DAT200 CA3 2022

Kaggle username: Mohamed Atteyeh

Imports

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
```

Reading data

In [2]:

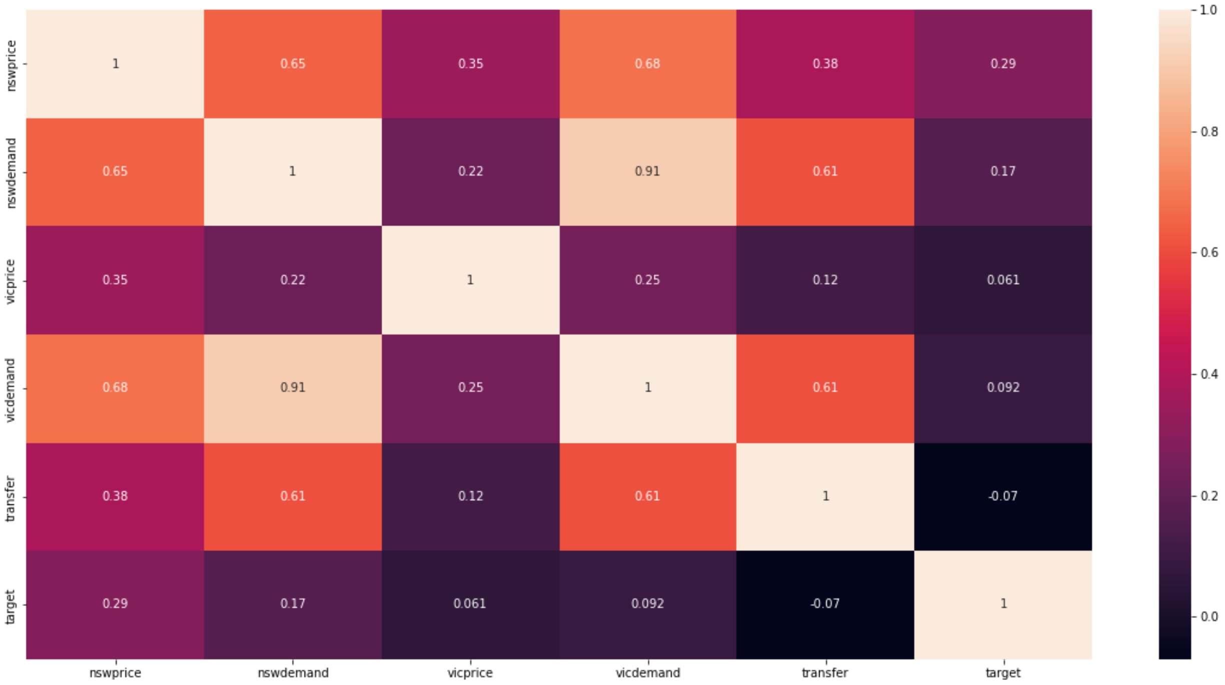
```
training_data = pd.read_csv('train.csv', index_col= 0 ) # Training Data
test_data = pd.read_csv('test.csv',index_col = 0) # Test Data
```

Data exploration and visualisation

In [3]:

```
# Insert your code below
# ====================== # loading the data
nan_values = training_data.isna().sum() # Checking for nan values , this gave
corr_data=training_data[training_data.columns].corr() # finding the correlati

# Here is where we plot
plt.figure(figsize=(20,10))
sns.heatmap(corr_data, annot=True)
plt.show()
```



Data cleaning

In [4]:

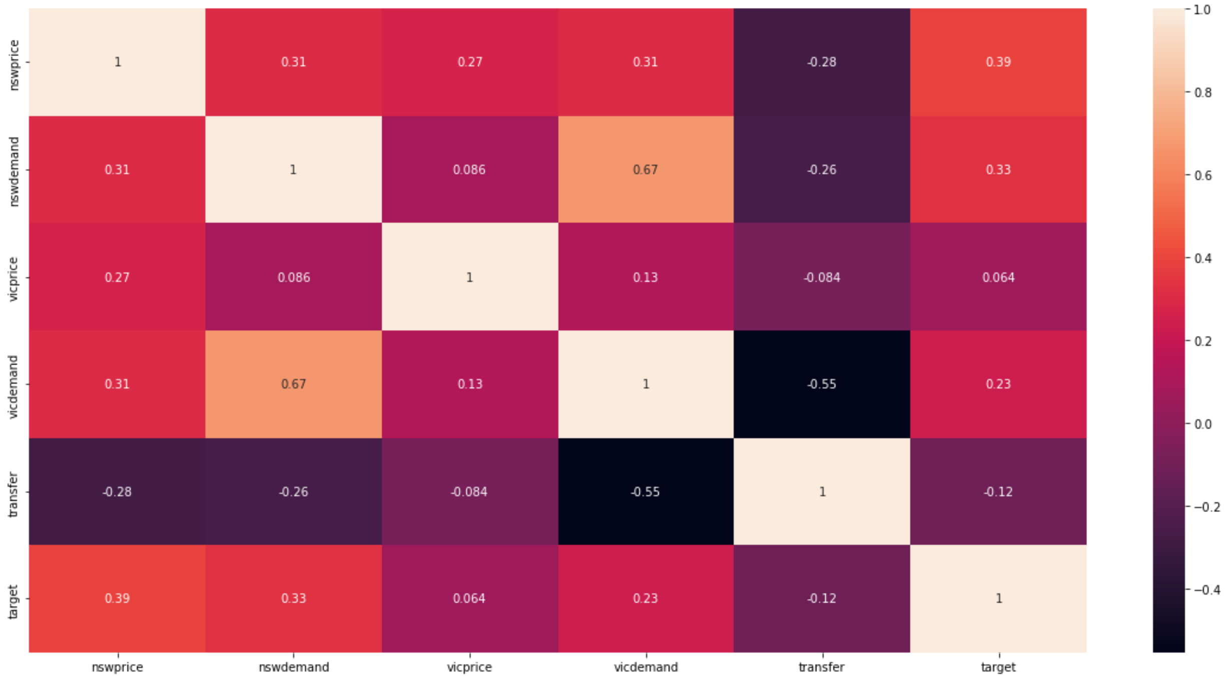
```
outliers = training_data.loc[training_data['transfer'] < 0]
training_data = training_data.drop(training_data.index[list(outliers.index)])
```

Data exploration after cleaning

In [5]:

```
# Insert your code below
# ====================== # loading the data
nan_values = training_data.isna().sum() # Checking for nan values , this gave
corr_data=training_data[training_data.columns].corr() # finding the correlati

# Here is where we plot
plt.figure(figsize=(20,10))
sns.heatmap(corr_data, annot=True)
plt.show()
```



Data preprocessing

In [6]:

```
# Processsing the data, and splitting the X intercept and y intercept
X = training_data.iloc[:,:-1].copy()
y = training_data.iloc[:,-1].copy()
```

Modelling

In [ ]:

```
all_acc_test = []
all_acc_train = []
n_values = []

# Testing with multiple train and test splits, and finding the best value

for n in range(100,300):
    train_acc = []
    test_acc = []
    n_values.append(n)
    for r in range(1,10):
        X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,ra

        forest = RandomForestClassifier(criterion='gini',
                                        n_estimators=n,
                                        random_state= 1,
                                        n_jobs=-1)
        forest.fit(X_train, y_train)
        train_acc.append(forest.score(X_train, y_train))
        test_acc.append(forest.score(X_test, y_test))

    all_acc_test.append(np.mean(test_acc))
    all_acc_train.append(np.mean(train_acc))
```

Evaluation

In [ ]:

```
#Evaluating the model, and checking the accuaracy
jmax = max(all_acc_test)
n_value = (str(i) for i,j in zip(n_values,all_acc_test) if j == jmax)
train_accuarcy = (str(k) for k,j in zip(all_acc_train,all_acc_test) if j == jı
print(','.join(n_value), ','.join(train_accuarcy), jmax) # Here i looked at t.
```

Kaggle submission

In [ ]:

```
# The submission model, with chosen parameters from the Evaluation.

forest_xtrain = X
forest_ytrain = y
forest_test = test_data
forest = RandomForestClassifier(criterion='gini',
                                n_estimators=163,
                                random_state= 100,
                                n_jobs=-1)
forest.fit(forest_xtrain, forest_ytrain)

# The pridiction and the submission file to Kaggle
forest_target_predictions = forest.predict(forest_test)
output = pd.DataFrame({'index': forest_test.index,'target': forest_target_pre
output.to_csv('submission_forest',index=False)
```