Predicting 2018 Housing Prices on Kaggle

Carol Chiu

March 6, 2020

General Assembly

Data Science Immersive

Summer 2019

What is the problem?

-Company Real Estate X wants to predict housing prices for next year 2019 from 2018 housing data obtained from Kaggle.com.

-My task is to build a model that predicts housing prices accurately.

Import Housing Dataset from Kaggle.com

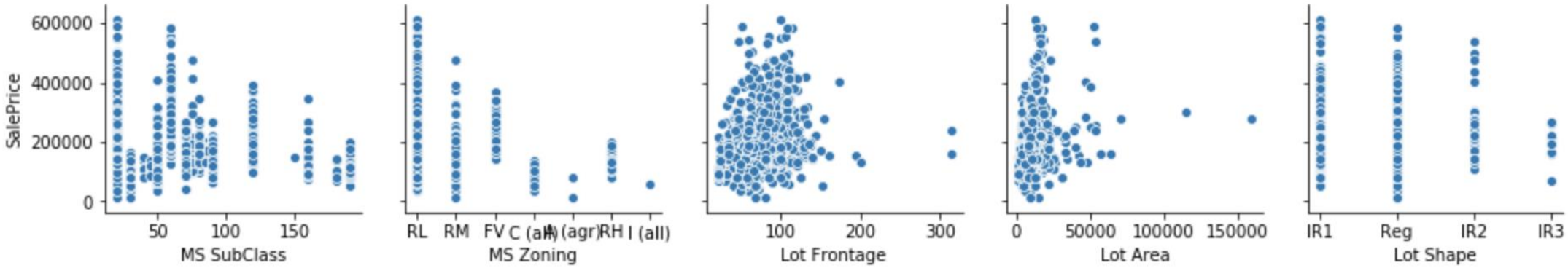
	Id	PID	MS SubClass	MS Zoning	Lot Frontage	Lot Area	Street	Alley	Lot Shape	Land Contour	...	Screen Porch	Pool Area	Pool QC	Fence	Misc Feature	Misc Val	Mo Sold	Yr Sold	Sale Type	S
0	109	533352170	60	RL	NaN	13517	Pave	NaN	IR1	Lvl	...	0	0	NaN	NaN	NaN	0	3	2010	WD	
1	544	531379050	60	RL	43.0	11492	Pave	NaN	IR1	Lvl	...	0	0	NaN	NaN	NaN	0	4	2009	WD	
2	153	535304180	20	RL	68.0	7922	Pave	NaN	Reg	Lvl	...	0	0	NaN	NaN	NaN	0	1	2010	WD	
3	318	916386060	60	RL	73.0	9802	Pave	NaN	Reg	Lvl	...	0	0	NaN	NaN	NaN	0	4	2010	WD	
4	255	906425045	50	RL	82.0	14235	Pave	NaN	IR1	Lvl	...	0	0	NaN	NaN	NaN	0	3	2010	WD	

Average Sales Price for Houses

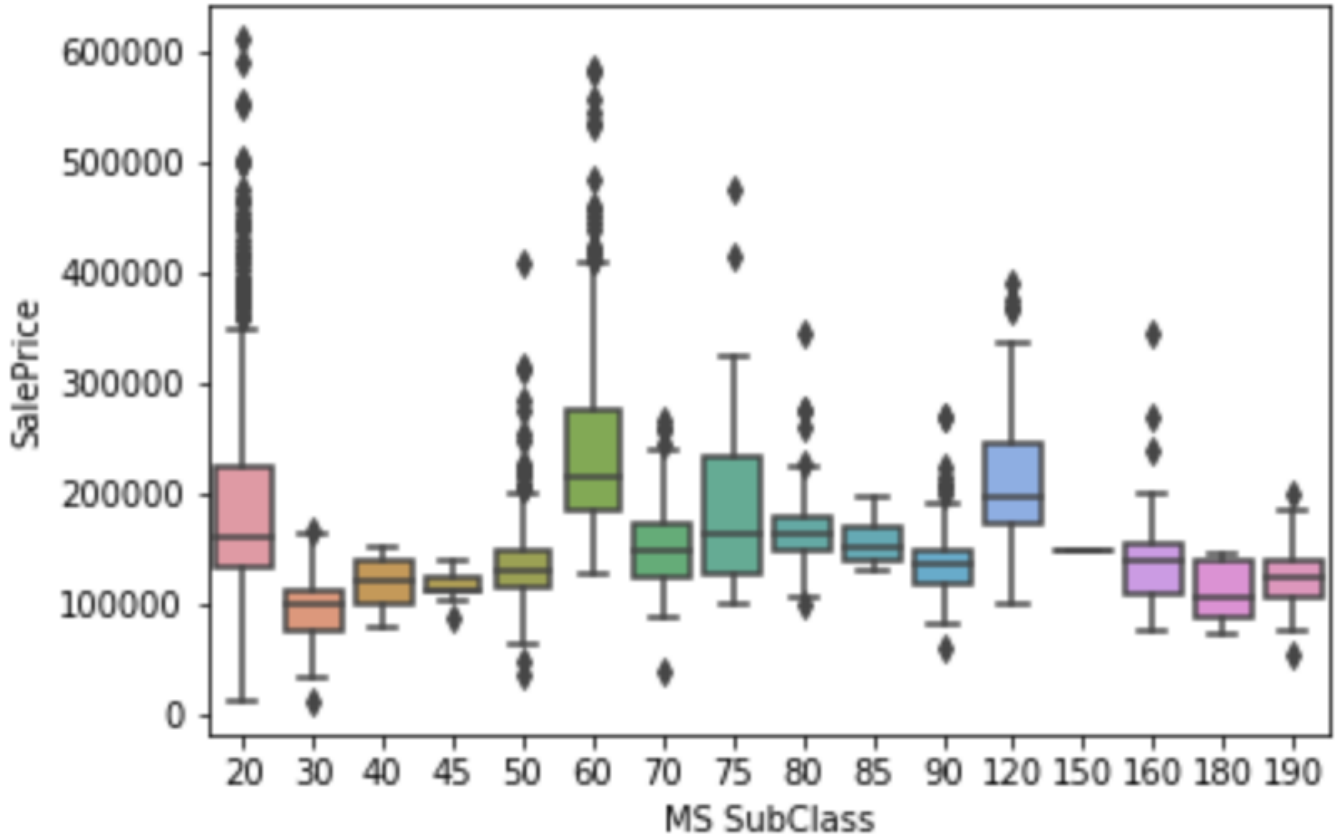
```
import statistics
statistics.mean(train["SalePrice"])
```

```
181469.70160897123
```

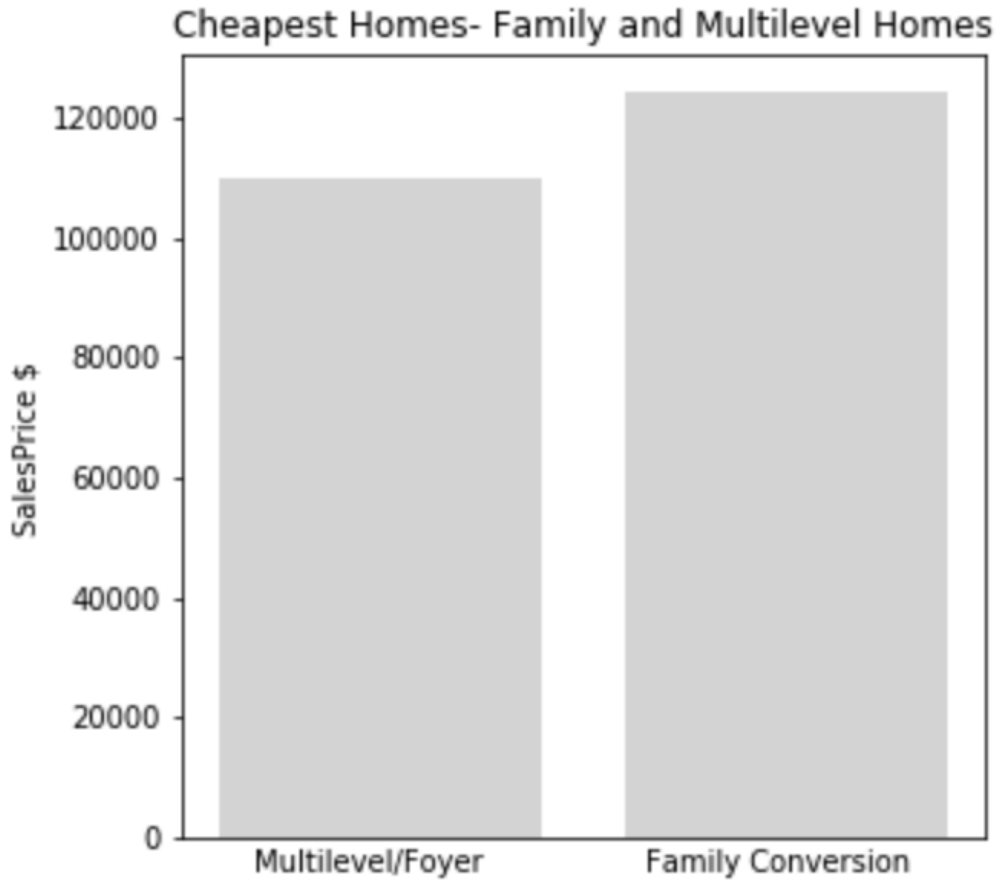
Visualize relationship between features



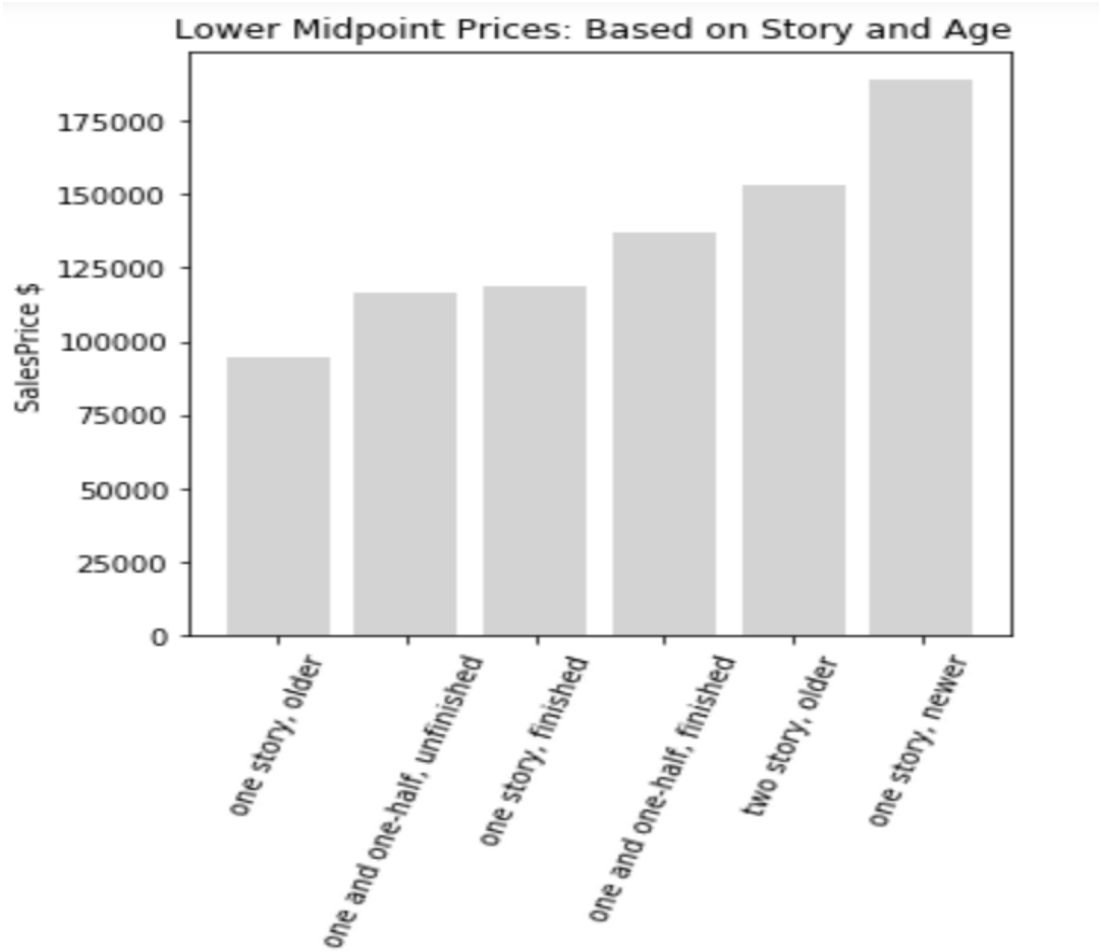
Relationship between Type of House and Sales price



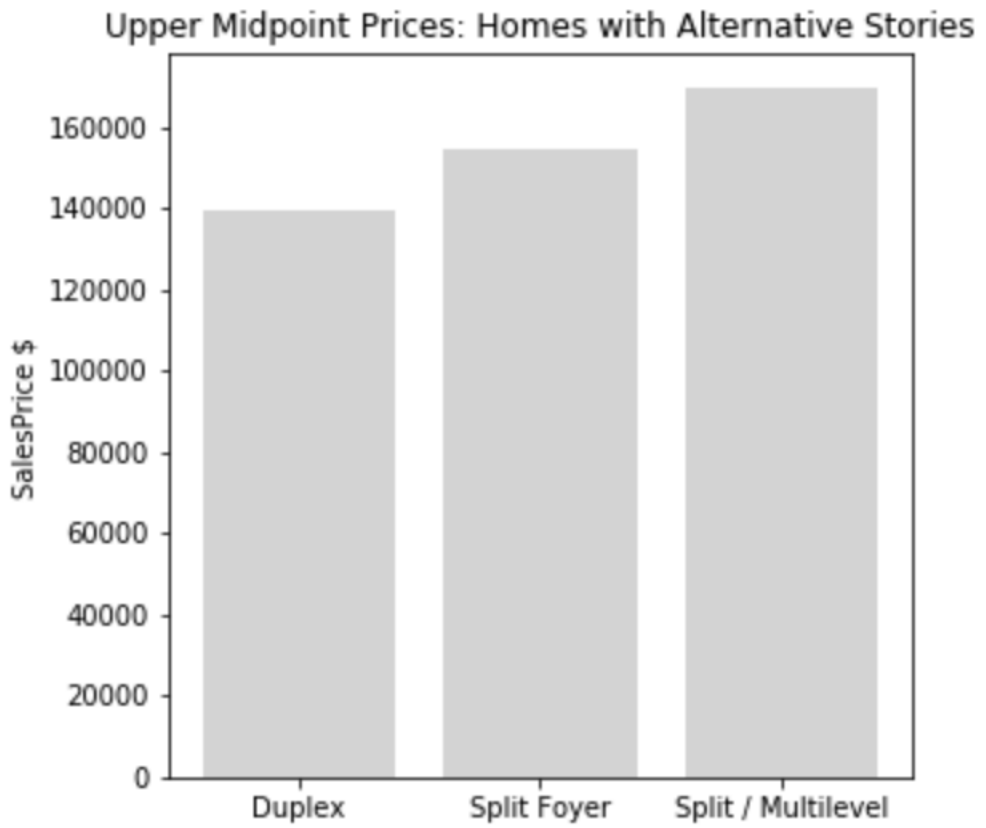
Cheapest Homes ~ 110,000 Dollars



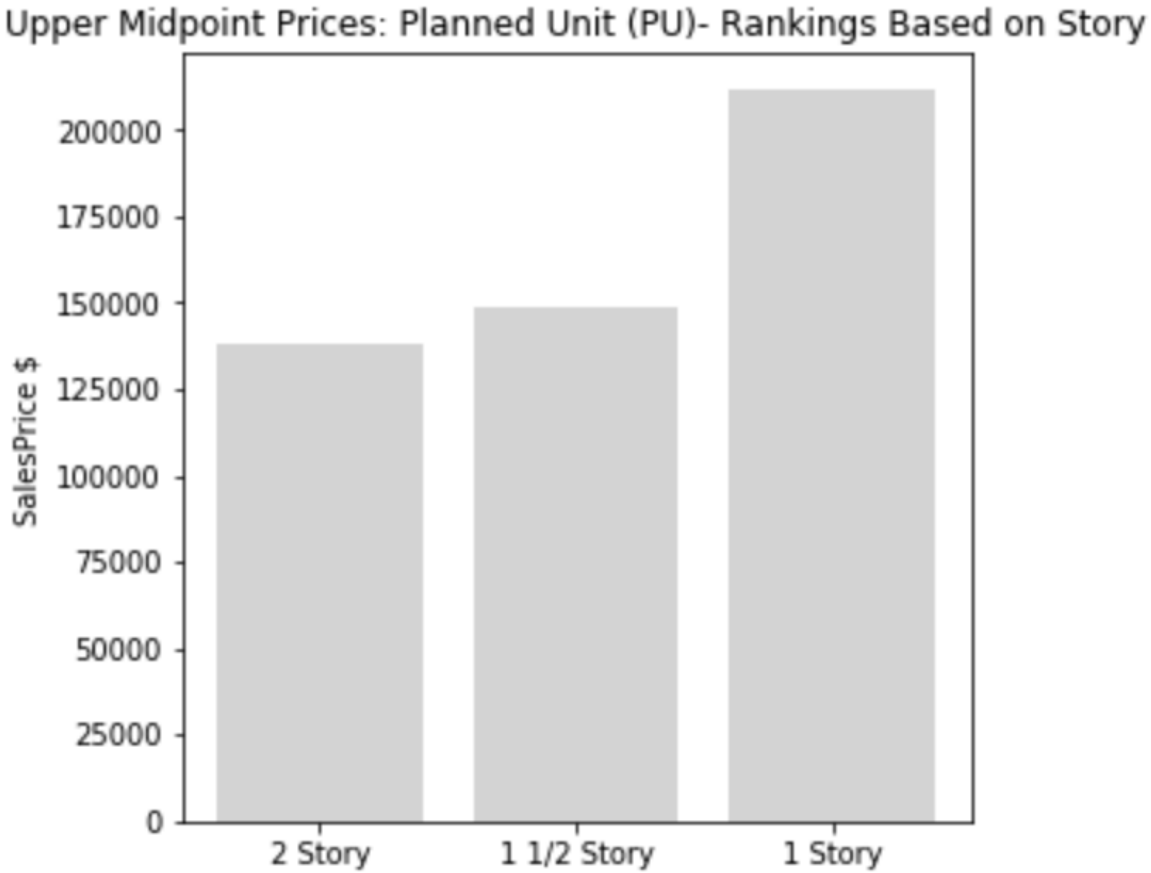
Lower Midpoint Prices ~ 125,000 Dollars



Midpoint Prices ~ 150,000 Dollars

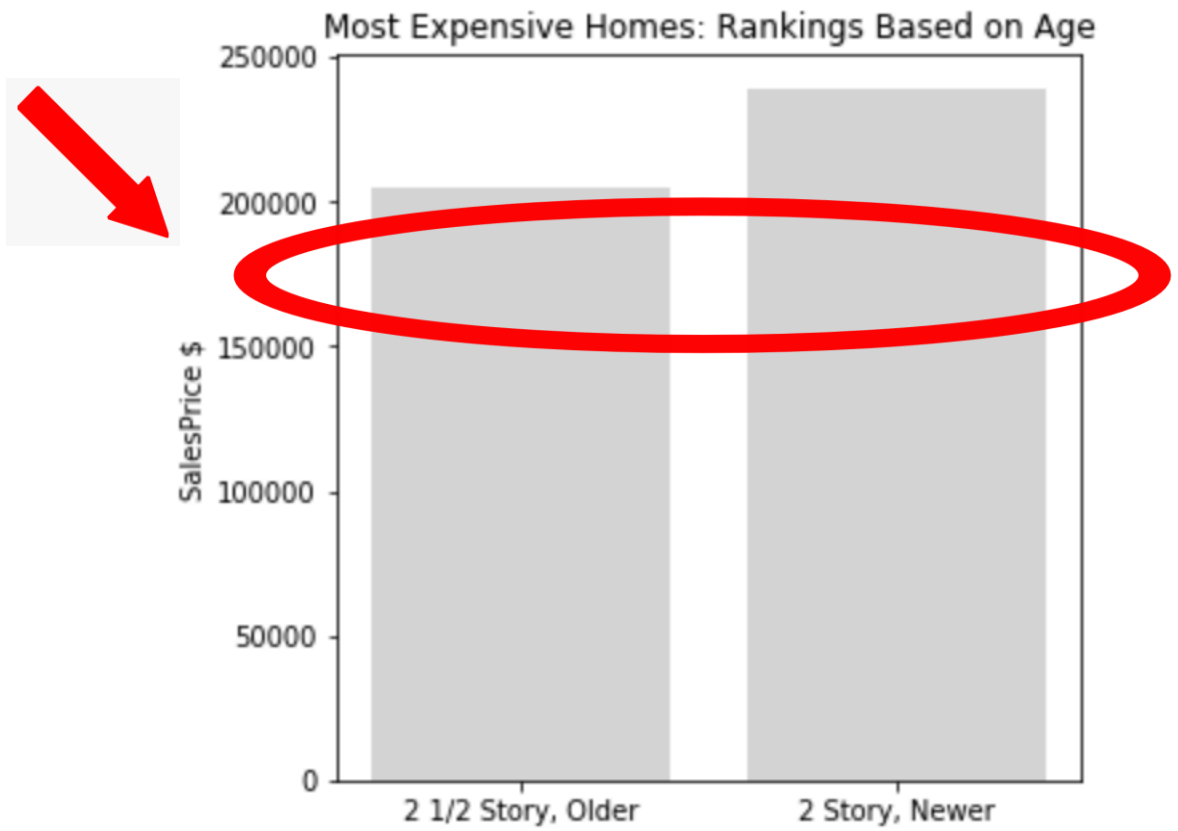


Upper Midpoint Prices ~ 160,000 Dollars



Most Expensive Prices ~ 225,000 Dollars

Mean
~ $180K



Run Linear Model

R-squared value on train set: 0.246

R-squared value on test set: 0.231

-Data exhibits non-linear behavior, bad r-squared score on test set

Run LassoCV Regression

```
# Instantiate linear regression model without regularization.
lr_model = LassoCV(alphas=np.logspace(0,10,100))

# Fit linear regression model.
lr_model.fit(X_train, y_train)

# Score
lr_model.score(X_train, y_train), lr_model.score(X_test, y_test)
```

R-squared on train: 0.244

R-squared on test: 0.233

Run Ridge Model

```
# Instantiate.
ridge_model = Ridge(alpha=100)

# Fit.
ridge_model.fit(X_train, y_train)

# Generate predictions.
ridge_preds = ridge_model.predict(X_test)
ridge_preds_train = ridge_model.predict(X_train)

# Evaluate model using R2.
print(r2_score(y_train, ridge_preds_train))
print(r2_score(y_test, ridge_preds))
```

R-squared on train set: 0.244

R-squared on test set: 0.233

Run model Using Logistic Regression

```
# Step 1: Instantiate our model.
logreg = LogisticRegression()

# Step 2: Fit our model.
logreg.fit(X_train, y_train)
```

R-squared on train set: 0.520

R-squared on test set: 0.215

Results/Summary

-The models have low predictive accuracy.

-I need to try to find a pattern in the features so that I can put the right features into the train-test split.

References

1) “Project 2 Predicting Housing Prices Using Logistic Regression”, *General Assembly Data Science Intensive*, Atlanta, Georgia, March 6, 2020

2) Gupta, Prashant, “Regularization in Machine Learning, “ *towards Data Science*, Nov. 15, 2017, https://towardsdatascience.com/regularization-in-machine-learning-76441ddcf99a, accessed March 6, 2020