Deep Azure@McKesson Steve Klonicki

Final Project

Using Azure Machine Learning to predict house prices

Problem Statement:

In 2017, around 5.57 million existing homes were sold in the US (Statista, U.S. existing home sales 2005-2018). Determining the optimal sell price is a critical decision of the seller. If priced too low, the seller leaves money on the table. If priced too high, the house will sit on the market unsold. A negative perception can result when a house is on the market for a considerable amount of time, or when the price is reduced often. Using existing sales, how accurately can we predict the selling price of a home before it is put onto the market? We will attempt to build such a system to predict housing prices using Azure Machine Learning.

Data Source:

Kaggle House Prices: Advanced Regression Techniques
https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data
I am working with the training data, which allows me to evaluate the effectiveness of the model against known data. The training data consists of 1,460 rows of data of house sales, each with 81 attributes.

- SalePrice - the property's sale price in dollars. This is the target variable that you're trying to predict. **SalePrice is the attribute my system will predict.**
- MSSubClass: The building class
- MSZoning: The general zoning classification
- LotFrontage: Linear feet of street connected to property
- LotArea: Lot size in square feet
- Street: Type of road access
- Alley: Type of alley access
- LotShape: General shape of property
- LandContour: Flatness of the property
- Utilities: Type of utilities available
- LotConfig: Lot configuration
- LandSlope: Slope of property
- Neighborhood: Physical locations within Ames city limits
- Condition1: Proximity to main road or railroad
- Condition2: Proximity to main road or railroad (if a second is present)
- BldgType: Type of dwelling
- HouseStyle: Style of dwelling
- OverallQual: Overall material and finish quality
- OverallCond: Overall condition rating
- YearBuilt: Original construction date
- YearRemodAdd: Remodel date
- RoofStyle: Type of roof
- RoofMatl: Roof material
- Exterior1st: Exterior covering on house
- Exterior2nd: Exterior covering on house (if more than one material)
- MasVnrType: Masonry veneer type
- MasVnrArea: Masonry veneer area in square feet

- ExterQual: Exterior material quality
- ExterCond: Present condition of the material on the exterior
- Foundation: Type of foundation
- BsmtQual: Height of the basement
- BsmtCond: General condition of the basement
- BsmtExposure: Walkout or garden level basement walls
- BsmtFinType1: Quality of basement finished area
- BsmtFinSF1: Type 1 finished square feet
- BsmtFinType2: Quality of second finished area (if present)
- BsmtFinSF2: Type 2 finished square feet
- BsmtUnfSF: Unfinished square feet of basement area
- TotalBsmtSF: Total square feet of basement area
- Heating: Type of heating
- HeatingQC: Heating quality and condition
- CentralAir: Central air conditioning
- Electrical: Electrical system
- 1stFlrSF: First Floor square feet
- 2ndFlrSF: Second floor square feet
- LowQualFinSF: Low quality finished square feet (all floors)
- GrLivArea: Above grade (ground) living area square feet
- BsmtFullBath: Basement full bathrooms
- BsmtHalfBath: Basement half bathrooms
- FullBath: Full bathrooms above grade
- HalfBath: Half baths above grade
- Bedroom: Number of bedrooms above basement level
- Kitchen: Number of kitchens
- KitchenQual: Kitchen quality
- TotRmsAbvGrd: Total rooms above grade (does not include bathrooms)
- Functional: Home functionality rating
- Fireplaces: Number of fireplaces
- FireplaceQu: Fireplace quality
- GarageType: Garage location
- GarageYrBlt: Year garage was built
- GarageFinish: Interior finish of the garage
- GarageCars: Size of garage in car capacity
- GarageArea: Size of garage in square feet
- GarageQual: Garage quality
- GarageCond: Garage condition
- PavedDrive: Paved driveway
- WoodDeckSF: Wood deck area in square feet
- OpenPorchSF: Open porch area in square feet
- EnclosedPorch: Enclosed porch area in square feet
- 3SsnPorch: Three season porch area in square feet
- ScreenPorch: Screen porch area in square feet
- PoolArea: Pool area in square feet
- PoolQC: Pool quality

- Fence: Fence quality
- MiscFeature: Miscellaneous feature not covered in other categories
- MiscVal: $Value of miscellaneous feature
- MoSold: Month Sold
- YrSold: Year Sold
- SaleType: Type of sale
- SaleCondition: Condition of sale

Hardware Used:
Windows 10 64 bit processor desktop

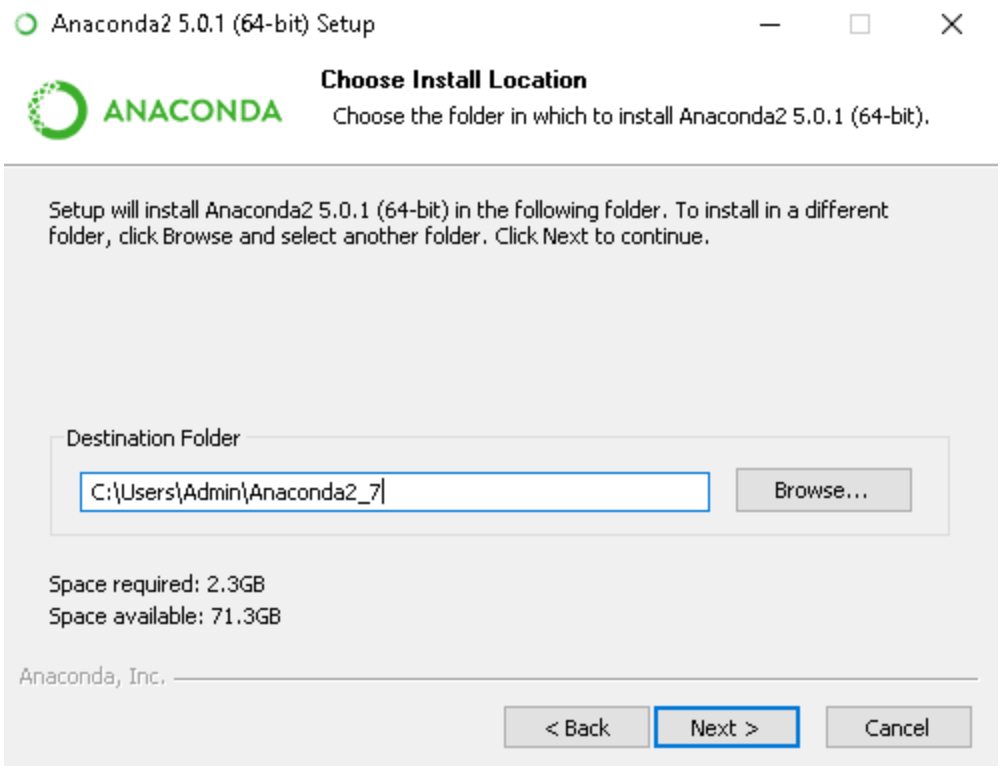
Software Used:
Microsoft Azure Machine Learning Studio (https://studio.azureml.net/)
Anaconda 5.0 distribution of Python 64 bit (https://www.anaconda.com/), which includes:

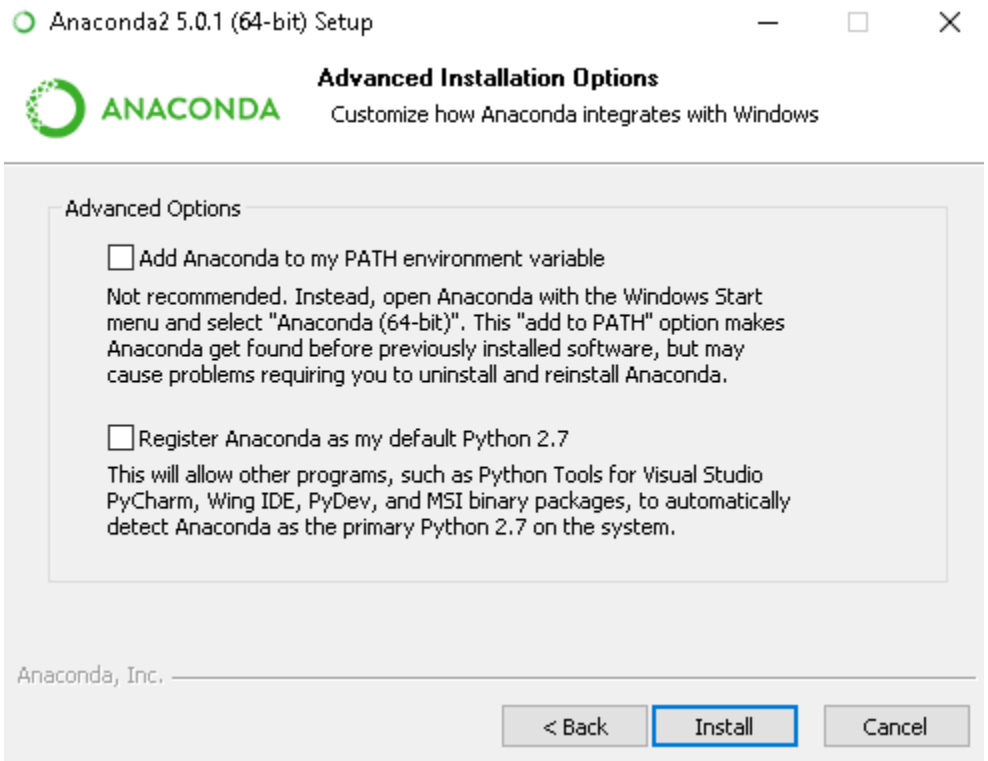
- Python 2.7.14
- Jupyter (visualization tool)

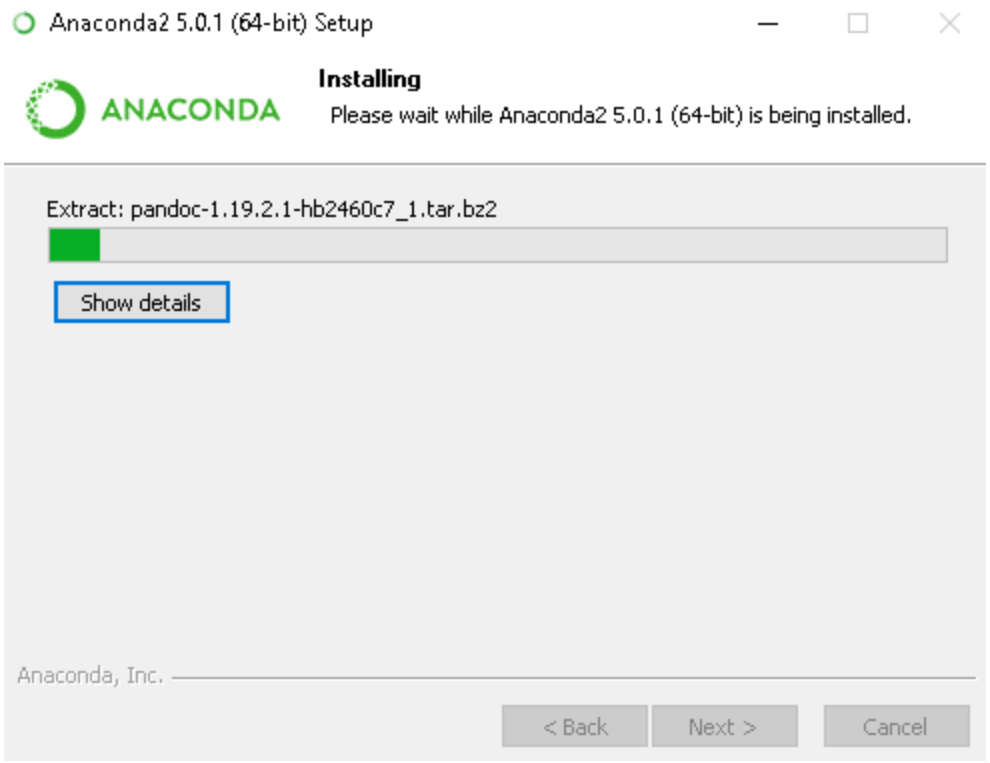
Microsoft Azure Machine Learning Studio is a web based tool, which requires no installation or configuration to use.

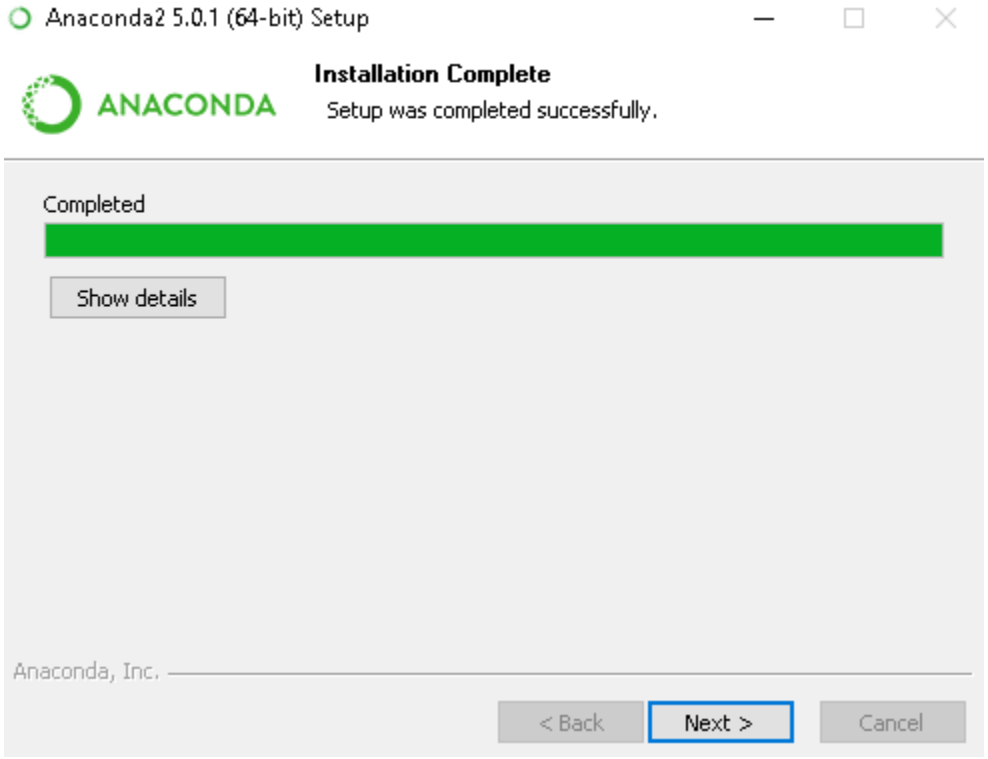
Jupyter Notebooks, using Python 2.7, within Microsoft Azure Machine Learning Studio is a web based tool, which requires no installation or configuration to use.

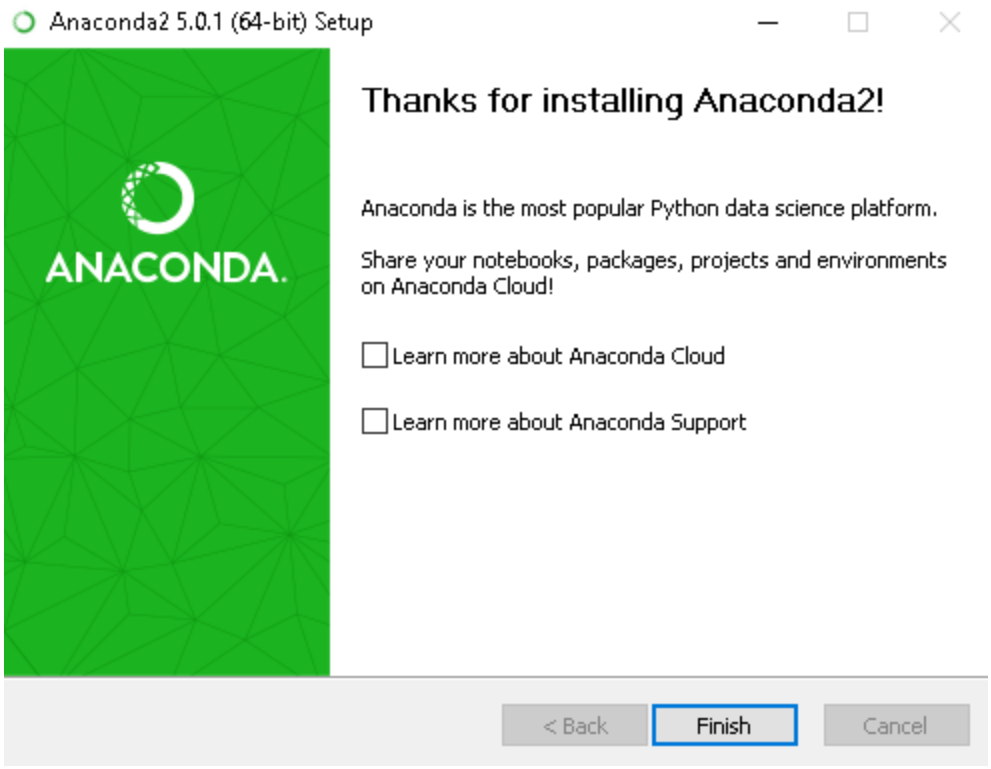
Anaconda installation steps:


Anaconda2 5.0.1 (64-bit) Setup
ANACONDA
Choose Install Location
Choose the folder in which to install Anaconda2 5.0.1 (64-bit).
Setup will install Anaconda2 5.0.1 (64-bit) in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.
Destination Folder
C:\Users\Admin\Anaconda2_7
Browse...
Space required: 2.3GB
Space available: 71.3GB
Anaconda, Inc.
< Back
Next >
Cancel

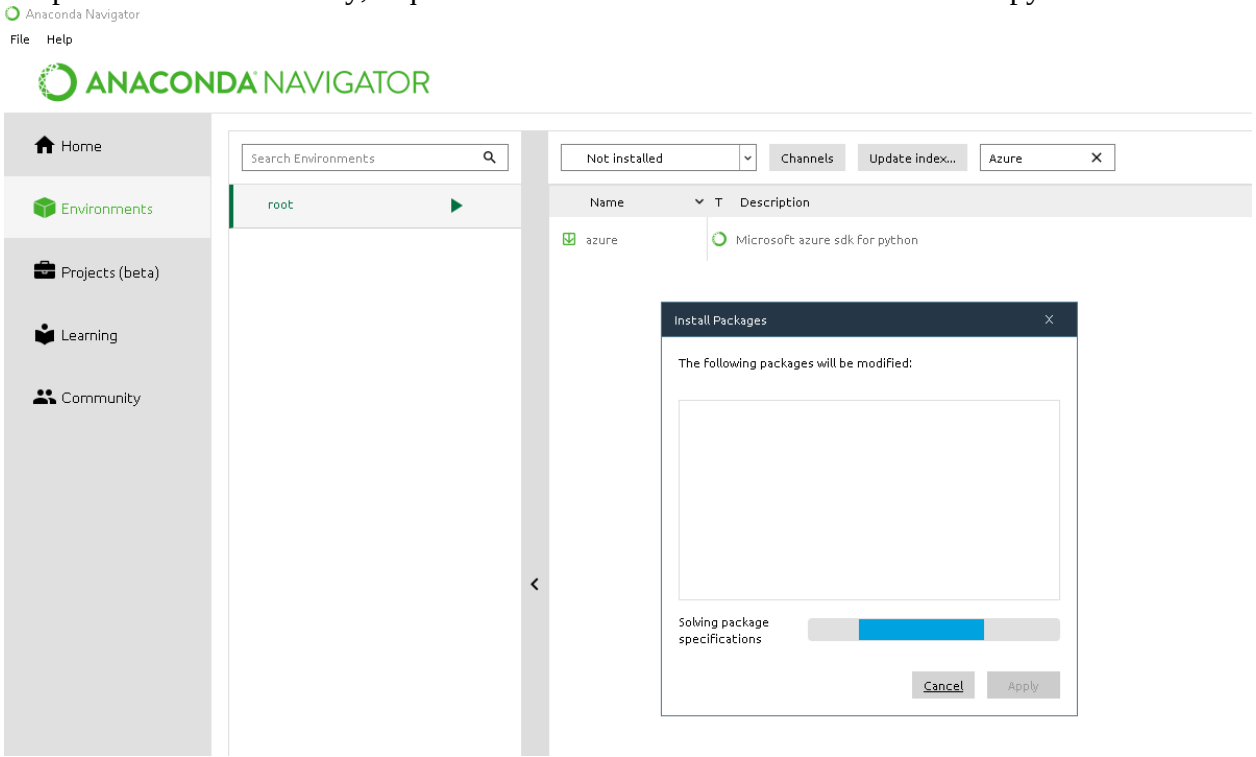

Anaconda2 5.0.1 (64-bit) Setup
ANACONDA
Advanced Installation Options
Customize how Anaconda integrates with Windows
Advanced Options
Add Anaconda to my PATH environment variable
Not recommended. Instead, open Anaconda with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.
Register Anaconda as my default Python 2.7
This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 2.7 on the system.
Anaconda, Inc.
< Back
Install
Cancel


Anaconda2 5.0.1 (64-bit) Setup
ANACONDA
Installing
Please wait while Anaconda2 5.0.1 (64-bit) is being installed.
Extract: pandoc-1.19.2.1-hb2460c7_1.tar.bz2
Show details
Anaconda, Inc.
< Back
Next >
Cancel


Anaconda2 5.0.1 (64-bit) Setup
ANACONDA
Installation Complete
Setup was completed successfully.
Completed
Show details
Anaconda, Inc.
< Back
Next >
Cancel



I imported the Azure library, required to run the Azure visualization code in Jupyter.

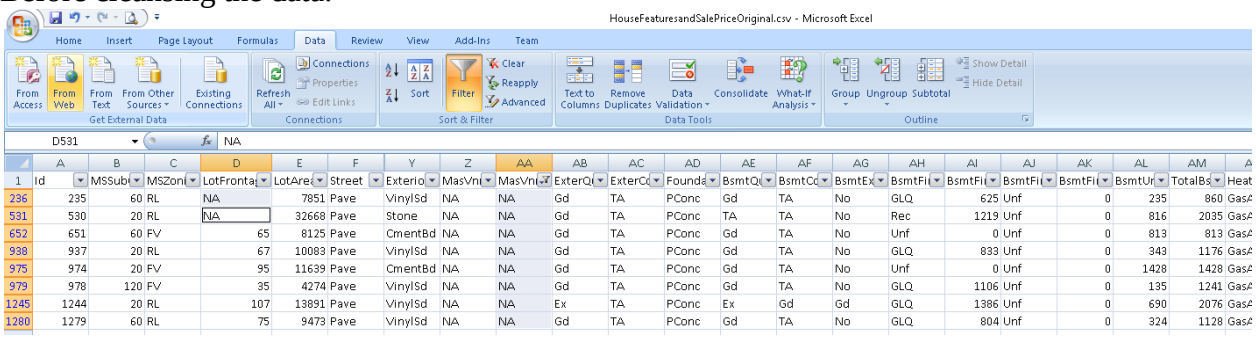


Data (and cleansing):

Deep Azure@McKesson Steve Klonicki

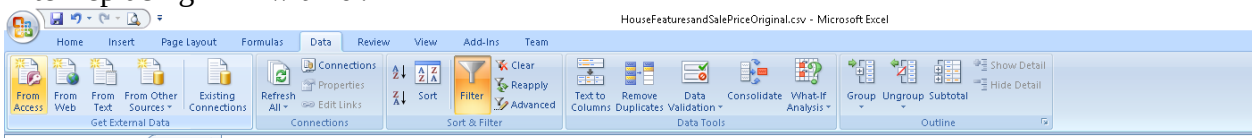
Columns LotFrontage and MasVnrArea are numeric. The creators of the data used the value 'NA' to denote no value. In these cases, I replaced NA with '0'. Cleansing is necessary to supply correct values to the machine learning algorithms.

Before cleansing the data.



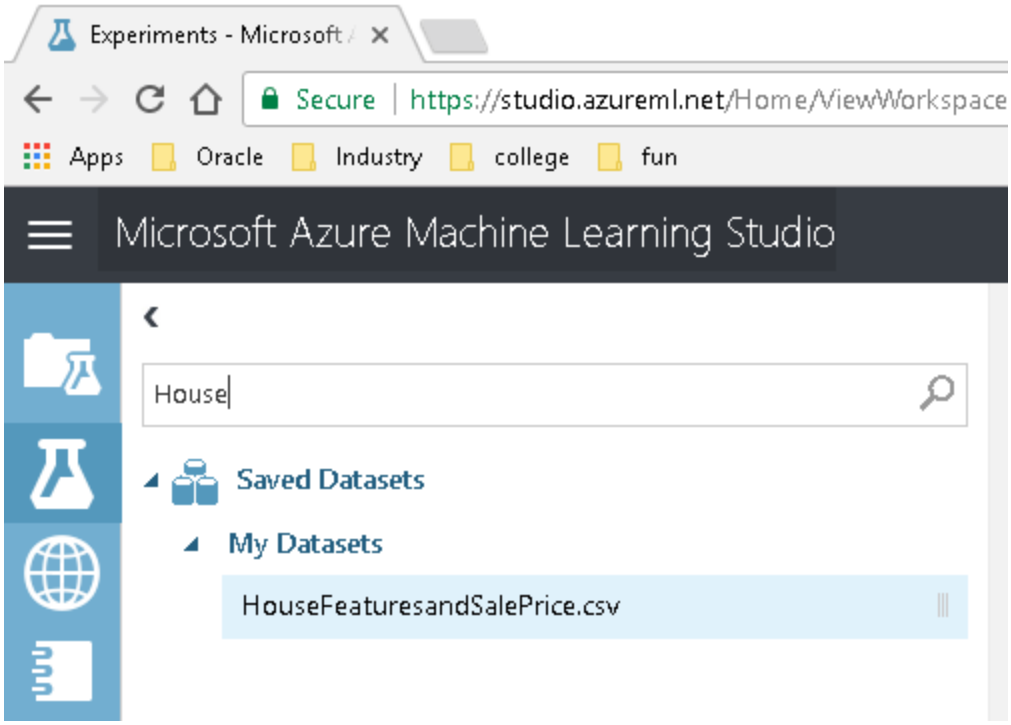
	Id	MSSubC	MSZoni	LotFronta	LotAre	Street	Exterio	MasVni	MasVni	ExterQ	ExterC	Founda	BsmtQ	BsmtC	BsmtEx	BsmtFi	BsmtFi	BsmtFi	BsmtFi	BsmtUr	TotalBs	Heat
236	235	60	RL	NA	7851	Pave	VinylSd	NA	NA	Gd	TA	PConc	Gd	TA	No	GLQ	625	Unf	0	235	860	GasA
531	530	20	RL	NA	32668	Pave	Stone	NA	NA	Gd	TA	PConc	TA	TA	No	Rec	1219	Unf	0	816	2035	GasA
652	651	60	FV	65	8125	Pave	CmentBd	NA	NA	Gd	TA	PConc	Gd	TA	No	Unf	0	Unf	0	813	813	GasA
938	937	20	RL	67	10083	Pave	VinylSd	NA	NA	Gd	TA	PConc	Gd	TA	No	GLQ	833	Unf	0	343	1176	GasA
975	974	20	FV	95	11639	Pave	CmentBd	NA	NA	Gd	TA	PConc	Gd	TA	No	Unf	0	Unf	0	1428	1428	GasA
979	978	120	FV	35	4274	Pave	VinylSd	NA	NA	Gd	TA	PConc	Gd	TA	No	GLQ	1106	Unf	0	135	1241	GasA
1245	1244	20	RL	107	13891	Pave	VinylSd	NA	NA	Ex	TA	PConc	Ex	Gd	Gd	GLQ	1386	Unf	0	690	2076	GasA
1280	1279	60	RL	75	9473	Pave	VinylSd	NA	NA	Gd	TA	PConc	Gd	TA	No	GLQ	804	Unf	0	324	1128	GasA

After replacing 'NA' with '0'.

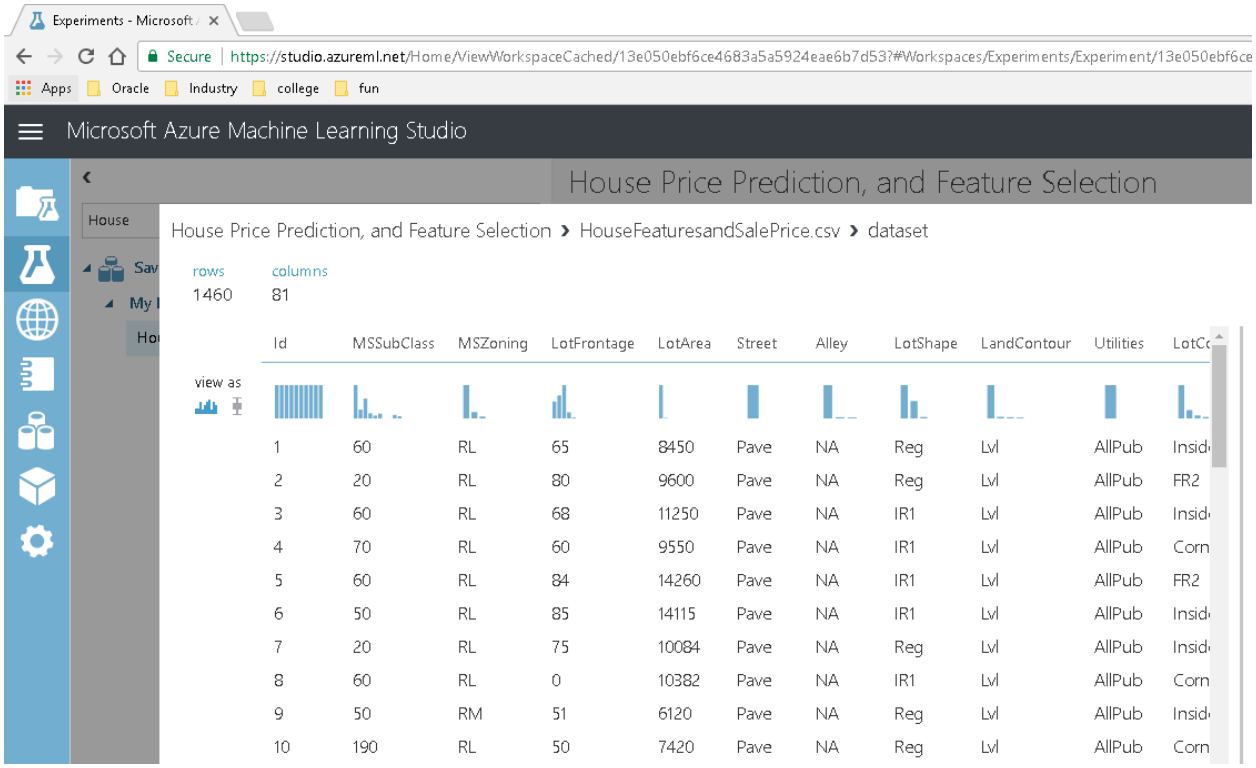


	Id	MSSubC	MSZoni	LotFronta	LotAre	Street	Exterio	MasVni	MasVni	ExterQ	ExterC	Founda	BsmtQ	BsmtC	BsmtEx	BsmtFi	BsmtFi	BsmtFi	BsmtFi	BsmtUr	TotalBs
236	235	60	RL	0	7851	Pave	VinylSd	NA	0	Gd	TA	PConc	Gd	TA	No	GLQ	625	Unf	0	235	860
531	530	20	RL	0	32668	Pave	Stone	NA	0	Gd	TA	PConc	TA	TA	No	Rec	1219	Unf	0	816	2035
652	651	60	FV	65	8125	Pave	CmentBd	NA	0	Gd	TA	PConc	Gd	TA	No	Unf	0	Unf	0	813	813
938	937	20	RL	67	10083	Pave	VinylSd	NA	0	Gd	TA	PConc	Gd	TA	No	GLQ	833	Unf	0	343	1176
975	974	20	FV	95	11639	Pave	CmentBd	NA	0	Gd	TA	PConc	Gd	TA	No	Unf	0	Unf	0	1428	1428
979	978	120	FV	35	4274	Pave	VinylSd	NA	0	Gd	TA	PConc	Gd	TA	No	GLQ	1106	Unf	0	135	1241
1245	1244	20	RL	107	13891	Pave	VinylSd	NA	0	Ex	TA	PConc	Ex	Gd	Gd	GLQ	1386	Unf	0	690	2076
1280	1279	60	RL	75	9473	Pave	VinylSd	NA	0	Gd	TA	PConc	Gd	TA	No	GLQ	804	Unf	0	324	1128

I uploaded the cleansed data to Azure Machine Learning Studio.

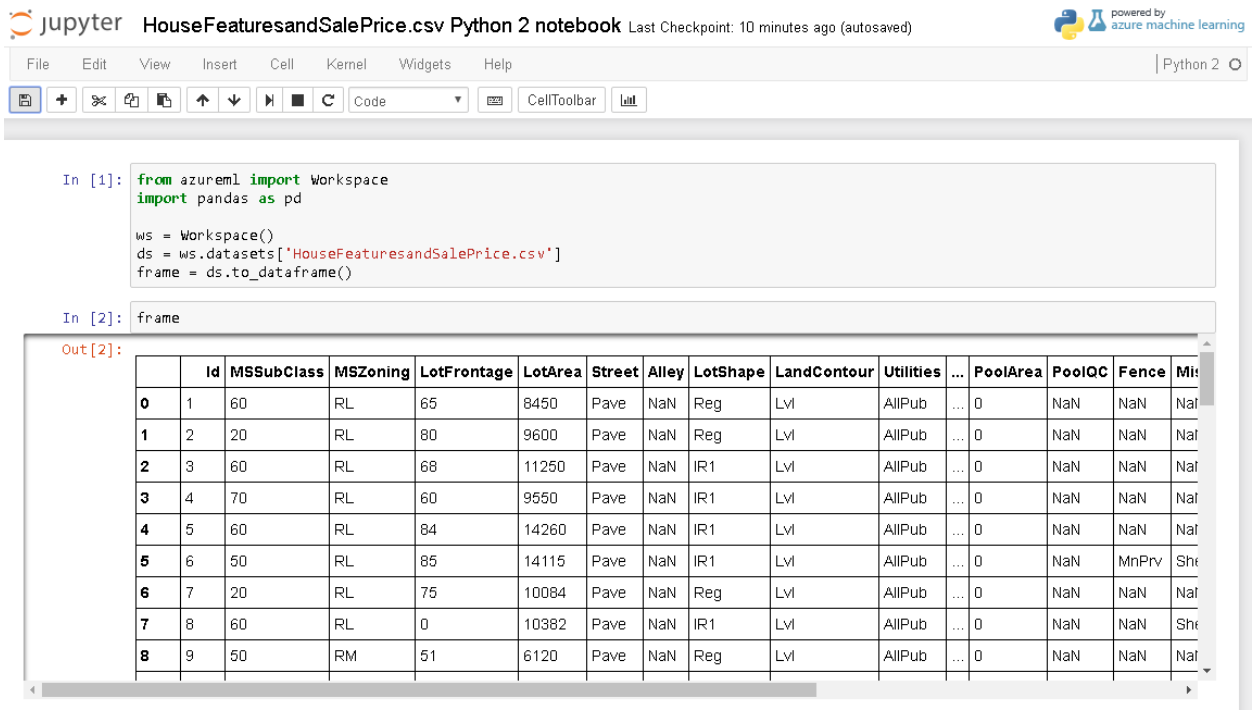


I created a new experiment, House Price Prediction, and Feature Selection, and added the dataset.



Deep Azure@McKesson Steve Klonicki

Azure Machine Learning Studio does not support the manipulation of experiments from a programming language. It does support accessing data that is the output of its various modules. I built the experiment using Studio's graphical interface but will visualize the data using Jupyter.



	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	Mis
0	1	60	RL	65	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	Nal
1	2	20	RL	80	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	Nal
2	3	60	RL	68	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	Nal
3	4	70	RL	60	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	Nal
4	5	60	RL	84	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	Nal
5	6	50	RL	85	14115	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	MnPrv	She
6	7	20	RL	75	10084	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	Nal
7	8	60	RL	0	10382	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	She
8	9	50	RM	51	6120	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	Nal

Not all 81 attributes will be useful to model SalesPrice. For example, 'Street' is 'Pave' for 1,454 entries and 'Grvl' for 6 entries. Given that 99.5% of the entries for 'Street' are the same value, it is doubtful that there is a correlation between this attribute and SalesPrice.

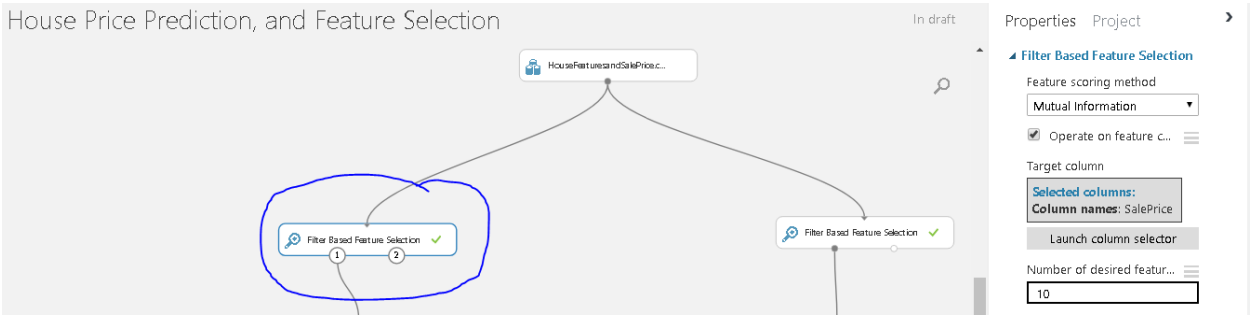
```
In [21]: frame.Street.unique()
Out[21]: array([u'Pave', u'Grvl'], dtype=object)

In [23]: frame.Street.count()
Out[23]: 1460

In [24]: frame.groupby(['Street']).size()
Out[24]: Street
         Grvl       6
         Pave    1454
         dtype: int64

In [27]: print(1454 / float(1460))
         0.995890410959
```

I decided to use the Filter Based Feature Selection module in Studio rather than identifying attributes on my own. On the left hand side, 'SalesPrice' was the target column, and I configured the module to identify the 10 most relevant attributes using the Mutual information scoring method.



From left to right, in descending order, the Filter Based Feature Selection module selected these columns that are most relevant for predicting SalesPrice.

```
In [28]: experiment = ws.experiments['13e050ebf6ce4683a5a5924eae6b7d53.f-id.bd61882e87b644c29a7de2c3f741be68']
ds = experiment.get_intermediate_dataset(
    node_id='f193f8bb-2455-4dee-a6e7-76f3803cbd91-3207',
    port_name='Results dataset',
    data_type_id='GenericCSV'
)
frame = ds.to_dataframe()
```

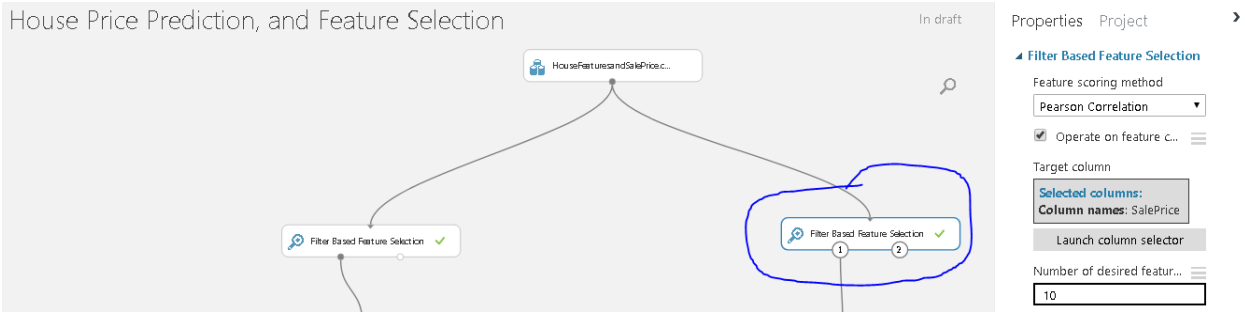
```
In [29]: frame
```

Out[29]:

	SalePrice	OverallQual	Neighborhood	GrLivArea	GarageCars	GarageArea	YearBuilt	TotalBsmtSF	BsmtQual	ExterQual	KitchenQual
0	208500	7	CollgCr	1710	2	548	2003	856	Gd	Gd	Gd
1	181500	6	Veenker	1262	2	460	1976	1262	Gd	TA	TA
2	223500	7	CollgCr	1786	2	608	2001	920	Gd	Gd	Gd
3	140000	7	Crawfor	1717	3	642	1915	756	TA	TA	Gd
4	250000	8	NoRidge	2198	3	836	2000	1145	Gd	Gd	Gd
5	143000	5	Mitchel	1362	2	480	1993	796	Gd	TA	TA
6	307000	8	Somerst	1694	2	636	2004	1686	Ex	Gd	Gd
7	200000	7	NWAmes	2090	2	484	1973	1107	Gd	TA	TA
8	129900	7	OldTown	1774	2	468	1931	952	TA	TA	TA

On the right hand side, 'SalesPrice' was the target column, and I configured the module to identify the 10 most relevant attributes using the Pearson correlation scoring method.

Deep Azure@McKesson Steve Klonicki




```
In [31]: from azureml import Workspace
ws = Workspace()
experiment = ws.experiments['13e050ebf6ce4683a5a5924eae6b7d53.f-id.bd61882e87b644c29a7de2c3f741be68']
ds = experiment.get_intermediate_dataset(
    node_id='f193f8bb-2455-4dee-a6e7-76f3803cbd91-5160',
    port_name='Results dataset',
    data_type_id='GenericCSV'
)
frame = ds.to_dataframe()
```

```
In [32]: frame
```

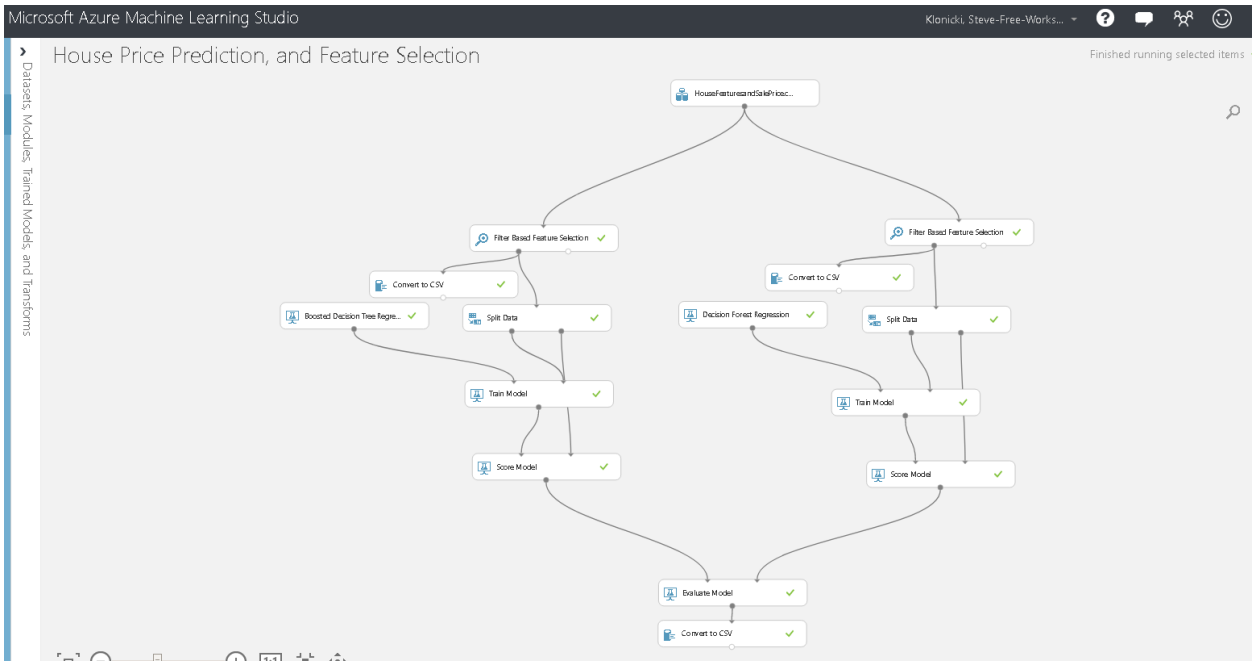
Out[32]:

	SalePrice	OverallQual	GrLivArea	GarageCars	GarageArea	TotalBsmtSF	1stFlrSF	FullBath	TotRmsAbvGrd	YearBuilt	YearRemodA
0	208500	7	1710	2	548	856	856	2	8	2003	2003
1	181500	6	1262	2	460	1262	1262	2	6	1976	1976
2	223500	7	1786	2	608	920	920	2	6	2001	2002
3	140000	7	1717	3	642	756	961	1	7	1915	1970
4	250000	8	2198	3	836	1145	1145	2	9	2000	2000
5	143000	5	1362	2	480	796	796	1	5	1993	1995
6	307000	8	1694	2	636	1686	1694	2	7	2004	2005
7	200000	7	2090	2	484	1107	1107	2	7	1973	1973
8	129900	7	1774	2	468	952	1022	2	8	1931	1950

I observed how the 2 Feature Scoring methods came to different conclusions. Listed in descending order of importance

Mutual Information	Pearson Correlation
Overall Quality	OverallQuality
Neighborhood	GrLivArea
GrLivArea	GarageCars
GarageCars	GarageArea
GarageArea	TotalBsmtSF
YearBuilt	1stFlrSF
TotalBsmtSF	FullBath
BsmtQual	TotRmsAbvGrd
ExterQual	YearBuilt
KitchenQual	YearRemodled

This is my Azure Machine Learning Studio experiment.



After loading the data, and using the Filter Based Feature Selection module to select the features to be used by the Regression algorithms. The input contained SalesPrice for all rows. For the 2 regression algorithms to be evaluated, I used the Split Data module to use 75% of the data to train the algorithm. The remaining 25% of the data was used to score the algorithm, i.e. determine SalesPrice using the trained algorithm. Finally, I evaluated the results of the 2 algorithms.

In a comparison of the 2 algorithms Boosted Decision Tree Regression outperformed Decision Forest Regression. This was determined by their Coefficient of Determination 0.858581 to 0.833426 respectively. In regression, the R^2**coefficient of determination** is a statistical measure of how well the regression line approximates the real data points. An R^2 of 1 indicates that the regression line perfectly fits the data. (cite https://en.wikipedia.org/wiki/Coefficient_of_determination)

In [34]:
```
experiment = ws.experiments['13e050ebf6ce4683a5a5924eae6b7d53.f-id.bd61882e87b644c29a7de2c3f741be68']
ds = experiment.get_intermediate_dataset(
    node_id='f193f8bb-2455-4dee-a6e7-76f3803cbd91-5169',
    port_name='Results dataset',
    data_type_id='GenericCSV'
)
frame = ds.to_dataframe()
```

In [35]:
```
frame
```

Out[35]:

	Negative Log Likelihood	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Relative Squared Error	Coefficient of Determination
0	∞	20501.911836	29997.296732	0.360987	0.141419	0.858581
1	4377.84695669526	22088.663480	32556.124367	0.388926	0.166574	0.833426

Code:

```
# coding: utf-8

# Load housing data.

from azureml import Workspace
import pandas as pd

ws = Workspace()
ds = ws.datasets['HouseFeaturesandSalePrice.csv']
frame = ds.to_dataframe()

# Print the housing data

frame

# Identify unique values for the Street attribute

frame.Street.unique()

# Print the number of rows with each unique value for the attribute Street

frame.groupby(['Street']).size()

# Calculate percentage of values that equal 'Pave'

print(1454 / float(1460))

# Print the results from the Filter Based Feature Selection, Mutual Information

experiment = ws.experiments['13e050ebf6ce4683a5a5924eae6b7d53.f-
id.bd61882e87b644c29a7de2c3f741be68']
ds = experiment.get_intermediate_dataset(
    node_id='f193f8bb-2455-4dee-a6e7-76f3803cbd91-3207',
    port_name='Results dataset',
    data_type_id='GenericCSV'
)
frame = ds.to_dataframe()

# Print the results from the Filter Based Feature Selection, Pearson Correlation

frame

# Print the results from the Filter Based Feature Selection, Pearson Correlation

from azureml import Workspace
ws = Workspace()
```

```
experiment = ws.experiments['13e050ebf6ce4683a5a5924eae6b7d53.f-
id.bd61882e87b644c29a7de2c3f741be68']
ds = experiment.get_intermediate_dataset(
    node_id='f193f8bb-2455-4dee-a6e7-76f3803cbd91-5160',
    port_name='Results dataset',
    data_type_id='GenericCSV'
)
frame = ds.to_dataframe()

# In[32]:

frame

# Print the results from the Evaluate Model module

experiment = ws.experiments['13e050ebf6ce4683a5a5924eae6b7d53.f-
id.bd61882e87b644c29a7de2c3f741be68']
ds = experiment.get_intermediate_dataset(
    node_id='f193f8bb-2455-4dee-a6e7-76f3803cbd91-5169',
    port_name='Results dataset',
    data_type_id='GenericCSV'
)
frame = ds.to_dataframe()

frame
```

Summary:
I concluded that Boosted Decision Tree Regression outperformed Decision Forest Regression.

With Microsoft Azure Machine Learning Studio, I was able to quickly generate an experiment that, in theory, could predict the price of a house with 85% accuracy. Negatives are the limited ways Azure Machine Learning can be used programmatically. It's possible to view the results of a module through Python, but it is not possible to configure or execute the module through a programming language.

If I were to continue, I would look to improve on 85% accuracy. I would research and employ additional statistical techniques such as removing outliers.

YouTube Links:
2 Min: https://youtu.be/kb44rs7QztQ
15 Min: https://youtu.be/GQZ3NAmyNxA

GitHub:
https://github.com/we814/AMLDeepDive

Deep Azure@McKesson Steve Klonicki

References:

Microsoft Azure Machine Learning Studio (https://studio.azureml.net/)
https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data
https://en.wikipedia.org/wiki/Coefficient_of_determination