EXPERIMENT NO. 10

AIM: Study and Implement Feature Engineering- Outline Detection

Theory:

Feature engineering is the process of selecting, modifying, or creating features (input variables) that can enhance the performance of machine learning algorithms. For outline detection, it involves the identification of boundaries or edges in data, typically in images, to highlight structures or patterns. This process is particularly important in computer vision and image processing, where algorithms detect and analyse the edges within an image to identify objects, shapes, or specific regions.

Outline detection relies on:

1. **Gradient-based Methods** (like the Sobel or Canny edge detector): Uses gradient changes in pixel intensity to locate edges.
2. **Thresholding Techniques**: Segments images based on pixel intensity to isolate and highlight edges.
3. **Morphological Operations**: Enhances or highlights outlines in binary or grayscale images.

By extracting these key features, the data becomes more informative for training models, helping them detect and classify objects or make other inferences based on the detected outlines.

Code:

```
# In[1]:
import pandas as pd
import matplotlib
from matplotlib import pyplot as plt
get_ipython().run_line_magic('matplotlib', 'inline')
matplotlib.rcParams['figure.figsize'] = (10,6)

# In[2]:
df = pd.read_csv("heights.csv")
df.sample(5)

# In[3]:
plt.hist(df.height, bins=20, rwidth=0.8)
plt.xlabel('Height (inches)')
plt.ylabel('Count')
plt.show()
```

```
#In[4]:
from scipy.stats import norm
import numpy as np
plt.hist(df.height, bins=20, rwidth=0.8, density=True)
plt.xlabel('Height (inches)')
plt.ylabel('Count')

rng = np.arange(df.height.min(), df.height.max(), 0.1)
plt.plot(rng, norm.pdf(rng,df.height.mean(),df.height.std()))

#In[5]:
df.height.mean()

#In[6]:
df.height.std()

#In[7]:
upper_limit = df.height.mean() + 3*df.height.std()
upper_limit

#In[8]:
inputs_n = inputs.drop(['company','job','degree'],axis='columns') lower_limit =
df.height.mean() -3*df.height.std()
lower_limit

#In[9]:
 df[(df.height>upper_limit) | (df.height<lower_limit)]

#In[10]:
df_no_outlier_std_dev = df[(df.height<upper_limit) & (df.height>lower_limit)]
df_no_outlier_std_dev.head()

#In[11]:
df_no_outlier_std_dev.shape

#In[12]:
df.shape

#In[13]:
df['zscore'] = ( df.height - df.height.mean() ) / df.height.std()
df.head(5)

#In[14]:
df[df['zscore']>3]
```

#In[15]:
```
df[df['zscore']<-3]
```

#In[16]:
```
df[(df.zscore<-3) | (df.zscore>3)]
```

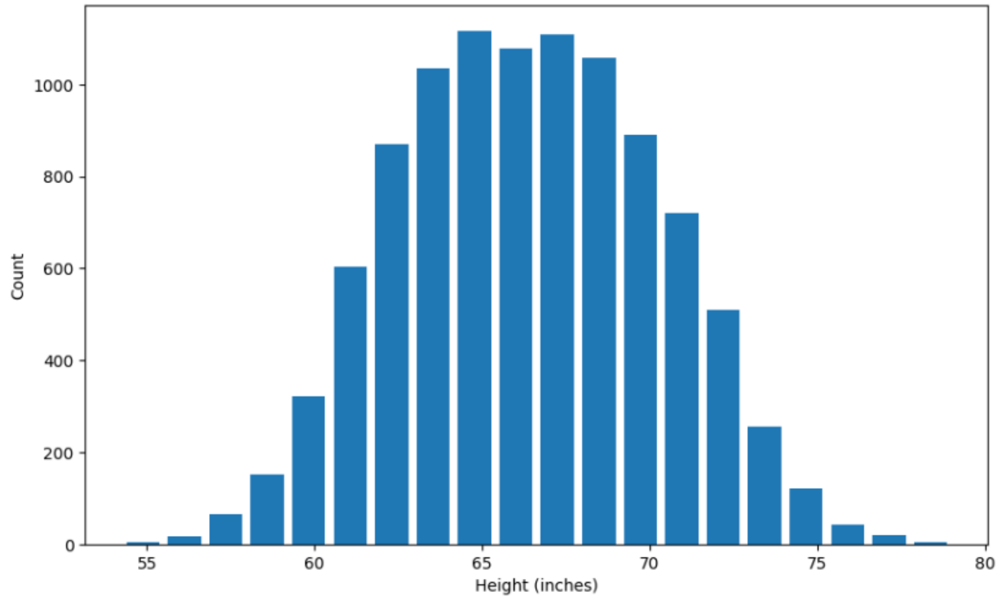
#In[17]:
```
df_no_outliers = df[(df.zscore>-3) & (df.zscore<3)]
df_no_outliers.head()
```

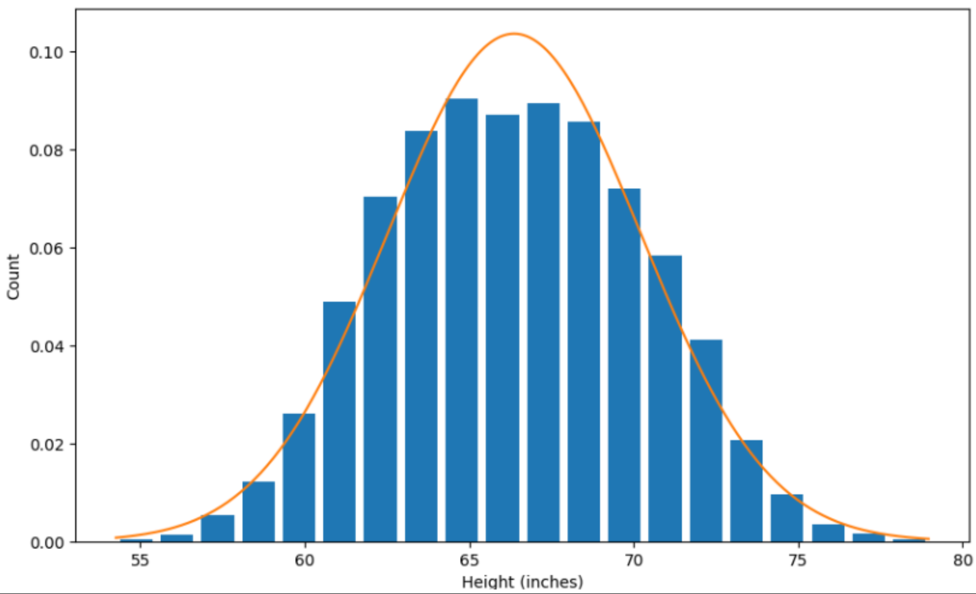
#In[18]:
```
df_no_outliers.shape
```

#In[19]:
```
df.shape
```

Output Snapshots:

	gender	height
49	Male	72.870360
5601	Female	59.762156
769	Male	65.747298
7417	Female	61.017988
8586	Female	68.222664






```
66.367559754866  3.847528120795573  77.91014411725271
54.824975392479274
```

	gender	height
994	Male	78.095867
1317	Male	78.462053
2014	Male	78.998742
3285	Male	78.528210
3757	Male	78.621374
6624	Female	54.616858
9285	Female	54.263133

	gender	height
0	Male	73.847017
1	Male	68.781904
2	Male	74.110105
3	Male	71.730978
4	Male	69.881796

```
(9993, 2)
```

```
(10000, 2)
```

	gender	height	zscore
0	Male	73.847017	1.943964
1	Male	68.781904	0.627505
2	Male	74.110105	2.012343
3	Male	71.730978	1.393991
4	Male	69.881796	0.913375

```
1.9453124999999998
```

	gender	height	zscore
994	Male	78.095867	3.048271
1317	Male	78.462053	3.143445
2014	Male	78.998742	3.282934
3285	Male	78.528210	3.160640
3757	Male	78.621374	3.184854

	gender	height	zscore
6624	Female	54.616858	-3.054091
9285	Female	54.263133	-3.146027

	gender	height	zscore
994	Male	78.095867	3.048271
1317	Male	78.462053	3.143445
2014	Male	78.998742	3.282934
3285	Male	78.528210	3.160640
3757	Male	78.621374	3.184854
6624	Female	54.616858	-3.054091
9285	Female	54.263133	-3.146027

	gender	height	zscore
0	Male	73.847017	1.943964
1	Male	68.781904	0.627505
2	Male	74.110105	2.012343
3	Male	71.730978	1.393991
4	Male	69.881796	0.913375

```
(9993, 3)
```

```
(10000, 3)
```

Learning Outcome: