDrive/4th sem Christ/ Machine Learning/Ball_by_Ball.csv")

match.head()
```

	id	season	city	date	team1	team2	toss_winner	toss_decision	resu
0	1	2008	Bangalore	2008-04-18	Kolkata Knight Riders	Royal Challengers Bangalore	Royal Challengers Bangalore	field	norm
1	2	2008	Chandigarh	2008-04-19	Chennai Super Kings	Kings XI Punjab	Chennai Super Kings	bat	norm
2	3	2008	Delhi	2008-04-19	Rajasthan Royals	Delhi Daredevils	Rajasthan Royals	bat	norm
3	4	2008	Mumbai	2008-04-20	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	norm
4	5	2008	Kolkata	2008-04-20	Deccan Chargers	Kolkata Knight Riders	Deccan Chargers	bat	norm

```
ball
```

	Match_Id	Innings_Id	Over_Id	Ball_Id	Team_Batting_Id	Team_Bowling_Id	Strike
0	335987	1	1	1	1	2	
1	335987	1	1	2	1	2	
2	335987	1	1	3	1	2	
3	335987	1	1	4	1	2	
4	335987	1	1	5	1	2	
...	...	...	...	...	...	...	

```
match.isnull()
```

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_ap
0	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	...	
572	False	False	False	False	False	False	False	False	False	
573	False	False	False	False	False	False	False	False	False	
574	False	False	False	False	False	False	False	False	False	
575	False	False	False	False	False	False	False	False	False	
576	False	False	False	False	False	False	False	False	False	

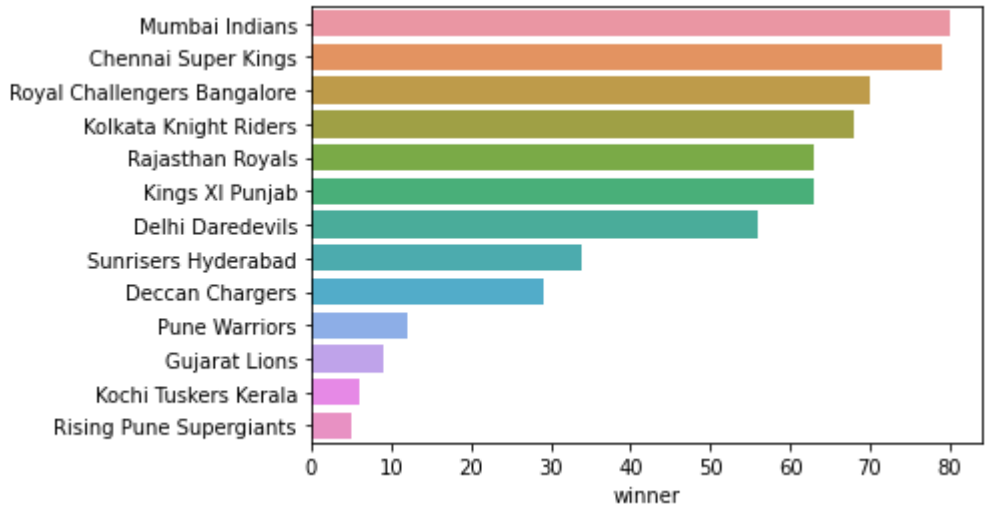
577 rows × 18 columns

```
match.columns
```

```
Index(['id', 'season', 'city', 'date', 'team1', 'team2', 'toss_winner',
       'toss_decision', 'result', 'dl_applied', 'winner', 'win_by_runs',
       'win_by_wickets', 'player_of_match', 'venue', 'umpire1', 'umpire2',
       'umpire3'],
      dtype='object')
```

Double-click (or enter) to edit

```
#Most Successful IPL Teams
data = match.winner.value_counts()
sns.barplot(y = data.index, x = data, orient='h');
```



A barplot is basically used to aggregate the categorical data according to some methods and by default it's the mean. It can also be understood as a visualization of the group by action. To use this plot we choose a categorical column for the x-axis and a numerical column for the y-axis, and we see that it creates a plot taking a mean per categorical column.

```
labels = ['KKR','SH','KXP','RC','DC','DD','CSK','KT','MI','RPS','RR']
sizes = [14, 4, 9, 21, 5, 14 , 20 ,2 ,18 ,1 , 17]
colors = ['#EEE5DE','#FFC1C1','#FF8247','#FF3030','#FFEC8B','#9AC9CB','#FFFFE0','#C0FF3E','#0
explode = (0, 0, 0, 0.25, 0, 0, 0, 0, 0, 0, 0)  # only "explode" the 7th slice (i.e. 'RC')
plt.pie(sizes, explode=explode, labels=labels, colors=colors,
        autopct='%1.1f%%', shadow=True, startangle=90,radius=1.5)
plt.title("Winning percentage of teams by large margin(more than 50 runs or by more than 8 wi
```

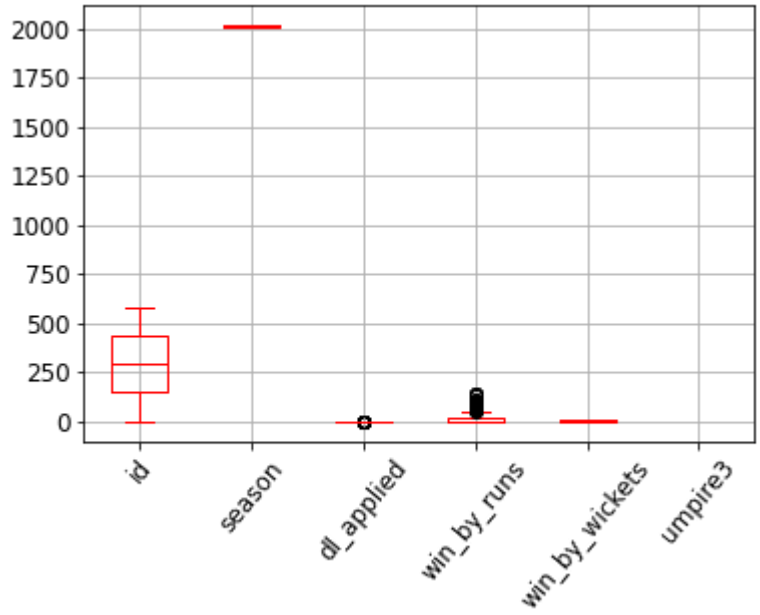
```
Text(0, -0.3, 'Winning percentage of teams by large margin(more than 50 runs or by more
```

Double-click (or enter) to edit


```
plt.figure(figsize=(20,10))
plt.title("Season Champions",fontweight='bold',fontsize=20)
plt.xlabel('Teams',fontweight='bold',fontsize=30)
plt.ylabel('Total Seasons',fontweight='bold',fontsize=20)
plt.xticks(rotation='60')
plt.tick_params(labelsize=10)
sns.countplot(x=match['winner'],palette=['#F535AA','#BCC6CC','yellow','#461B7E','blue','#F872
```

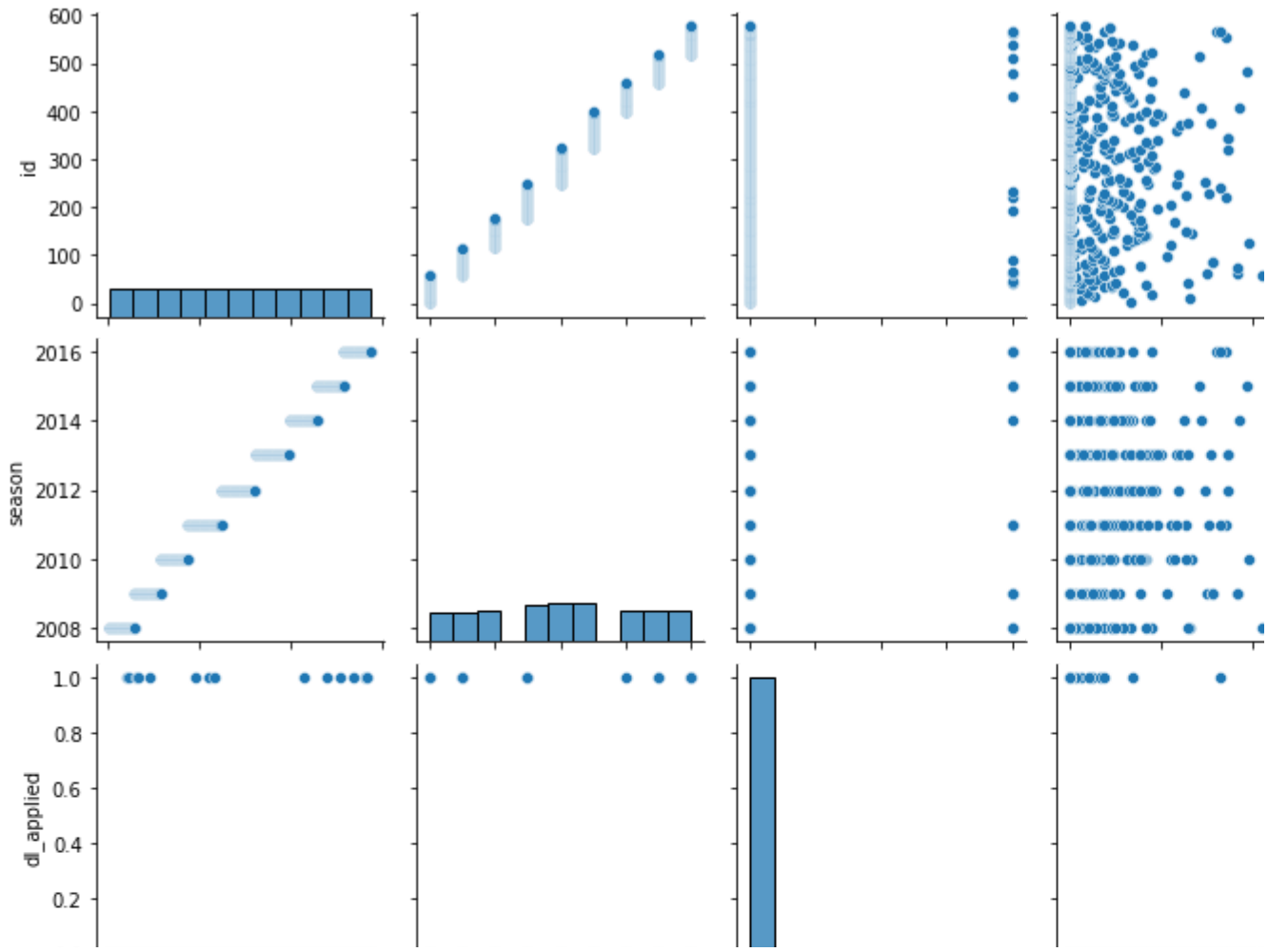
```
match.boxplot(grid='false', color='red',fontsize=12, rot=50 )
```

```
/usr/local/lib/python3.7/dist-packages/matplotlib/cbook/__init__.py:1376: VisibleDeprecа
  X = np.atleast_1d(X.T if isinstance(X, np.ndarray) else np.asarray(X))
<matplotlib.axes._subplots.AxesSubplot at 0x7fad5e4717d0>
```



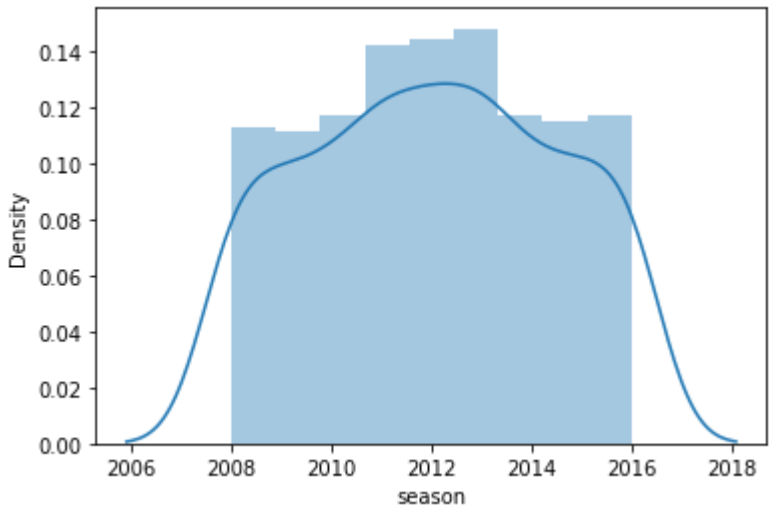
```
sns.pairplot(data=match)
```

```
<seaborn.axisgrid.PairGrid at 0x7fad65b98b50>
```



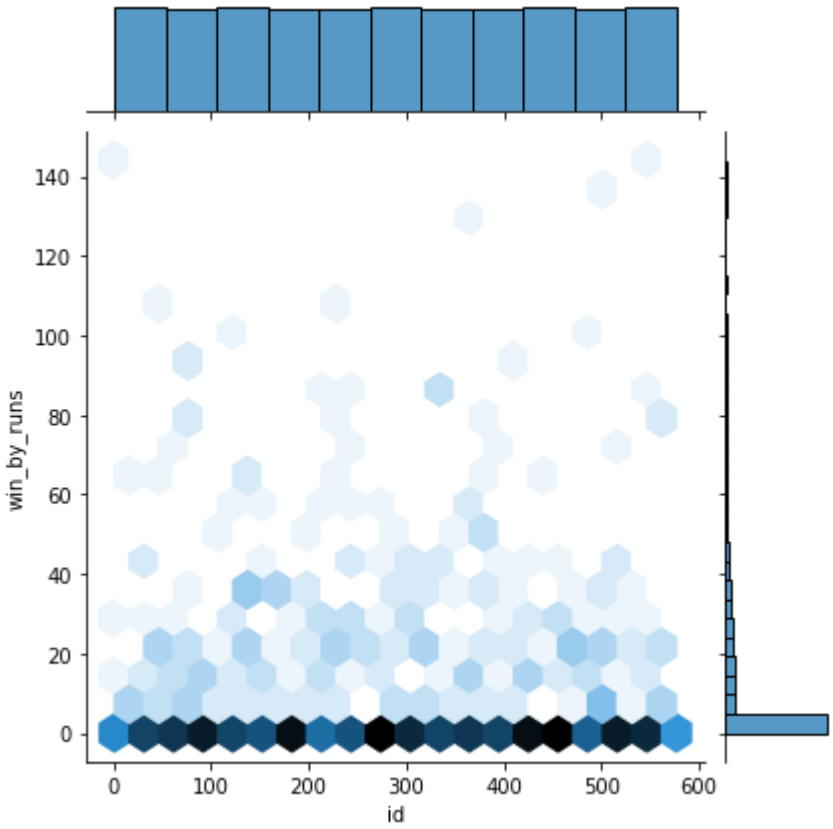

```
sns.distplot(match.season)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `di
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7fad623c70d0>
```



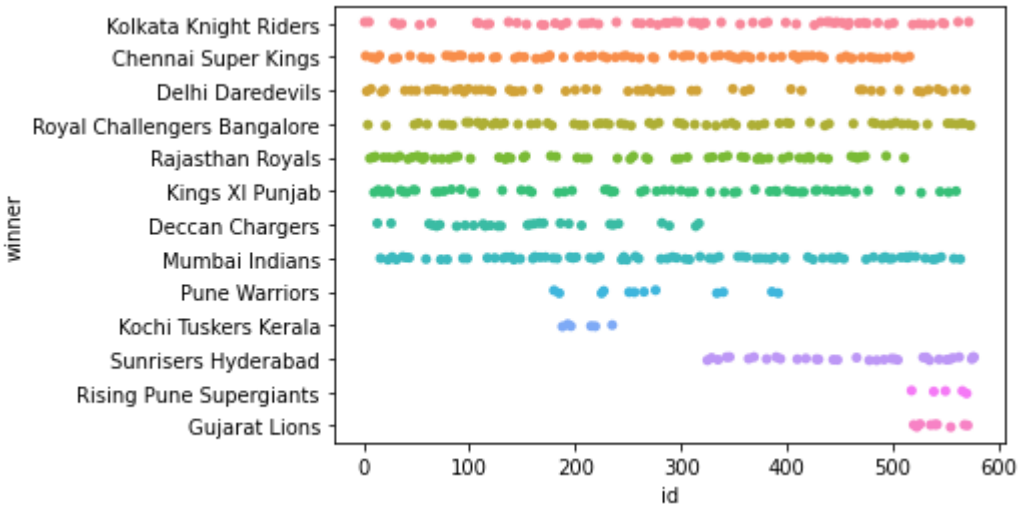

```
sns.jointplot("id", "win_by_runs", data=match, kind="hex")
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass th
  FutureWarning
<seaborn.axisgrid.JointGrid at 0x7fad6277b150>
```

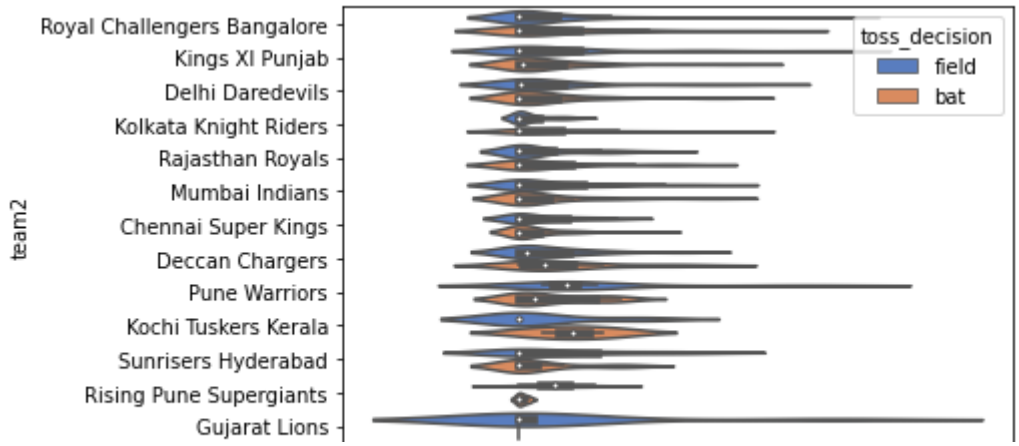



```
sns.stripplot(x="id", y="winner", data=match, jitter=0.05)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fad6098cbd0>
```

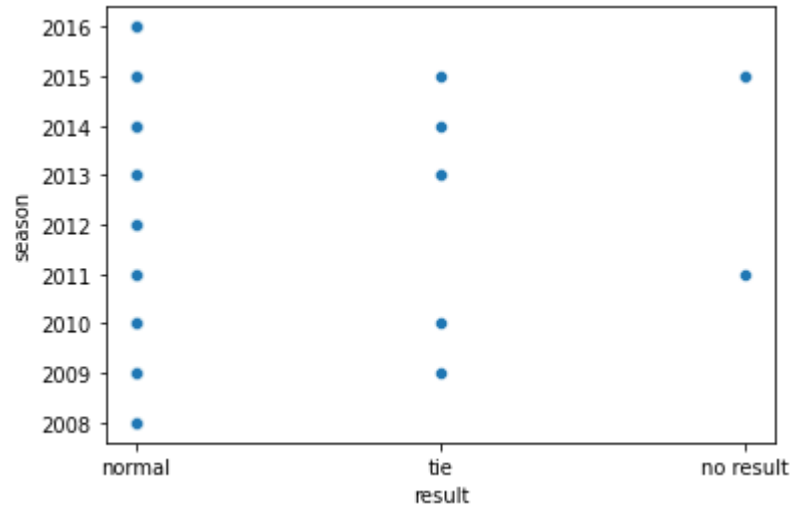



```
ax = sns.violinplot(x="win_by_runs", y="team2", hue="toss_decision",
                    data=match, palette="muted")
```




```
sns.scatterplot(data=match, x="result", y="season")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fad606ec150>
```



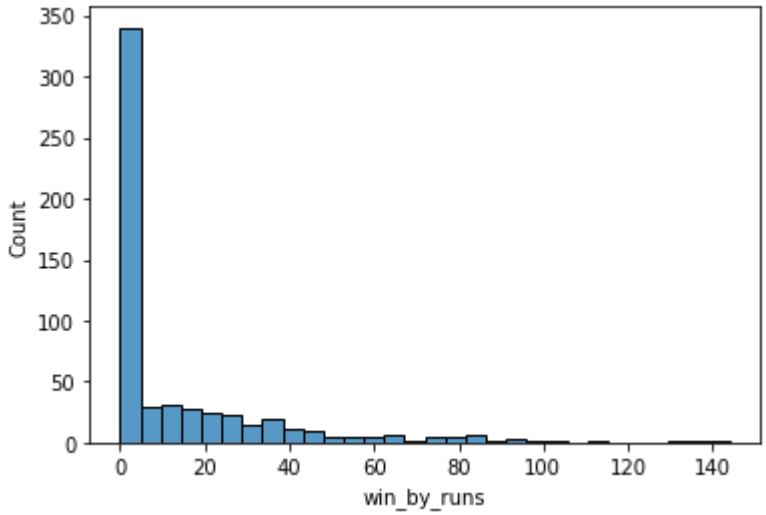

```
df = match.groupby(['season', 'toss_decision']).agg(mean_total_bill=("win_by_runs", 'mean'))
df = match.reset_index()

# plot barplot
sns.barplot(x="season",
            y="win_by_runs",
            hue="toss_decision",
            data=match)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fad606ac5d0>
```

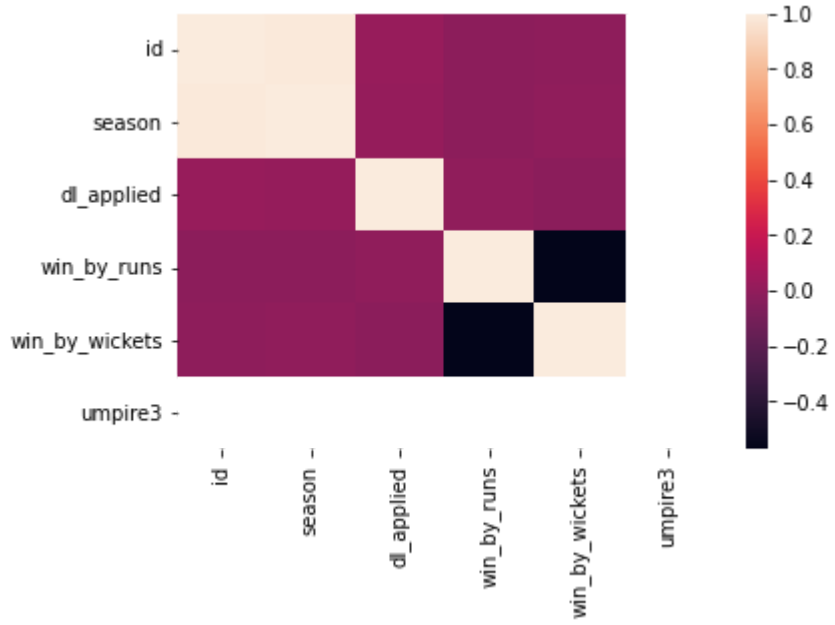

```
sns.histplot(data=match, x="win_by_runs")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fad5f9bba10>
```




```
dataplot=sns.heatmap(match.corr())

# displaying heatmap
plt.show()
```



KNN

[ ] ↳ *27 cells hidden*

Naïve Bayes

[ ] ↳ *7 cells hidden*

K-Means clustering

[ ] ↳ *13 cells hidden*

Hierarchical clustering

[ ] ↳ *10 cells hidden*

Decision Tree

[ ] ↳ *21 cells hidden*

Linear Regression

[ ] ↳ *10 cells hidden*

Logistic Regression

[ ] ↳ *38 cells hidden*

SVM

[ ] ↳ *10 cells hidden*

Inference

In the left panel, we see the model and the support vectors for 60 training points. In the right panel, we have doubled the number of training points, but the model has not changed: the three support vectors from the left panel are still the support vectors from the right panel. This insensitivity to the exact behavior of distant points is one of the strengths of the SVM model.

Multi layer feed forward neural network

[ ] ↳ *31 cells hidden*

BPN

[ ] ↳ *11 cells hidden*

Intrepretation

Backpropagation Neural Network is a simple and faster model compared to its earlier models. It is also a flexible and standard method. It does not need any prior knowledge for training.

BPN performance depends upon the kind of input data is used. It is quite sensitive to noisy data. We need to use a matrix-based approach instead of a mini-batch.

Backpropagation neural network is a method to optimize neural networks by propagating the error or loss into a backward direction. It finds loss for each node and updates its weights accordingly in order to minimize the loss using gradient descent.