

AI_Valuation_Model

March 19, 2022

#Added starter code that we can later use for the actual data.

```
[ ]: import pandas as pd

# The file to read
used_car_data_file = 'ExampleData.csv'

# Read the file into the variable used_car_data
used_car_data = pd.read_csv(used_car_data_file)
```

```
[2]: # The variable y is the target that we want to predict
y = used_car_data.SalePrice
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-2-560a9ccaa3ef> in <module>()
      1 # The variable y is the target that we want to predict
----> 2 y = used_car_data.SalePrice

NameError: name 'used_car_data' is not defined
```

```
[ ]: # List of features that we want to use for the prediction of the target ↵
      ↪ (SalePrice)
prediction_features = ['Make', 'Model', 'Badge', 'Colour', 'Odometer']

# The variable X is a dataframe that contains the choice of predictive features
X = used_car_data[prediction_features]
```

```
[ ]: from sklearn.model_selection import train_test_split

# Splitting the data into training data (train) and validation data (val)
train_X, val_X, train_y, val_y = train_test_split(X, y, random_state = 1)
```

```
[ ]: from sklearn.ensemble import RandomForestRegressor

# Using a random forest regressor algorithm to keep it a bit more simple for ↵
      ↪ the moment
```

```
used_car_data_model = RandomForestRegressor(random_state = 1)

# Fit the model using the predictive features and the target we are predicting
used_car_data_model.fit(train_X,train_y)
```

```
[ ]: # The price predictions
val_price_predictions = used_car_data_model.predict(val_X)
print(val_price_predictions)
```

```
[ ]: from sklearn.metrics import mean_absolute_error

# Calculating the Mean Absolute Error (MAE) in the validation data
val_mae = mean_absolute_error(val_y, val_price_predictions)

print(val_mae)
```

```
[ ]:
```