Data Science Project - Sales Forecast

June 24, 2021

1 Data Science Project - Sales Forecast

Our challenge is to be able to predict the sales that we will have in a given period based on spending on ads in the 3 large networks that the Hashtag company invests in: TV, Newspaper and Radio

- Step by Step of a Data Science Project
- Step 1: Understanding the Challenge
- Step 2: Understanding the Area/Company
- Step 3: Data Extraction/Obtainment
- Step 4: Data Adjustment (Treatment/Cleaning)
- Step 5: Exploratory Analysis
- Step 6: Modeling + Algorithms (This is where Artificial Intelligence comes in, if necessary)
- Step 7: Interpretation of Results

[2]:
```
import pandas as pd
table = pd.read_csv("advertising.csv")
display(table)
```

```
        TV  Radio  Newspaper  Sales
0    230.1   37.8       69.2   22.1
1     44.5   39.3       45.1   10.4
2     17.2   45.9       69.3   12.0
3    151.5   41.3       58.5   16.5
4    180.8   10.8       58.4   17.9
..     ...    ...        ...    ...
195   38.2    3.7       13.8    7.6
196   94.2    4.9        8.1   14.0
197  177.0    9.3        6.4   14.8
198  283.6   42.0       66.2   25.5
199  232.1    8.6        8.7   18.4

[200 rows x 4 columns]
```

[3]:
```
table.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
```

```
 0   TV          200 non-null    float64
 1   Radio       200 non-null    float64
 2   Newspaper   200 non-null    float64
 3   Sales       200 non-null    float64
dtypes: float64(4)
memory usage: 6.4 KB
```

[11]:
```
display(table["TV"].sum())
display(table["Radio"].sum())
display(table["Newspaper"].sum())
display(table["Sales"].sum())
```

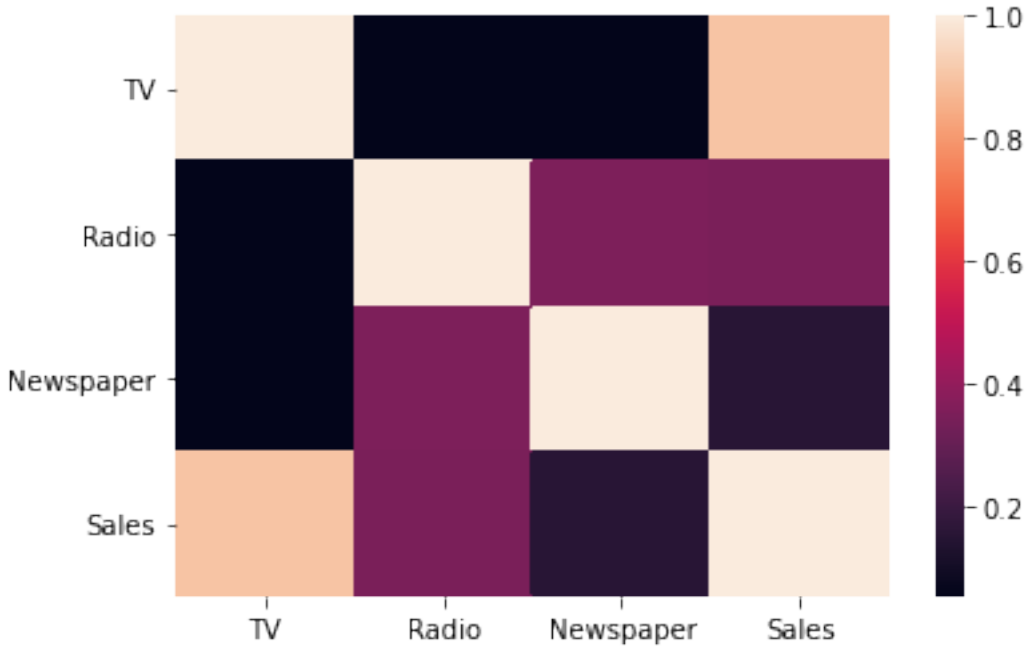
```
29408.5
```

```
4652.800000000001
```

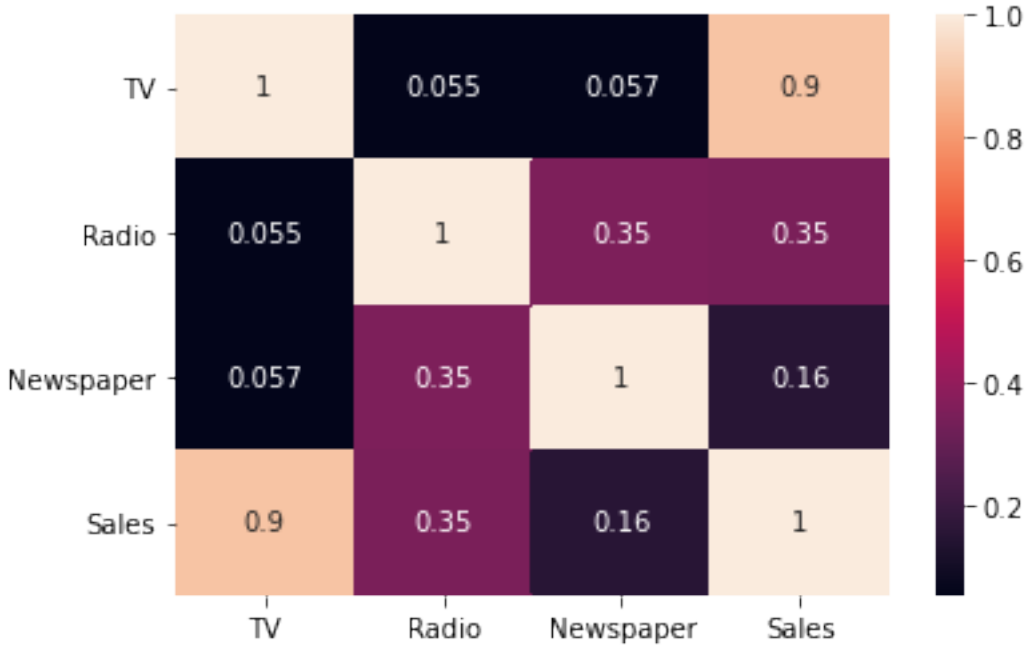
```
6110.799999999999
```

```
3026.1000000000004
```

[17]:
```
#vamos ver a correlacao(-1 para 1) entre cada um dos itens

import seaborn as sns
import matplotlib.pyplot as plt
sns.heatmap(table.corr())
plt.show()
```



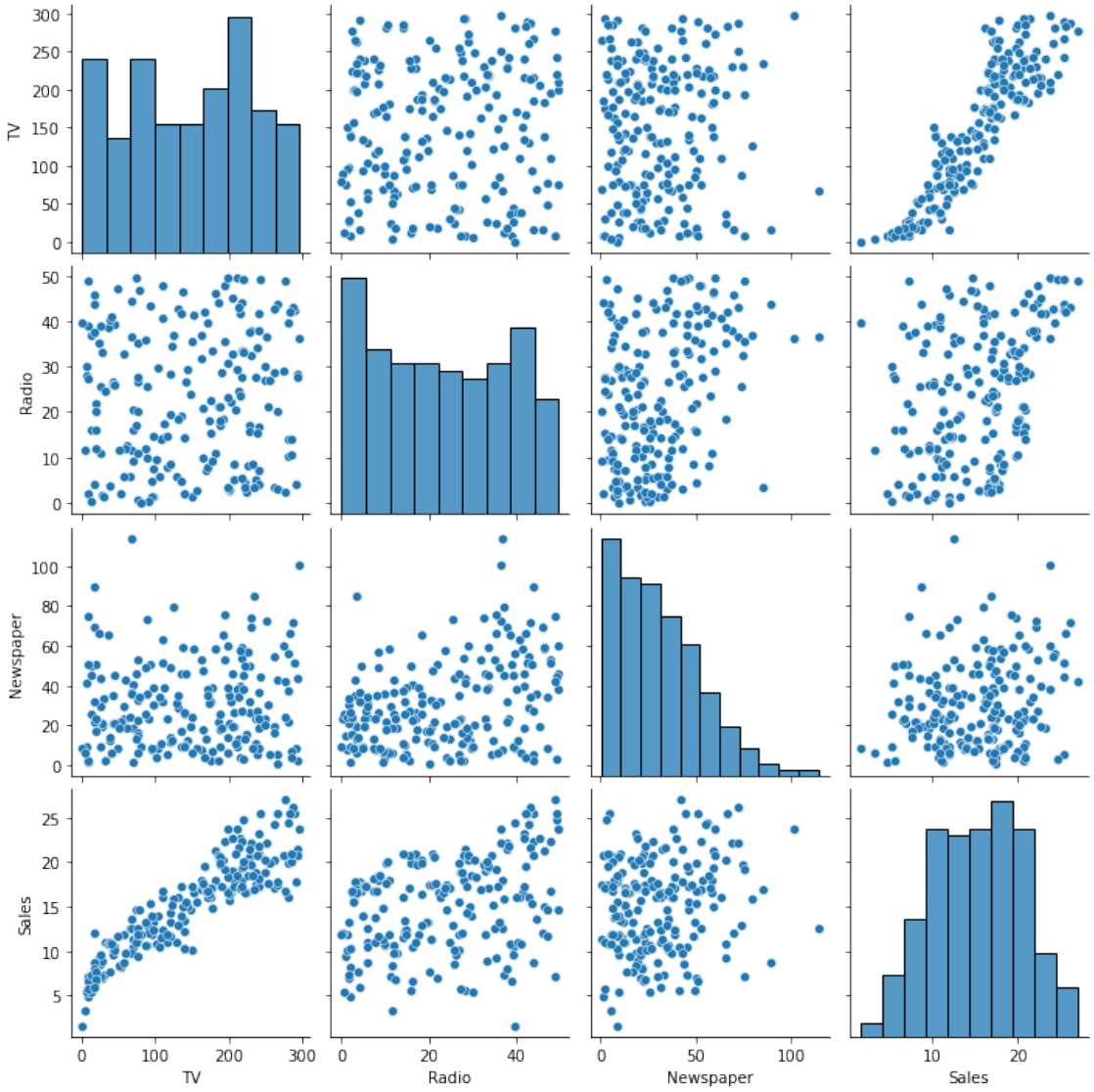
[18]:
```
sns.heatmap(table.corr(), annot=True)
plt.show()
```



[19]:
```
#sales and TV have a corr of 0,9. What that means? When I increase my␣
↪investment in TV, my sales will probably increase.
#sales and radio have a corr of 0,35. That means when I increase my investment␣
↪in Radio, my sales may have less impact.
```

[20]:
```
sns.heatmap(table.corr(), annot=True, cmap="Wistia")
plt.show()
```

[23]:
```
sns.pairplot(table)
plt.show()

# what are the features (TV,Newspaper,Radio) that have the greatest correlation␣
 ↪with sales?
# we always have to check the features that have a bigger corr with sales. In␣
 ↪that case, first TV, next Radio and then Newspaper.
# we also have to check the corr between the features. The features are not␣
 ↪supposed to have a big corr between them.
```



[26]:
```
# Now let's separate training data and testing data
# The AI will use the training data to learn. Then, the test data to see if it␣
 ↪learned well
# y = sales (what I want to find out) and x = TV, Radio and Newspaper
# x_training x_testing y_training y_testing
```

[31]:
```
from sklearn.model_selection import train_test_split
# now we are gonna create x and y
y = table["Sales"]
x = table.drop("Sales", axis=1)
# we are gonna separete the database between testing data and training data -␣
 ↪train_test_split(always the same code to do that)
```

```
# x_training, x_testing, y_training, y_testing = train_test_split(x, y)
x_training, x_testing, y_training, y_testing = train_test_split(x, y,␣
↪test_size=0.3, random_state=1)
```

- We have a regression problem - Let's choose the models we're going to use:
  - Linear Regression
  - Random Forest (Decision Tree)

[33]:
```
# first we import the infomations

from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor

# Next, we are going to create the AI models

model_linearregression = LinearRegression()
model_decisiontree = RandomForestRegressor()

# Then, now we are going to training (fit) both AI models

model_linearregression.fit(x_training, y_training)
model_decisiontree.fit(x_training, y_training)
```

[33]:
```
RandomForestRegressor()
```

- AI Test and Best Model Evaluation
  - Let's use R^2 -> says the % that our model can explain what happens
  - We will also look at the MSE (Mean Square Error) -> it says how much our model "errs" when trying to make a prediction

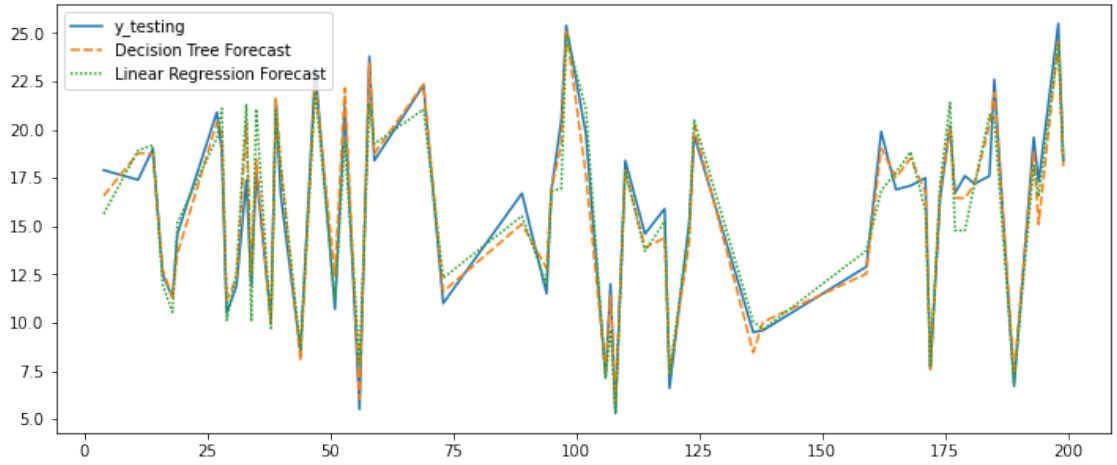
[41]:
```
# R² -> says the % that our model can explain what happens.
from sklearn import metrics

# create the forecasts
forecast_linearregression = model_linearregression.predict(x_testing)
forecast_decisiontree = model_decisiontree.predict(x_testing)

# compare the models

print(metrics.r2_score(y_testing, forecast_decisiontree)*100)
print(metrics.r2_score(y_testing, forecast_linearregression)*100)
```

```
95.97252731896087
90.71151423684273
```

[42]:
```
#Decision Tree is better
```

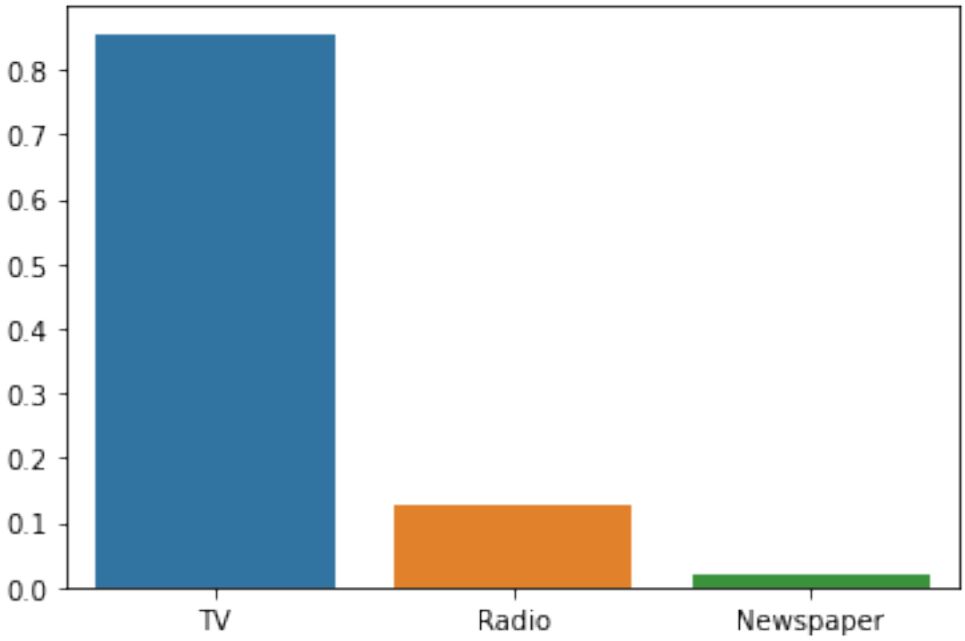
- Visualização Gráfica das Previsões

[45]:
```
#create an empty table
new_table = pd.DataFrame()

# I'm gonna add info in that table
new_table["y_testing"] = y_testing
new_table["Decision Tree Forecast"] = forecast_decisiontree
new_table["Linear Regression Forecast"] = forecast_linearregression

# show this table as a graph
plt.figure(figsize=(12,5))
sns.lineplot(data=new_table)
plt.show()
```



[47]:
```
# How important is each feature to sales?

sns.barplot(x=x_training.columns, y=model_decisiontree.feature_importances_)
plt.show
```

[47]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



2 Conclusions:

- Decision Tree is better
- TV is super important - we should invest more in Ads on TV