In [35]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import preprocessing,svm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [36]:

```
df=pd.read_csv(r"C:\Users\raja\Downloads\fiat500_VehicleSelection_Dataset.csv")
df
```

Out[36]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	
0	1	lounge	51	882	25000	1	44.907242	8.611
1	2	pop	51	1186	32500	1	45.666359	12.241
2	3	sport	74	4658	142228	1	45.503300	11.417
3	4	lounge	51	2739	160000	1	40.633171	17.634
4	5	pop	73	3074	106880	1	41.903221	12.495
...	...	...	...	...	...	...	...	
1533	1534	sport	51	3712	115280	1	45.069679	7.704
1534	1535	lounge	74	3835	112000	1	45.845692	8.666
1535	1536	pop	51	2223	60457	1	45.481541	9.413
1536	1537	lounge	51	2557	80750	1	45.000702	7.682
1537	1538	pop	51	1766	54276	1	40.323410	17.568

1538 rows × 9 columns

In [37]:

```
df=df[['engine_power','km']]
```

In [38]:

```
df.columns=['power','dist']
df
```

Out[38]:

	power	dist
0	51	25000
1	51	32500
2	74	142228
3	51	160000
4	73	106880
...	...	...
1533	51	115280
1534	74	112000
1535	51	60457
1536	51	80750
1537	51	54276

1538 rows × 2 columns

In [39]:

```
df.head()
```

Out[39]:

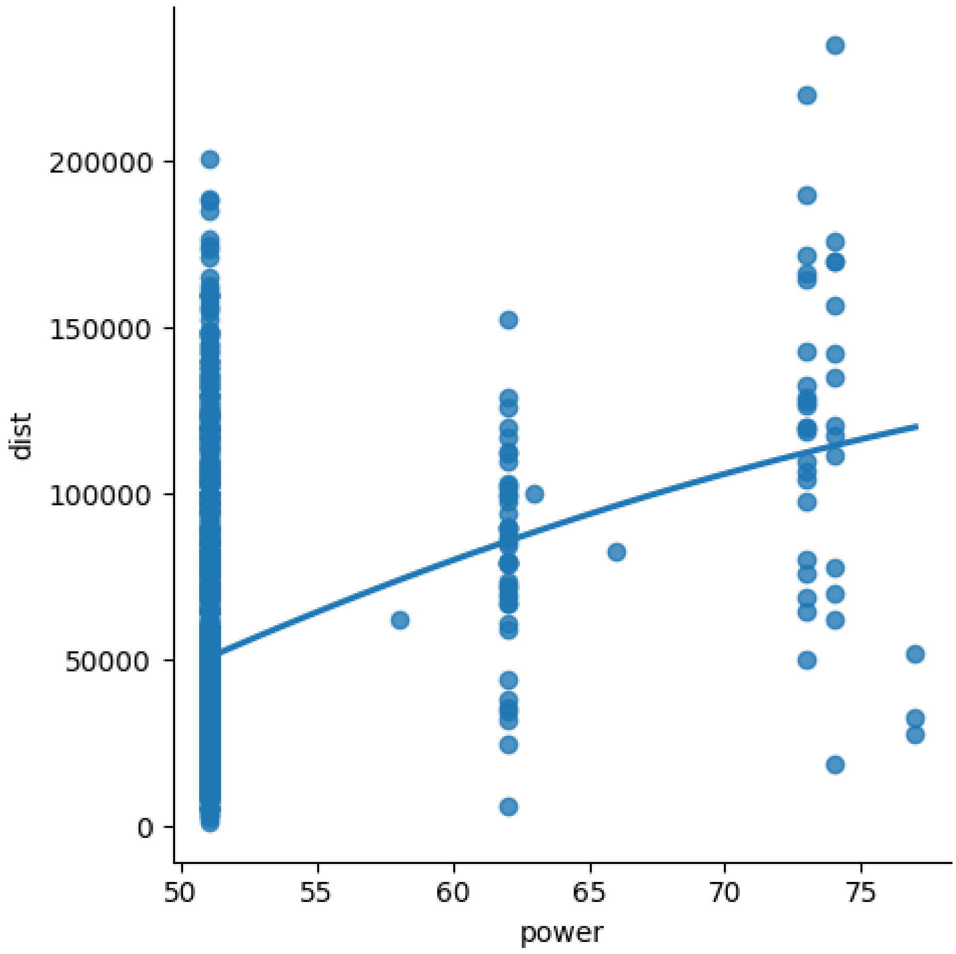
	power	dist
0	51	25000
1	51	32500
2	74	142228
3	51	160000
4	73	106880

In [40]:

```
sns.lmplot(x="power",y="dist",data=df,order=2,ci=None)
```

Out[40]:

```
<seaborn.axisgrid.FacetGrid at 0x1ba2b2a4790>
```



In [41]:

```
df.describe()
```

Out[41]:

	power	dist
count	1538.000000	1538.000000
mean	51.904421	53396.011704
std	3.988023	40046.830723
min	51.000000	1232.000000
25%	51.000000	20006.250000
50%	51.000000	39031.000000
75%	51.000000	79667.750000
max	77.000000	235000.000000

In [42]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1538 entries, 0 to 1537
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   power   1538 non-null   int64
 1   dist    1538 non-null   int64
dtypes: int64(2)
memory usage: 24.2 KB
```

In [43]:

```
df.isna().any()
```

Out[43]:

```
power    False
dist     False
dtype: bool
```

In [45]:

```
x=np.array(df['power']).reshape(-1,1)
y=np.array(df['dist']).reshape(-1,1)
```

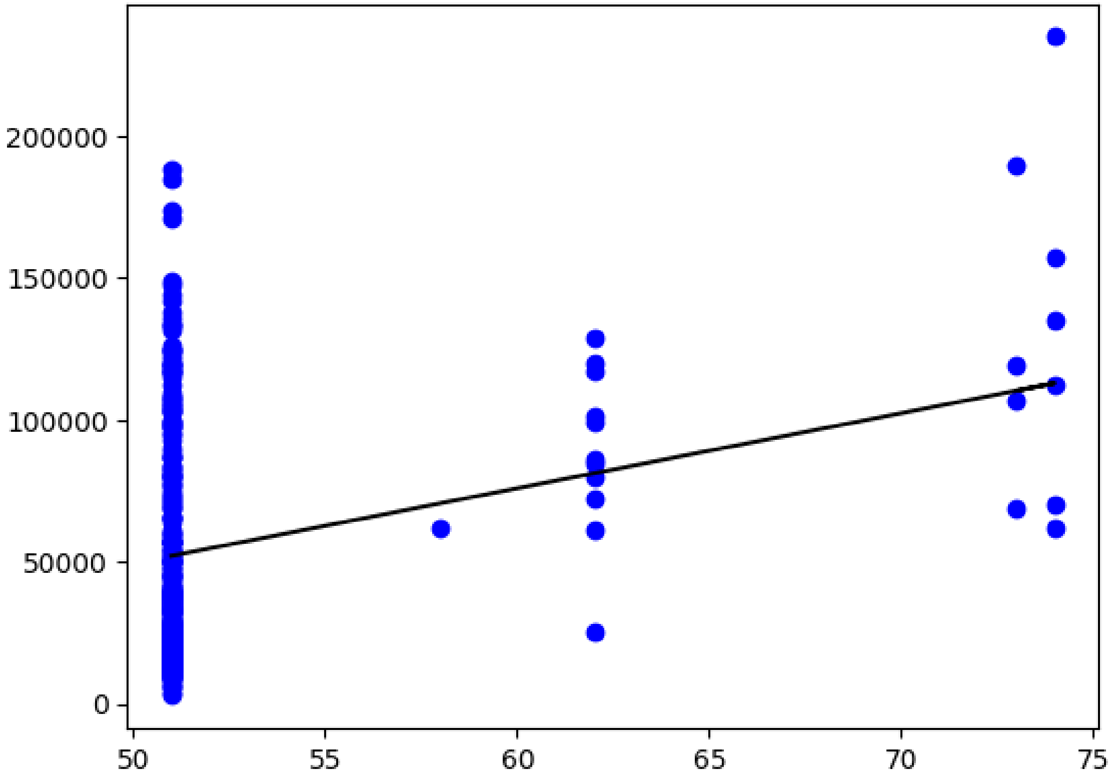
In [46]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
regr=LinearRegression()
regr.fit(x_train,y_train)
print(regr.score(x_test,y_test))
```

```
0.10407606187567098
```

In [47]:

```
y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='k')
plt.show()
```

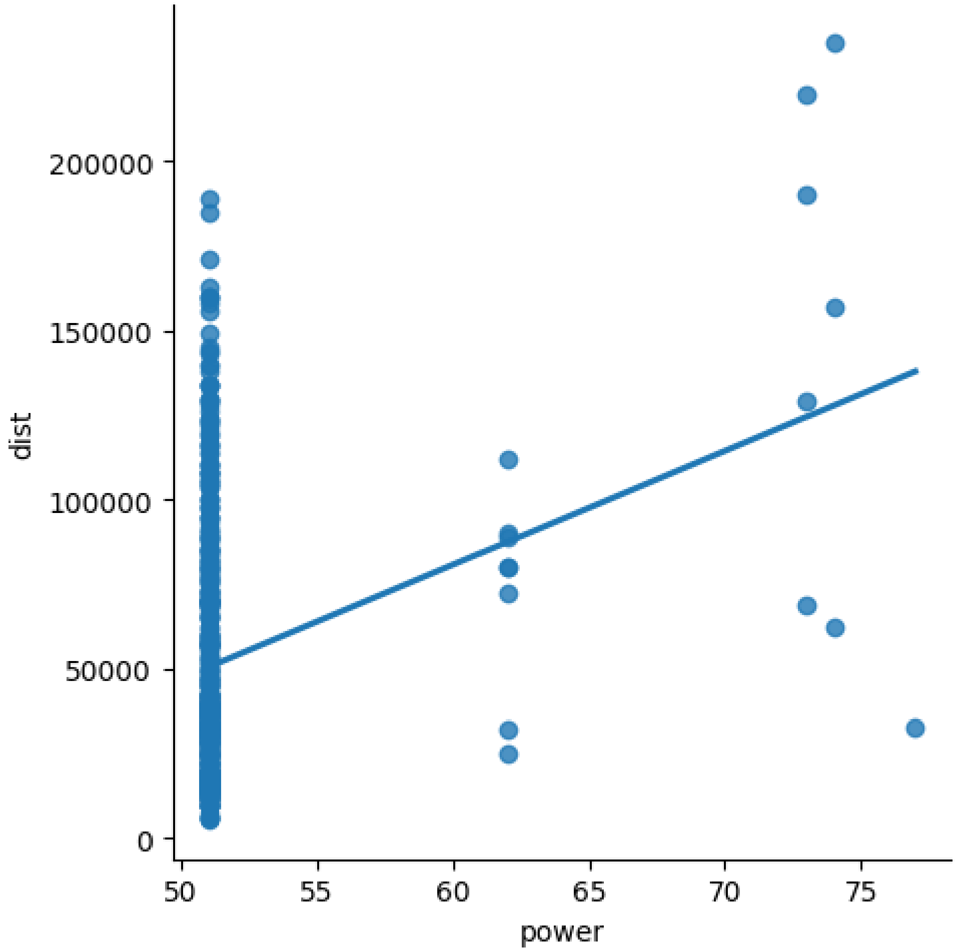


In [55]:

```
df500=df[:][1000:1500]
sns.lmplot(x='power',y='dist',data=df500,order=1,ci=None)
```

Out[55]:

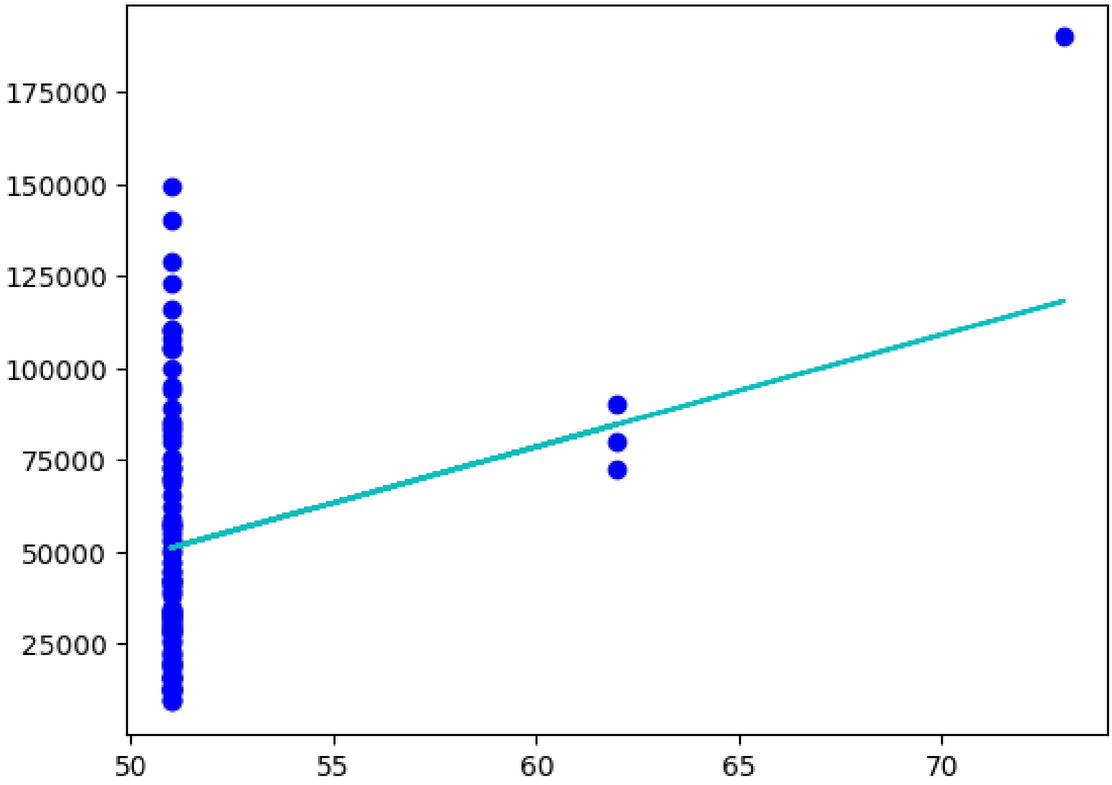
```
<seaborn.axisgrid.FacetGrid at 0x1ba2e6abca0>
```



In [56]:

```
df500.fillna(method='ffill',inplace=True)
x=np.array(df500['power']).reshape(-1,1)
y=np.array(df500['dist']).reshape(-1,1)
df500.dropna(inplace=True)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
regr=LinearRegression()
regr.fit(x_train,y_train)
print("regression:",regr.score(x_test,y_test))
y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='c')
plt.show()
```

```
regression: 0.11205332531587187
```



In [57]:

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
model=LinearRegression()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
r2=r2_score(y_test,y_pred)
print("r2 score:",r2)
```

```
r2 score: 0.11205332531587187
```

In [ ]:

```
#conclusion:
      The model is very poor because r2 score and regression is very less
      while taking small dataset also the r2 score is very less .so linear
      regression is not accurate.
      -->distance is increases power is also increases
```