In [35]:
```
import tensorflow as tf
import tensorflow_decision_forests as tfdf
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Comment this if the data visualisations doesn't work on your side
%matplotlib inline
```

In [3]:
```
train_file_path = "train.csv"
dataset_df = pd.read_csv(train_file_path)
print("Full train dataset shape is {}".format(dataset_df.shape))
```

```
Full train dataset shape is (1460, 81)
```

In [4]:
```
dataset_df.head(10)
```

Out[4]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandCo
0	1	60	RL	65.0	8450	Pave	NaN	Reg	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	
5	6	50	RL	85.0	14115	Pave	NaN	IR1	
6	7	20	RL	75.0	10084	Pave	NaN	Reg	
7	8	60	RL	NaN	10382	Pave	NaN	IR1	
8	9	50	RM	51.0	6120	Pave	NaN	Reg	
9	10	190	RL	50.0	7420	Pave	NaN	Reg	

10 rows × 81 columns

```
Warning: Total number of columns (81) exceeds max_columns (20) limiting to first
(20) columns.
```

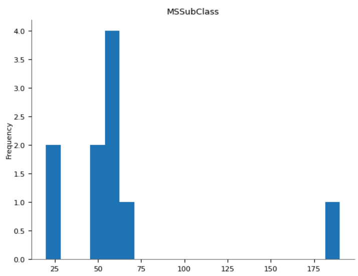
In [5]:
```
dataset_df = dataset_df.drop('Id', axis=1)
dataset_df.head(10)
```

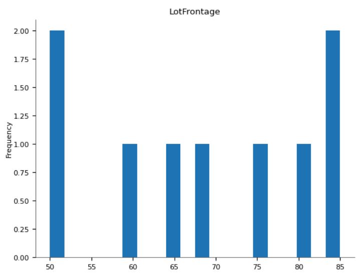
Out[5]:

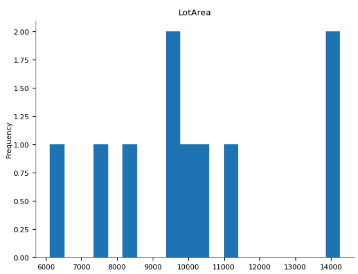
	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContou
0	60	RL	65.0	8450	Pave	NaN	Reg	Lv
1	20	RL	80.0	9600	Pave	NaN	Reg	Lv
2	60	RL	68.0	11250	Pave	NaN	IR1	Lv
3	70	RL	60.0	9550	Pave	NaN	IR1	Lv
4	60	RL	84.0	14260	Pave	NaN	IR1	Lv
5	50	RL	85.0	14115	Pave	NaN	IR1	Lv
6	20	RL	75.0	10084	Pave	NaN	Reg	Lv
7	60	RL	NaN	10382	Pave	NaN	IR1	Lv
8	50	RM	51.0	6120	Pave	NaN	Reg	Lv
9	190	RL	50.0	7420	Pave	NaN	Reg	Lv

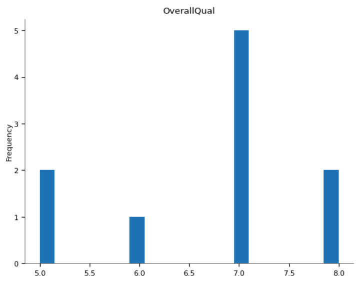
10 rows × 80 columns

Distributions

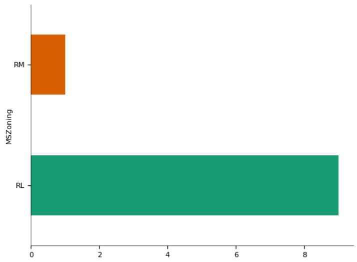


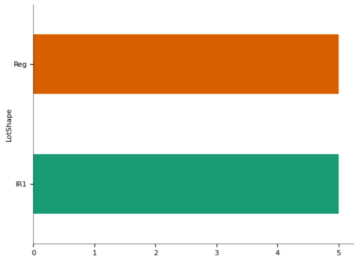


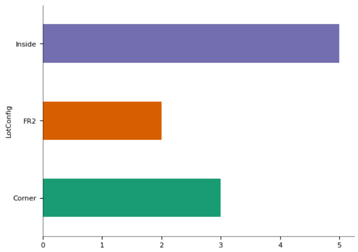


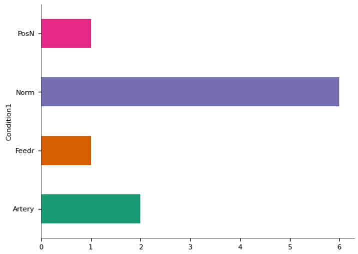


Categorical distributions

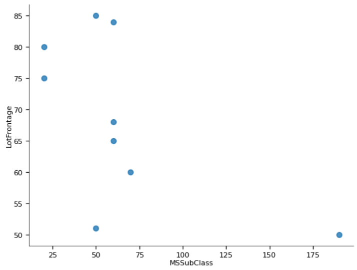


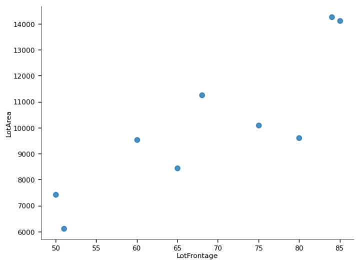


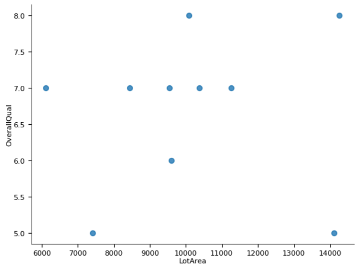


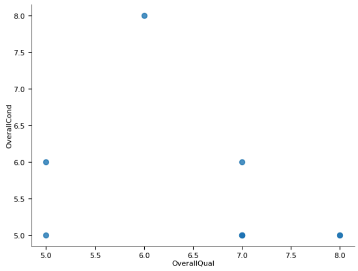


2-d distributions

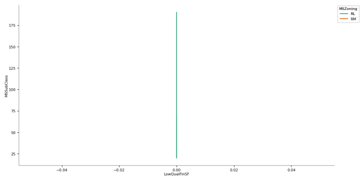


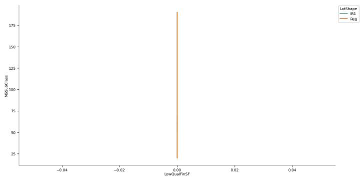


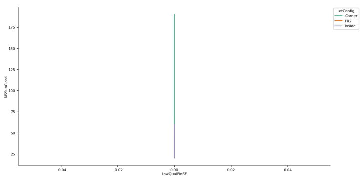


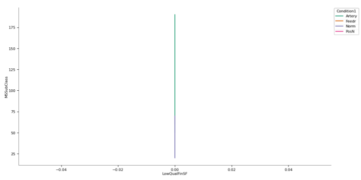


Time series

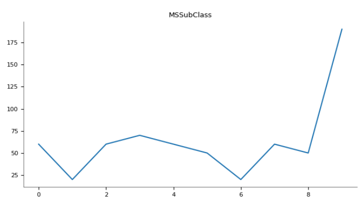


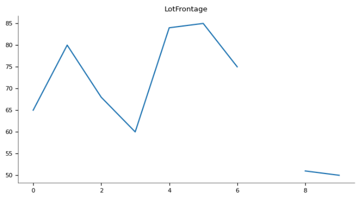


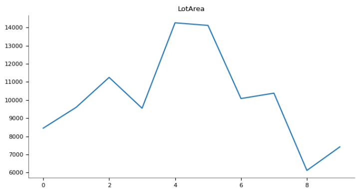


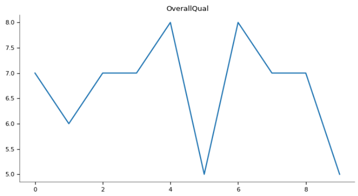


Values

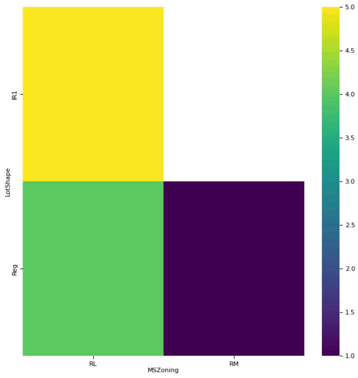


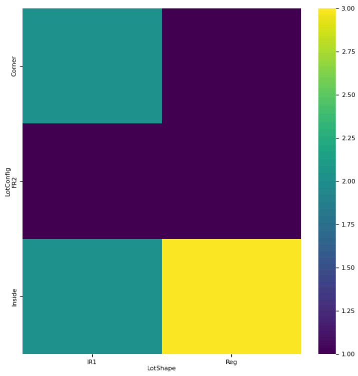


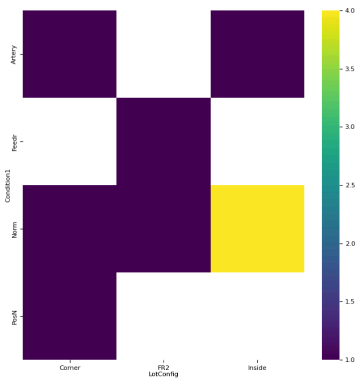


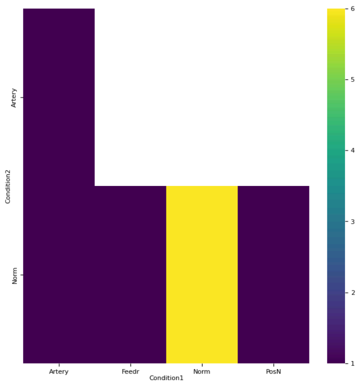


2-d categorical distributions



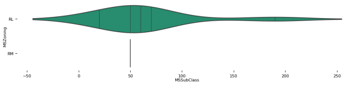




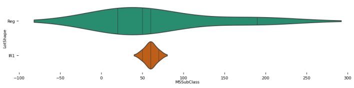


Faceted distributions

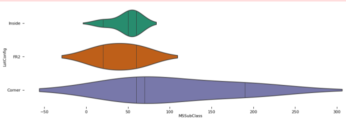
```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effe
ct.
```



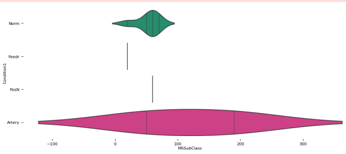
```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effe
ct.
```



```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effe
ct.
```



```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effe
ct.
```



In [6]:
```
dataset_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 80 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   MSSubClass     1460 non-null   int64  
 1   MSZoning       1460 non-null   object 
 2   LotFrontage    1201 non-null   float64
 3   LotArea        1460 non-null   int64  
 4   Street         1460 non-null   object 
 5   Alley          91 non-null     object 
 6   LotShape       1460 non-null   object 
 7   LandContour    1460 non-null   object 
 8   Utilities      1460 non-null   object 
 9   LotConfig      1460 non-null   object 
 10  LandSlope      1460 non-null   object 
 11  Neighborhood   1460 non-null   object 
 12  Condition1     1460 non-null   object 
 13  Condition2     1460 non-null   object 
 14  BldgType       1460 non-null   object 
 15  HouseStyle     1460 non-null   object 
 16  OverallQual    1460 non-null   int64  
 17  OverallCond    1460 non-null   int64  
 18  YearBuilt      1460 non-null   int64  
 19  YearRemodAdd   1460 non-null   int64  
 20  RoofStyle      1460 non-null   object 
 21  RoofMatl       1460 non-null   object 
 22  Exterior1st    1460 non-null   object 
 23  Exterior2nd    1460 non-null   object 
 24  MasVnrType     588 non-null    object 
 25  MasVnrArea     1452 non-null   float64
 26  ExterQual      1460 non-null   object 
 27  ExterCond      1460 non-null   object 
 28  Foundation     1460 non-null   object 
 29  BsmtQual       1423 non-null   object 
 30  BsmtCond       1423 non-null   object 
 31  BsmtExposure   1422 non-null   object 
 32  BsmtFinType1   1423 non-null   object 
 33  BsmtFinSF1     1460 non-null   int64  
 34  BsmtFinType2   1422 non-null   object 
 35  BsmtFinSF2     1460 non-null   int64  
 36  BsmtUnfSF      1460 non-null   int64  
 37  TotalBsmtSF    1460 non-null   int64  
 38  Heating        1460 non-null   object 
 39  HeatingQC      1460 non-null   object 
 40  CentralAir     1460 non-null   object 
 41  Electrical     1459 non-null   object 
 42  1stFlrSF       1460 non-null   int64  
 43  2ndFlrSF       1460 non-null   int64  
 44  LowQualFinSF   1460 non-null   int64  
 45  GrLivArea      1460 non-null   int64  
 46  BsmtFullBath   1460 non-null   int64  
 47  BsmtHalfBath   1460 non-null   int64  
 48  FullBath       1460 non-null   int64  
 49  HalfBath       1460 non-null   int64  
 50  BedroomAbvGr   1460 non-null   int64  
 51  KitchenAbvGr   1460 non-null   int64  
 52  KitchenQual    1460 non-null   object 
 53  TotRmsAbvGrd   1460 non-null   int64  
 54  Functional     1460 non-null   object 
```

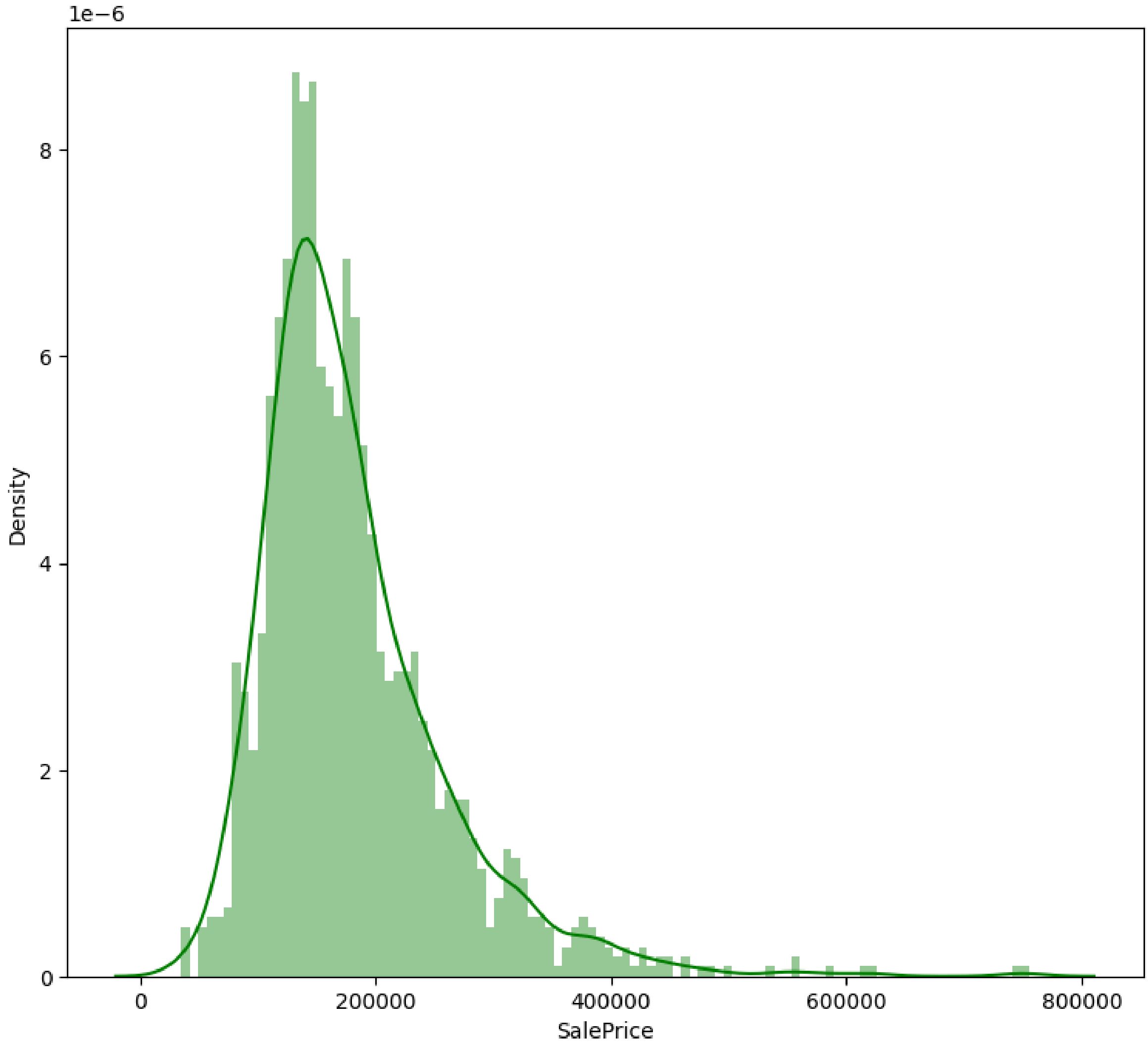
```
 55  Fireplaces     1460 non-null   int64
 56  FireplaceQu    770 non-null    object
 57  GarageType     1379 non-null   object
 58  GarageYrBlt    1379 non-null   float64
 59  GarageFinish   1379 non-null   object
 60  GarageCars     1460 non-null   int64
 61  GarageArea     1460 non-null   int64
 62  GarageQual     1379 non-null   object
 63  GarageCond     1379 non-null   object
 64  PavedDrive     1460 non-null   object
 65  WoodDeckSF     1460 non-null   int64
 66  OpenPorchSF    1460 non-null   int64
 67  EnclosedPorch  1460 non-null   int64
 68  3SsnPorch      1460 non-null   int64
 69  ScreenPorch    1460 non-null   int64
 70  PoolArea       1460 non-null   int64
 71  PoolQC         7 non-null      object
 72  Fence          281 non-null    object
 73  MiscFeature    54 non-null     object
 74  MiscVal        1460 non-null   int64
 75  MoSold         1460 non-null   int64
 76  YrSold         1460 non-null   int64
 77  SaleType       1460 non-null   object
 78  SaleCondition  1460 non-null   object
 79  SalePrice      1460 non-null   int64
dtypes: float64(3), int64(34), object(43)
memory usage: 912.6+ KB
```

In [7]:

```
print(dataset_df['SalePrice'].describe())
plt.figure(figsize=(9, 8))
sns.distplot(dataset_df['SalePrice'], color='g', bins=100, hist_kws={'alpha': 0.
```

```
count      1460.000000
mean     180921.195890
std       79442.502883
min       34900.000000
25%      129975.000000
50%      163000.000000
75%      214000.000000
max      755000.000000
Name: SalePrice, dtype: float64
```

```
<ipython-input-7-dc911a47893e>:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(dataset_df['SalePrice'], color='g', bins=100, hist_kws={'alpha':
0.4});
```



In [8]:
```
list(set(dataset_df.dtypes.tolist()))
```

Out[8]: [dtype('float64'), dtype('int64'), dtype('O')]

In [9]:
```
df_num = dataset_df.select_dtypes(include = ['float64', 'int64'])
df_num.head()
```

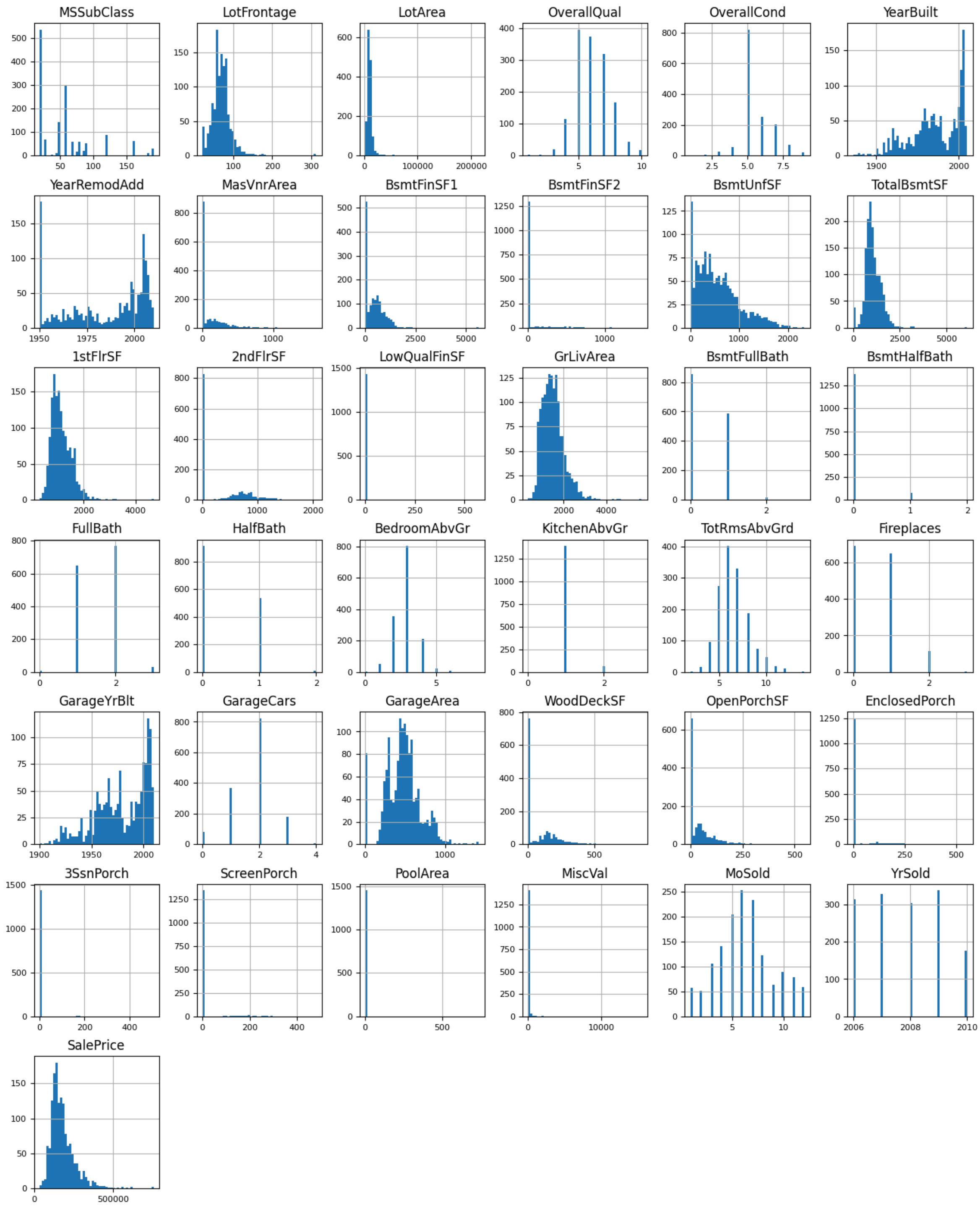
Out[9]:

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodA
0	60	65.0	8450	7	5	2003	2
1	20	80.0	9600	6	8	1976	1
2	60	68.0	11250	7	5	2001	2
3	70	60.0	9550	7	5	1915	1
4	60	84.0	14260	8	5	2000	2

5 rows × 37 columns

```
Warning: Total number of columns (37) exceeds max_columns (20) limiting to first
(20) columns.
```

In [10]:
```
df_num.hist(figsize=(16, 20), bins=50, xlabelsize=8, ylabelsize=8);
```



In [11]:
```
import numpy as np

def split_dataset(dataset, test_ratio=0.30):
  test_indices = np.random.rand(len(dataset)) < test_ratio
  return dataset[~test_indices], dataset[test_indices]

train_ds_pd, valid_ds_pd = split_dataset(dataset_df)
print("{} examples in training, {} examples in testing.".format(
    len(train_ds_pd), len(valid_ds_pd)))
```

```
1017 examples in training, 443 examples in testing.
```

In [12]:
```
label = 'SalePrice'
train_ds = tfdf.keras.pd_dataframe_to_tf_dataset(train_ds_pd, label=label, task
valid_ds = tfdf.keras.pd_dataframe_to_tf_dataset(valid_ds_pd, label=label, task
```

In [13]:
```
tfdf.keras.get_all_models()
```

Out[13]:
```
[tensorflow_decision_forests.keras.RandomForestModel,
 tensorflow_decision_forests.keras.GradientBoostedTreesModel,
 tensorflow_decision_forests.keras.CartModel,
 tensorflow_decision_forests.keras.DistributedGradientBoostedTreesModel]
```

In [14]:
```
rf = tfdf.keras.RandomForestModel(task = tfdf.keras.Task.REGRESSION)
rf.compile(metrics=["mse"])
```

```
Use /tmp/tmp90d7frb1 as temporary training directory
```

In [15]:
```
rf.fit(x=train_ds)
```

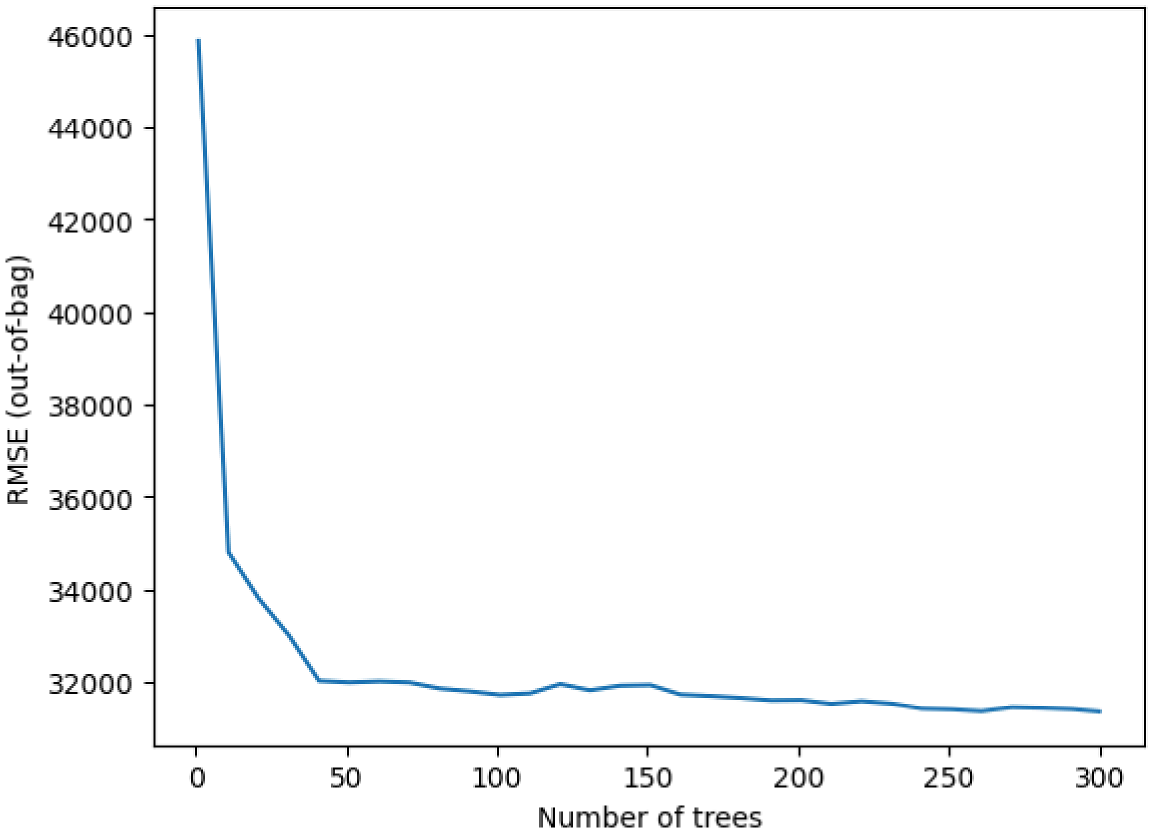
```
Reading training dataset...
Training dataset read in 0:00:10.971011. Found 1017 examples.
Training model...
Model trained in 0:00:03.811701
Compiling model...
Model compiled.
```

Out[15]:
```
<tf_keras.src.callbacks.History at 0x7aaca12398a0>
```

In [16]:
```
tfdf.model_plotter.plot_model_in_colab(rf, tree_idx=0, max_depth=3)
```

Out[16]:

In [17]:
```
import matplotlib.pyplot as plt
logs = rf.make_inspector().training_logs()
plt.plot([log.num_trees for log in logs], [log.evaluation.rmse for log in logs])
plt.xlabel("Number of trees")
plt.ylabel("RMSE (out-of-bag)")
plt.show()
```



In [18]:
```
inspector = rf.make_inspector()
inspector.evaluation()
```

Out[18]:
```
Evaluation(num_examples=1017, accuracy=None, loss=None, rmse=31358.87495781327,
ndcg=None, aucs=None, auuc=None, qini=None)
```

In [19]:
```
evaluation = rf.evaluate(x=valid_ds,return_dict=True)

for name, value in evaluation.items():
  print(f"{name}: {value:.4f}")
```

```
1/1 [==============================] - 9s 9s/step - loss: 0.0000e+00 - mse: 55074
5088.0000
loss: 0.0000
mse: 550745088.0000
```

In [20]:
```
print(f"Available variable importances:")
for importance in inspector.variable_importances().keys():
  print("\t", importance)
```

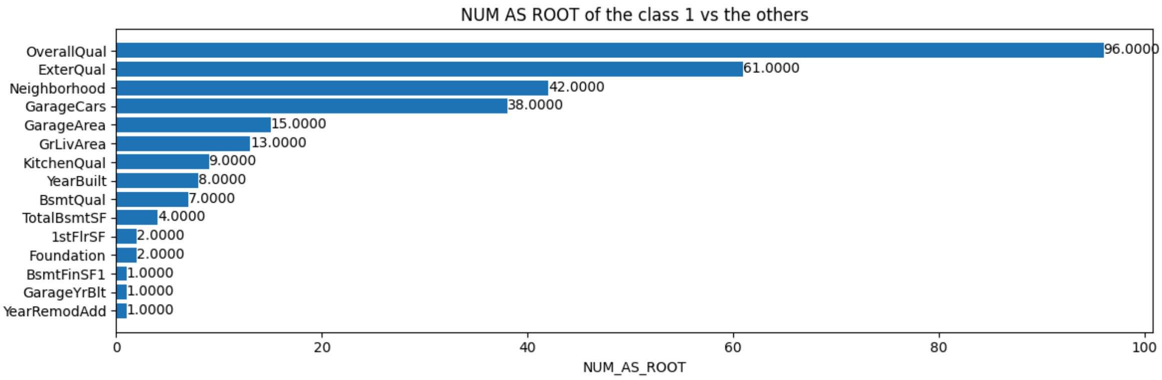
```
Available variable importances:
	 SUM_SCORE
	 NUM_AS_ROOT
	 INV_MEAN_MIN_DEPTH
	 NUM_NODES
```

In [21]:
```
inspector.variable_importances()["NUM_AS_ROOT"]
```

Out[21]:
```
[("OverallQual" (1; #62), 96.0),
 ("ExterQual" (4; #22), 61.0),
 ("Neighborhood" (4; #59), 42.0),
 ("GarageCars" (1; #32), 38.0),
 ("GarageArea" (1; #31), 15.0),
 ("GrLivArea" (1; #38), 13.0),
 ("KitchenQual" (4; #44), 9.0),
 ("YearBuilt" (1; #76), 8.0),
 ("BsmtQual" (4; #14), 7.0),
 ("TotalBsmtSF" (1; #73), 4.0),
 ("1stFlrSF" (1; #0), 2.0),
 ("Foundation" (4; #28), 2.0),
 ("BsmtFinSF1" (1; #8), 1.0),
 ("GarageYrBlt" (1; #37), 1.0),
 ("YearRemodAdd" (1; #77), 1.0)]
```

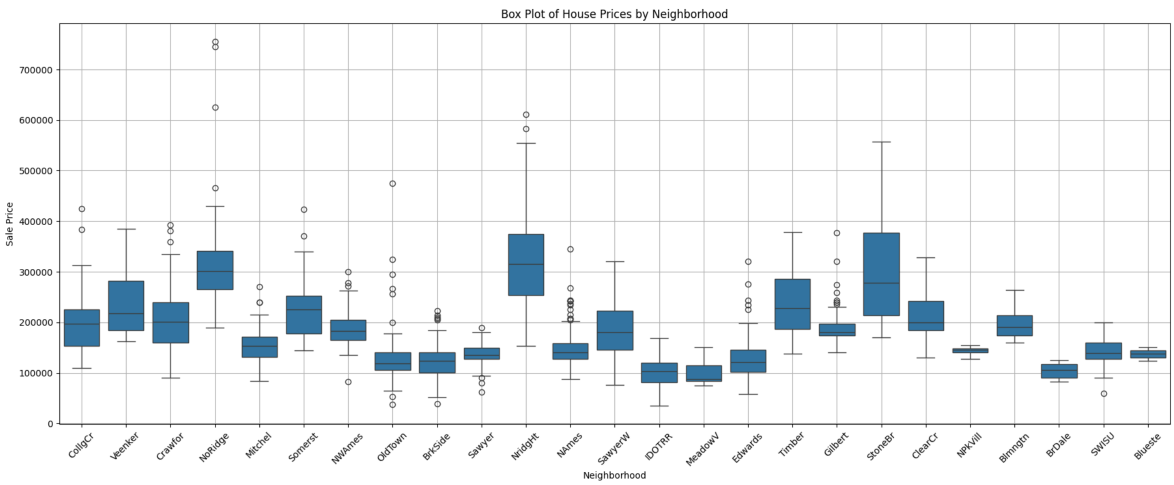
In [22]:
```
plt.figure(figsize=(12, 4))

# Mean decrease in AUC of the class 1 vs the others.
variable_importance_metric = "NUM_AS_ROOT"
variable_importances = inspector.variable_importances()[variable_importance_metr

# Extract the feature name and importance values.
#
# `variable_importances` is a list of <feature, importance> tuples.
feature_names = [vi[0].name for vi in variable_importances]
feature_importances = [vi[1] for vi in variable_importances]
# The feature are ordered in decreasing importance value.
feature_ranks = range(len(feature_names))

bar = plt.barh(feature_ranks, feature_importances, label=[str(x) for x in featur
plt.yticks(feature_ranks, feature_names)
```

```
plt.gca().invert_yaxis()

# TODO: Replace with "plt.bar_label()" when available.
# Label each bar with values
for importance, patch in zip(feature_importances, bar.patches):
  plt.text(patch.get_x() + patch.get_width(), patch.get_y(), f"{importance:.4f}"

plt.xlabel(variable_importance_metric)
plt.title("NUM AS ROOT of the class 1 vs the others")
plt.tight_layout()
plt.show()
```



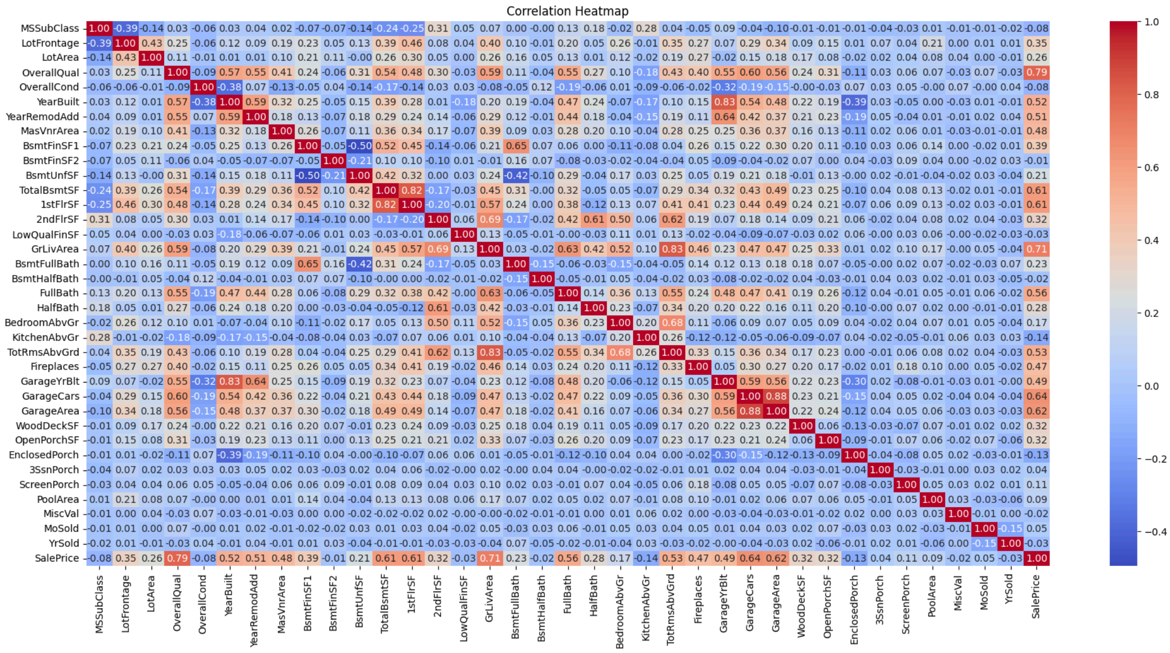
In [24]:

```
plt.figure(figsize=(22, 8))
sns.boxplot(x='Neighborhood', y='SalePrice', data=dataset_df)
plt.xticks(rotation=45)
plt.title('Box Plot of House Prices by Neighborhood')
plt.xlabel('Neighborhood')
plt.ylabel('Sale Price')
plt.grid(True)
plt.show()
```



In [27]:

```
numerical_cols = dataset_df.select_dtypes(include=['number']).columns
numerical_df = dataset_df[numerical_cols]

# Calculate the correlation matrix
correlation_matrix = numerical_df.corr()

# Plot the heatmap
plt.figure(figsize=(22, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap')
plt.show()
```



In [28]:
```python
test_file_path = "test.csv"
data_df = pd.read_csv(test_file_path)
print("Full train dataset shape is {}".format(data_df.shape))
```

```
Full train dataset shape is (1459, 80)
```

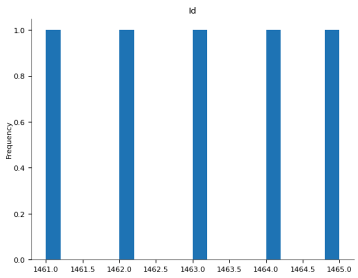
In [31]:
```python
test_file_path = "test.csv"
test_data = pd.read_csv(test_file_path)
ids = test_data.pop('Id')

test_ds = tfdf.keras.pd_dataframe_to_tf_dataset(
    test_data,
    task = tfdf.keras.Task.REGRESSION)

preds = rf.predict(test_ds)
output = pd.DataFrame({'Id': ids,
                       'SalePrice': preds.squeeze()})

output.head()
```

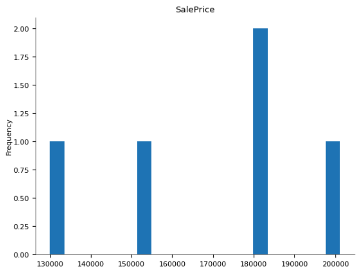
```
2/2 [==============================] - 3s 70ms/step
```

Out[31]:

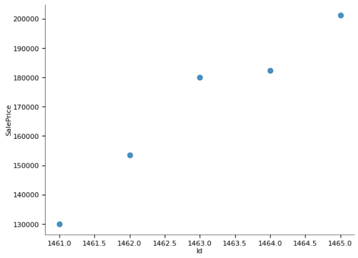
	Id	SalePrice
0	1461	129972.406250
1	1462	153419.515625
2	1463	180060.406250
3	1464	182446.125000
4	1465	201203.093750

Distributions

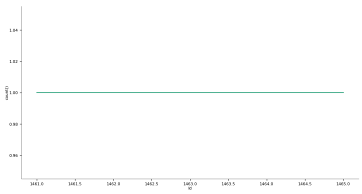


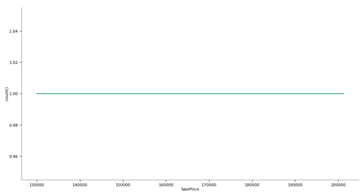


2-d distributions

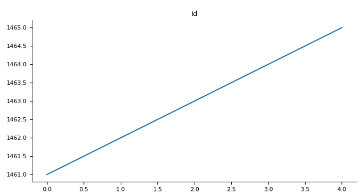


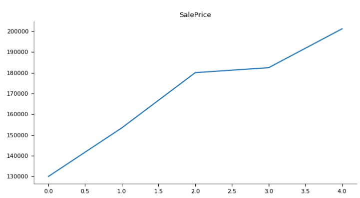
Time series





Values





In [34]:
```
sample_submission_df = pd.read_csv('sample_submission.csv')
sample_submission_df['SalePrice'] = rf.predict(test_ds)
```

```
sample_submission_df.to_csv('submission.csv', index=False)
```

```
2/2 [==============================] - 0s 28ms/step
```

In [ ]: