CT5133 / CT5145 Deep Learning (/Online) 2022-2023

Assignment 2

James McDermott

- Student ID(s): **Jash Prakash Rana**
- Student name(s): **22222806**

Due date: midnight Sunday 19 March (end Week 10).

Weighting: 20% of the module.

In this assignment the goal is to take advantage of pre-trained NN models to create an embedding with a dataset of movie posters, and demonstrate how to use that embedding.

The dataset is provided, along with some skeleton code for loading it.

The individual steps to be carried out are specified below, with `### YOUR CODE HERE` markers, together with the number of marks available for each part.

- **Topics**: in Part 5 below, students are asked to add some improvement to their models. In general, these improvements will differ between students (or student groups). **The proposed improvement must be notified to the lecturer at least 1 week before submission, and approved by the lecturer**. If working in a group, the two members of the group should not work on different topics in Part 5: they must work on the same topic and submit identical submissions.

- Students are not required to work incrementally on the parts. It is ok to do all the work in one day, so long as you abide by the rules on notifying groups and notifying topics.

- **Groups**: students may work solo or in a group of two. A student may not work together in a group with any student they have previously worked on a group project with, in this module or any other in the MSc programme. **Groups must be notified to the lecturer in writing before beginning work and at least 1 week before submission.** If working in a group, both students must submit and both submissions must be identical. If working in a group, both students may be asked to explain any aspect of the code in interview (see below), therefore working independently on separate components is not recommended. Any emails concerning the project should be cc-ed to the other group member.

- **Libraries**: code can be written in Keras/Tensorflow, or in PyTorch.

- **Plagiarism**: students may discuss the assignment together, but you may not look at another student or group's work or allow other students to view yours (other than within a group). You may use snippets of code (eg 1-2 lines) from the internet, **if you provide a citation with URL**. You may also use a longer snippet of code if it is a utility function, again only with citation. You may not use code from the internet to carry out the core of the assignment. You may not use a large language model to generate code.

- **Submission**: after completing your work in this Jupyter notebook, submit the notebook both in `.ipynb` and `.pdf` formats. The content should be identical.

- **Interviews**: a number of students may be selected for interview, post-submission. The selection will depend on submissions, and random chance may be used also. Interviews will be held in-person (CT5133) or online (CT5145). Interviews will last approximately 10 minutes. The purpose of interviews will be to assess students' understanding of their own submission.

Dataset Credits

The original csv file is from:

https://www.kaggle.com/datasets/neha1703/movie-genre-from-its-poster

I have added the *year* column for convenience.

I believe most of the information is originally from the famous MovieLens dataset:

- https://grouplens.org/datasets/movielens/
- https://movielens.org/

However, I'm not clear whether the poster download URLs (Amazon AWS URLs) which are in the csv obtained from the Kaggle URL above are from a MovieLens source, or elsewhere.

To create the dataset we are using, I have randomly sampled a small proportion of the URLs in the csv, and downloaded the images. I have removed those which fail to download. Code below also filters out those which are in black and white, ie 1 channel only.

Imports

You can add more imports if needed.

In [1]:
```
import numpy as np
import pandas as pd
import os
import math
from PIL import Image
import matplotlib.pyplot as plt
from scipy.spatial.distance import cdist, pdist, squareform # useful for distances
```

In [2]:
```
import tensorflow as tf
from tensorflow import keras
from keras import layers, models
from tensorflow.keras.callbacks import ModelCheckpoint
from sklearn.metrics import mean_absolute_error, mean_squared_error
from tensorflow.keras.applications.resnet50 import ResNet50, preprocess_input

np.random.seed(2023)
tf.random.set_seed(2023)
```

Utility functions

These functions are provided to save you time. You might not need to understand any of the details here.

In [3]:
```
# walk the directory containing posters and read them in. all are the same shape: (
# all have 3 channels, with a few exceptions (see below).
# each is named <imdbId>.jpg, which will later allow us to get the metadata from th
IDs = []
images = []
for dirname, _, filenames in os.walk('DL_Sample'):
    for filename in filenames:
        if filename.endswith(".jpg"):
            ID = int(filename[:-4])
            pathname = os.path.join(dirname, filename)
            im = Image.open(pathname)
            imnp = np.array(im, dtype=float)
            if len(imnp.shape) != 3: # we'll ignore a few black-and-white (1 channe
                print("This is 1 channel, so we omit it", imnp.shape, filename)
                continue # do not add to our list
            IDs.append(ID)
            images.append(imnp)
```

In [4]:
```
img_array = np.array(images)
```

In [5]:
```
img_array.shape
```

Out[5]:
```
(1238, 268, 182, 3)
```

In [14]:
```
# read the csv
df = pd.read_csv("Movie_Genre_Year_Poster.csv",
                 encoding="ISO-8859-1", index_col="Unnamed: 0")
df.head()
```

Out[14]:

	imdbId	Imdb Link	Title	IMDB Score	Genre	
0	114709	http://www.imdb.com/title/tt114709	Toy Story (1995)	8.3	Animation|Adventure|Comedy	https:/ amazor
1	113497	http://www.imdb.com/title/tt113497	Jumanji (1995)	6.9	Action|Adventure|Family	https:/ amazor

2	113228	http://www.imdb.com/title/tt113228	Grumpier Old Men (1995)	6.6	Comedy|Romance	https:/ amazor
3	114885	http://www.imdb.com/title/tt114885	Waiting to Exhale (1995)	5.7	Comedy|Drama|Romance	https:/ amazor
4	113041	http://www.imdb.com/title/tt113041	Father of the Bride Part II (1995)	5.9	Comedy|Family|Romance	https:/ amazor

In [7]:
```
df2 = df.drop_duplicates(subset=["imdbId"]) # some imdbId values are duplicates - j
```

In [8]:
```
df3 = df2.set_index("imdbId") # the imdbId is a more useful index, eg as in the nex
```

In [9]:
```
df4 = df3.loc[IDs] # ... we can now use .loc to take a subset
```

In [10]:
```
df4.shape # 1254 rows matches the image data shape above
```

Out[10]:
```
(1238, 6)
```

In [11]:
```
df4
```

Out[11]:

	Imdb Link	Title	IMDB Score	Genre	
imdbId					
1000771	http://www.imdb.com/title/tt1000771	Recount (2008)	7.5	Drama|History	https://im amazon.co
100148	http://www.imdb.com/title/tt100148	Midnight Ride (1990)	5.2	Action|Horror|Thriller	https://im amazon.co
1001540	http://www.imdb.com/title/tt1001540	Perestroika (2009)	4.5	Drama	https://im amazon.co
100200	http://www.imdb.com/title/tt100200	Mr. & Mrs. Bridge (1990)	6.7	Drama	https://im amazon.co
1002563	http://www.imdb.com/title/tt1002563	The Young Messiah (2016)	5.5	Drama	https://im amazon.co
...	...	...	...	...	
99611	http://www.imdb.com/title/tt99611	Frankenhooker (1990)	6.1	Comedy|Horror	https://im amazon.co
996967	http://www.imdb.com/title/tt996967	Otis (2008)	6.1	Comedy|Crime|Horror	https://im amazon.co

99726	http://www.imdb.com/title/tt99726	Hamlet (1990)	6.8	Drama	https://im amazon.co
99768	http://www.imdb.com/title/tt99768	Hidden Agenda (1990)	7.0	Drama|Thriller	https://im amazon.co
99836	http://www.imdb.com/title/tt99836	In nome del popolo sovrano (1990)	6.8	Drama|History	https://im amazon.co

1238 rows × 6 columns

In [12]:

```
years = df4["Year"].values
titles = df4["Title"].values

assert img_array.shape[0] == years.shape[0] == titles.shape[0]
```

In [13]:

```
def imread(filename):
    """Convenience function: we can supply an ID or a filename.
    We read and return the image in Image format.
    """

    if type(filename) == int:
        # assume its an ID, so create filename
        filename = f"DL_Sample/{filename}.jpg"

    # now we can assume it's a filename, so open and read
    im = Image.open(filename)

    return im

def imshow(im):
    plt.imshow(im)
    plt.axis('off')
```

Part 1. Create embedding [3 marks]

Use a pretrained model, eg as provided by Keras, to create a flat (ie 1D) embedding vector of some size `embedding_size` for each movie poster, and put all of these together into a single tensor of shape `(n_movies, embedding_size)`.

In [15]:

```
n_movies = img_array.shape[0]
embedding_size = 100352 # YOUR CODE HERE
X = tf.zeros((n_movies, embedding_size))

### YOUR CODE HERE
with tf.device("/GPU:0"):
    model = keras.models.Sequential([
        ResNet50(weights='imagenet',
                 include_top = False,
                 input_shape = (268, 182, 3)),
```

```
        layers.Flatten()
    ])
    x = preprocess_input(img_array)
    X = model.predict(x)


assert len(X.shape) == 2 # X should be (n_movies, embedding_size)
assert X.shape[0] == n_movies
```

```
39/39 [==============================] - 11s 79ms/step
```

In [16]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 resnet50 (Functional)       (None, 9, 6, 2048)        23587712  
                                                                 
 flatten (Flatten)           (None, 110592)            0         
                                                                 
=================================================================
Total params: 23,587,712
Trainable params: 23,534,592
Non-trainable params: 53,120
_________________________________________________________________
```

Part 2. Define a nearest-neighbour function [3 marks]

Write a function `def nearest(img, k)` which accepts an image `img`, and returns the `k` movies in the dataset whose posters are most similar to `img` (as measured in the embedding), ranked by similarity.

In [17]:
```
def k_nearest(img, k):
    ### YOUR CODE HERE
    if type(img) == int:
        image = f"DL_Sample/{img}.jpg"
        im = Image.open(image)
        img = np.array(im, dtype=float)
        img = tf.expand_dims(img, axis = 0)

    img = preprocess_input(img)
    pred = model.predict(img)

    distance = cdist(pred, X, metric = "hamming")
    img_index = np.argsort(distance[0])
    distance = np.sort(distance[0])

    neighbours = []
    for i in range(k+1):
        neighbours.append(IDs[img_index[i]])

    return np.array(distance[1:k+1]), np.array(neighbours[1:k+1])
```

In [17]:
```
k_nearest(90837, 2)
```

```
1/1 [==============================] - 1s 818ms/step
```

Out[17]:
```
(array([0.23546007, 0.23999928]), array([ 106215, 1977002]))
```

Part 3: Demonstrate your nearest-neighbour function [4 marks]

Choose any movie poster. Call this the query poster. Show it, and use your nearest-neighbour function to show the 3 nearest neighbours (excluding the query itself). This means **call** the function you defined above.

Write a comment: in what ways are they similar or dissimilar? Do you agree with the choice and the ranking? Why do you think they are close in the embedding? Do you notice, for example, that the nearest neighbours are from a similar era?

In [18]:
```python
### YOUR CODE HERE
'''
The images are showing as per below due to the lowest distance calculated
between them. The first image is the reference image and the other three
are the closest. There is a similarity of color in the 2nd image seen, while
there is not much similarity if we see the 3rd and the 4th image displayed,
but it feels the algorithm calculated the closest distance due to less colours
used in the posters. I don't agree to an extend with the ranking given by the
cdist algorithm as it's just calculating colour-based similarities and falls
over to the basic colour schemes when it doesn't find similar colours from the
given image. The embeddings may be close due to black and white being the most
in the pixels. I guess the era plays an important role in the poster designning
and so maybe the nearest neighbours are caught from the similar era as people
are trying to use the similar styles more often.
'''

fig = plt.figure(figsize = (15,4))


Q_idx = 90837 # YOUR VALUE HERE - DO NOT USE MY VALUE
distances, neighbours = k_nearest(90837, 3)




for i in range(0, 4):
    if i == 0:
        plt.subplot(1,4,i+1)
        plt.imshow(imread(Q_idx))
        plt.axis('off')
        plt.title('Original Image')
        continue

    plt.subplot(1,4,i+1)
    plt.imshow(imread(int(neighbours[i-1])))
    plt.axis('off')

plt.suptitle('Nearest Neighbours')
plt.show()
```

```
1/1 [==============================] - 1s 958ms/step
```

Part 4: Year regression [5 marks]

Let's investigate the last question ("similar era") above by running **regression** on the year, ie attempt to predict the year, given the poster. Use a train-test split. Build a suitable Keras neural network model for this, **as a regression head on top of the embedding from Part 1**. Include comments to explain the purpose of each part of the model. It should be possible to make a prediction, given a new poster (not part of the original dataset). Write a short comment on model performance: is it possible to predict the year? Based on this result, are there trends over time?

In [72]:

```
'''
We are making a new regression head with Sequential
model in which we keep 2 dense layers with `relu`
activation and the last dense layer acting as the
"linear" head
'''

regressor = keras.Sequential([
    layers.Dense(512, activation="relu"),
    layers.Dense(10, activation="relu"),
    layers.Dense(1, activation = "linear")
])
```

In [20]:

```
### YOUR CODE HERE
from sklearn.model_selection import train_test_split

y = df4.loc[IDs]
y = np.array(y["Year"])

assert X.shape[0] == y.shape[0]

# X is taken from Part 1 where the embeddings are created.
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, shuffle = True, random_state = 2023)
```

In [80]:

```
'''
Taking Adam Optimizer with learning rate 0.01,
and MeanSquaredError as Loss function as it is
a regresion task.
```

```
'''

regressor.compile(
    optimizer = tf.keras.optimizers.Adam(learning_rate = 0.01),
    loss = tf.keras.losses.MeanSquaredError()
)
print(regressor.optimizer)
print(regressor.loss)
```

```
<keras.optimizers.optimizer_v2.adam.Adam object at 0x000001CAD5DC80A0>
<keras.losses.MeanSquaredError object at 0x000001CAD5DC8100>
```

In [81]:

```
#Ref: https://www.tensorflow.org/api_docs/python/
#tf/keras/callbacks/ModelCheckpoint
'''
Model Checkpoint is the way to get the
best weights from the gradients as the
model's loss can vary, and finding the
lowest loss can mean a better regression
task. Here, validation loss is being used
to find the lowest loss and saving those
weights to use with the model.
'''

checkpoint_filepath = './tmp/checkpoint'
best_model = keras.callbacks.ModelCheckpoint(
    checkpoint_filepath, monitor='val_loss',
    save_best_only=True, save_weights_only=True, verbose = 1)
```

In [23]:

```
with tf.device("/GPU:0"):
    history = regressor.fit(X_train, y_train, epochs = 100,
                            batch_size = None, validation_split = 0.2,
                            callbacks = [best_model], shuffle = False)
```

```
Epoch 1/100
24/25 [===========================>..] - ETA: 0s - loss: 1502828.0000
Epoch 1: val_loss improved from inf to 164511.90625, saving model to ./tmp\checkpo
int
25/25 [==============================] - 3s 101ms/step - loss: 1465386.6250 - val_
loss: 164511.9062
Epoch 2/100
24/25 [===========================>..] - ETA: 0s - loss: 152599.6562
Epoch 2: val_loss improved from 164511.90625 to 56392.25781, saving model to ./tmp
\checkpoint
25/25 [==============================] - 3s 105ms/step - loss: 150925.8125 - val_l
oss: 56392.2578
Epoch 3/100
25/25 [==============================] - ETA: 0s - loss: 92736.5938
Epoch 3: val_loss did not improve from 56392.25781
25/25 [==============================] - 0s 20ms/step - loss: 92736.5938 - val_los
s: 65140.6914
Epoch 4/100
21/25 [=======================>.....] - ETA: 0s - loss: 30611.0742
Epoch 4: val_loss improved from 56392.25781 to 42580.81641, saving model to ./tmp
\checkpoint
25/25 [==============================] - 2s 68ms/step - loss: 28490.9121 - val_los
```

```
s: 42580.8164
Epoch 5/100
21/25 [========================>.....] - ETA: 0s - loss: 15930.2207
Epoch 5: val_loss improved from 42580.81641 to 37474.60938, saving model to ./tmp
\checkpoint
25/25 [==============================] - 2s 71ms/step - loss: 14895.6035 - val_los
s: 37474.6094
Epoch 6/100
22/25 [=========================>....] - ETA: 0s - loss: 8842.0312
Epoch 6: val_loss improved from 37474.60938 to 34844.38281, saving model to ./tmp
\checkpoint
25/25 [==============================] - 2s 83ms/step - loss: 8419.5547 - val_los
s: 34844.3828
Epoch 7/100
21/25 [========================>.....] - ETA: 0s - loss: 5178.4385
Epoch 7: val_loss improved from 34844.38281 to 34066.51172, saving model to ./tmp
\checkpoint
25/25 [==============================] - 2s 94ms/step - loss: 4923.6694 - val_los
s: 34066.5117
Epoch 8/100
24/25 [===========================>..] - ETA: 0s - loss: 3864.9250
Epoch 8: val_loss did not improve from 34066.51172
25/25 [==============================] - 0s 17ms/step - loss: 3819.6172 - val_los
s: 35224.4102
Epoch 9/100
21/25 [========================>.....] - ETA: 0s - loss: 4302.9424
Epoch 9: val_loss did not improve from 34066.51172
25/25 [==============================] - 0s 15ms/step - loss: 4732.1670 - val_los
s: 39472.2695
Epoch 10/100
22/25 [=========================>....] - ETA: 0s - loss: 7691.4409
Epoch 10: val_loss did not improve from 34066.51172
25/25 [==============================] - 0s 15ms/step - loss: 8538.3291 - val_los
s: 71326.8906
Epoch 11/100
23/25 [==========================>...] - ETA: 0s - loss: 18169.1758
Epoch 11: val_loss did not improve from 34066.51172
25/25 [==============================] - 0s 15ms/step - loss: 18771.5273 - val_los
s: 166969.4375
Epoch 12/100
21/25 [========================>.....] - ETA: 0s - loss: 53684.3086
Epoch 12: val_loss did not improve from 34066.51172
25/25 [==============================] - 0s 15ms/step - loss: 52720.3164 - val_los
s: 319127.5938
Epoch 13/100
25/25 [==============================] - ETA: 0s - loss: 125075.5156
Epoch 13: val_loss did not improve from 34066.51172
25/25 [==============================] - 0s 16ms/step - loss: 125075.5156 - val_lo
ss: 264667.6875
Epoch 14/100
21/25 [========================>.....] - ETA: 0s - loss: 506320.9375
Epoch 14: val_loss did not improve from 34066.51172
25/25 [==============================] - 0s 15ms/step - loss: 498225.1250 - val_lo
ss: 373416.5938
Epoch 15/100
21/25 [========================>.....] - ETA: 0s - loss: 410946.4375
```

```
Epoch 15: val_loss did not improve from 34066.51172
25/25 [==============================] - 0s 15ms/step - loss: 416432.2500 - val_lo
ss: 101915.3750
Epoch 16/100
21/25 [========================>.....] - ETA: 0s - loss: 73234.4141
Epoch 16: val_loss did not improve from 34066.51172
25/25 [==============================] - 0s 14ms/step - loss: 64752.6406 - val_los
s: 45913.0508
Epoch 17/100
24/25 [===========================>..] - ETA: 0s - loss: 29815.2363
Epoch 17: val_loss did not improve from 34066.51172
25/25 [==============================] - 0s 15ms/step - loss: 29669.8711 - val_los
s: 42051.4805
Epoch 18/100
22/25 [=========================>....] - ETA: 0s - loss: 16872.3086
Epoch 18: val_loss did not improve from 34066.51172
25/25 [==============================] - 0s 15ms/step - loss: 16492.3555 - val_los
s: 38505.0742
Epoch 19/100
24/25 [===========================>..] - ETA: 0s - loss: 11378.4912
Epoch 19: val_loss did not improve from 34066.51172
25/25 [==============================] - 0s 15ms/step - loss: 11271.8389 - val_los
s: 36936.2461
Epoch 20/100
21/25 [========================>.....] - ETA: 0s - loss: 7547.2231
Epoch 20: val_loss did not improve from 34066.51172
25/25 [==============================] - 0s 14ms/step - loss: 7254.2334 - val_los
s: 35606.8086
Epoch 21/100
21/25 [========================>.....] - ETA: 0s - loss: 5085.0967
Epoch 21: val_loss did not improve from 34066.51172
25/25 [==============================] - 0s 14ms/step - loss: 4831.7759 - val_los
s: 34693.7656
Epoch 22/100
21/25 [========================>.....] - ETA: 0s - loss: 3534.0559
Epoch 22: val_loss did not improve from 34066.51172
25/25 [==============================] - 0s 14ms/step - loss: 3309.6541 - val_los
s: 34085.8945
Epoch 23/100
25/25 [==============================] - ETA: 0s - loss: 2354.4038
Epoch 23: val_loss improved from 34066.51172 to 33622.48828, saving model to ./tmp
\checkpoint
25/25 [==============================] - 4s 181ms/step - loss: 2354.4038 - val_los
s: 33622.4883
Epoch 24/100
23/25 [==========================>...] - ETA: 0s - loss: 1842.4166
Epoch 24: val_loss improved from 33622.48828 to 33260.08984, saving model to ./tmp
\checkpoint
25/25 [==============================] - 2s 72ms/step - loss: 1763.7990 - val_los
s: 33260.0898
Epoch 25/100
25/25 [==============================] - ETA: 0s - loss: 1415.5349
Epoch 25: val_loss improved from 33260.08984 to 32947.67578, saving model to ./tmp
\checkpoint
25/25 [==============================] - 3s 130ms/step - loss: 1415.5349 - val_los
s: 32947.6758
```

```
Epoch 26/100
25/25 [==============================] - ETA: 0s - loss: 1211.9352
Epoch 26: val_loss improved from 32947.67578 to 32762.64453, saving model to ./tmp
\checkpoint
25/25 [==============================] - 5s 190ms/step - loss: 1211.9352 - val_los
s: 32762.6445
Epoch 27/100
24/25 [============================>..] - ETA: 0s - loss: 1133.1263
Epoch 27: val_loss improved from 32762.64453 to 32593.88867, saving model to ./tmp
\checkpoint
25/25 [==============================] - 6s 230ms/step - loss: 1106.4263 - val_los
s: 32593.8887
Epoch 28/100
24/25 [============================>..] - ETA: 0s - loss: 1076.5355
Epoch 28: val_loss improved from 32593.88867 to 32422.75977, saving model to ./tmp
\checkpoint
25/25 [==============================] - 4s 151ms/step - loss: 1049.6315 - val_los
s: 32422.7598
Epoch 29/100
23/25 [==========================>...] - ETA: 0s - loss: 1106.9125
Epoch 29: val_loss improved from 32422.75977 to 32277.82812, saving model to ./tmp
\checkpoint
25/25 [==============================] - 2s 103ms/step - loss: 1044.0121 - val_los
s: 32277.8281
Epoch 30/100
22/25 [=========================>....] - ETA: 0s - loss: 1131.7177
Epoch 30: val_loss improved from 32277.82812 to 32118.25000, saving model to ./tmp
\checkpoint
25/25 [==============================] - 3s 127ms/step - loss: 1057.0090 - val_los
s: 32118.2500
Epoch 31/100
21/25 [========================>.....] - ETA: 0s - loss: 1172.2039
Epoch 31: val_loss improved from 32118.25000 to 31882.93945, saving model to ./tmp
\checkpoint
25/25 [==============================] - 2s 74ms/step - loss: 1108.9742 - val_los
s: 31882.9395
Epoch 32/100
25/25 [==============================] - ETA: 0s - loss: 1184.7468
Epoch 32: val_loss improved from 31882.93945 to 31739.47461, saving model to ./tmp
\checkpoint
25/25 [==============================] - 1s 61ms/step - loss: 1184.7468 - val_los
s: 31739.4746
Epoch 33/100
24/25 [============================>..] - ETA: 0s - loss: 1319.0931
Epoch 33: val_loss improved from 31739.47461 to 31564.28320, saving model to ./tmp
\checkpoint
25/25 [==============================] - 2s 66ms/step - loss: 1287.1580 - val_los
s: 31564.2832
Epoch 34/100
21/25 [========================>.....] - ETA: 0s - loss: 1406.6533
Epoch 34: val_loss improved from 31564.28320 to 31477.81641, saving model to ./tmp
\checkpoint
25/25 [==============================] - 2s 64ms/step - loss: 1401.9070 - val_los
s: 31477.8164
Epoch 35/100
21/25 [========================>.....] - ETA: 0s - loss: 1522.1216
```

```
Epoch 35: val_loss improved from 31477.81641 to 31375.27344, saving model to ./tmp
\checkpoint
25/25 [==============================] - 2s 64ms/step - loss: 1549.6146 - val_los
s: 31375.2734
Epoch 36/100
21/25 [========================>.....] - ETA: 0s - loss: 1610.4963
Epoch 36: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 1693.1718 - val_los
s: 31505.1543
Epoch 37/100
21/25 [========================>.....] - ETA: 0s - loss: 1783.8988
Epoch 37: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 1890.2814 - val_los
s: 31777.4082
Epoch 38/100
21/25 [========================>.....] - ETA: 0s - loss: 1966.2169
Epoch 38: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 2106.8357 - val_los
s: 32504.0938
Epoch 39/100
21/25 [========================>.....] - ETA: 0s - loss: 2277.9536
Epoch 39: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 2402.2417 - val_los
s: 33513.6094
Epoch 40/100
21/25 [========================>.....] - ETA: 0s - loss: 2559.0496
Epoch 40: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 2740.3804 - val_los
s: 34732.8555
Epoch 41/100
21/25 [========================>.....] - ETA: 0s - loss: 2954.1670
Epoch 41: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 3224.8936 - val_los
s: 36410.2344
Epoch 42/100
21/25 [========================>.....] - ETA: 0s - loss: 3059.1499
Epoch 42: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 3609.4011 - val_los
s: 39467.8477
Epoch 43/100
21/25 [========================>.....] - ETA: 0s - loss: 3142.1619
Epoch 43: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 4174.2222 - val_los
s: 47288.0547
Epoch 44/100
21/25 [========================>.....] - ETA: 0s - loss: 3168.2017
Epoch 44: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 4599.1382 - val_los
s: 65774.1797
Epoch 45/100
21/25 [========================>.....] - ETA: 0s - loss: 4370.9380
Epoch 45: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 5682.2783 - val_los
s: 106485.6641
Epoch 46/100
21/25 [========================>.....] - ETA: 0s - loss: 7218.2510
```

```
Epoch 46: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 6934.4771 - val_los
s: 169308.8438
Epoch 47/100
21/25 [========================>.....] - ETA: 0s - loss: 17817.2969
Epoch 47: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 15471.5596 - val_los
s: 197121.7344
Epoch 48/100
21/25 [========================>.....] - ETA: 0s - loss: 40812.0312
Epoch 48: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 37547.5547 - val_los
s: 112913.3516
Epoch 49/100
25/25 [==============================] - ETA: 0s - loss: 84009.8125
Epoch 49: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 15ms/step - loss: 84009.8125 - val_los
s: 49584.4531
Epoch 50/100
21/25 [========================>.....] - ETA: 0s - loss: 210373.8750
Epoch 50: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 243728.1406 - val_lo
ss: 73965.4531
Epoch 51/100
21/25 [========================>.....] - ETA: 0s - loss: 216360.5000
Epoch 51: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 186887.6094 - val_lo
ss: 42169.1250
Epoch 52/100
21/25 [========================>.....] - ETA: 0s - loss: 80918.8750
Epoch 52: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 128104.6094 - val_lo
ss: 383173.7500
Epoch 53/100
21/25 [========================>.....] - ETA: 0s - loss: 155318.9375
Epoch 53: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 143070.9219 - val_lo
ss: 670593.5625
Epoch 54/100
21/25 [========================>.....] - ETA: 0s - loss: 131050.0000
Epoch 54: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 115267.0234 - val_lo
ss: 385770.3125
Epoch 55/100
21/25 [========================>.....] - ETA: 0s - loss: 42929.7070
Epoch 55: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 42656.9258 - val_los
s: 668777.3125
Epoch 56/100
21/25 [========================>.....] - ETA: 0s - loss: 70576.2812
Epoch 56: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 66213.0625 - val_los
s: 770957.5000
Epoch 57/100
21/25 [========================>.....] - ETA: 0s - loss: 78415.6406
Epoch 57: val_loss did not improve from 31375.27344
```

```
25/25 [==============================] - 0s 14ms/step - loss: 73715.0234 - val_los
s: 1054919.3750
Epoch 58/100
21/25 [========================>.....] - ETA: 0s - loss: 105719.2266
Epoch 58: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 96012.2812 - val_los
s: 1391709.8750
Epoch 59/100
21/25 [========================>.....] - ETA: 0s - loss: 145541.7031
Epoch 59: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 130431.7344 - val_lo
ss: 1772227.3750
Epoch 60/100
21/25 [========================>.....] - ETA: 0s - loss: 189464.3125
Epoch 60: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 165299.7812 - val_lo
ss: 2086479.0000
Epoch 61/100
21/25 [========================>.....] - ETA: 0s - loss: 266649.8438
Epoch 61: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 233789.4375 - val_lo
ss: 2436626.0000
Epoch 62/100
21/25 [========================>.....] - ETA: 0s - loss: 253784.1406
Epoch 62: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 223435.7500 - val_lo
ss: 2300924.5000
Epoch 63/100
21/25 [========================>.....] - ETA: 0s - loss: 379358.9688
Epoch 63: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 335256.1250 - val_lo
ss: 1718038.6250
Epoch 64/100
21/25 [========================>.....] - ETA: 0s - loss: 309557.5312
Epoch 64: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 383998.3750 - val_lo
ss: 332124.4375
Epoch 65/100
21/25 [========================>.....] - ETA: 0s - loss: 956708.6875
Epoch 65: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 1050667.5000 - val_l
oss: 361360.5312
Epoch 66/100
21/25 [========================>.....] - ETA: 0s - loss: 1040741.1250
Epoch 66: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 1169680.7500 - val_l
oss: 1193676.3750
Epoch 67/100
21/25 [========================>.....] - ETA: 0s - loss: 717005.0625
Epoch 67: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 744665.0000 - val_lo
ss: 534100.1875
Epoch 68/100
21/25 [========================>.....] - ETA: 0s - loss: 430351.6562
Epoch 68: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 469662.4688 - val_lo
```

```
ss: 456036.7812
Epoch 69/100
21/25 [=======================>.....] - ETA: 0s - loss: 319943.1562
Epoch 69: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 341175.9062 - val_lo
ss: 339976.3125
Epoch 70/100
21/25 [=======================>.....] - ETA: 0s - loss: 224875.6719
Epoch 70: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 238796.1875 - val_lo
ss: 274990.7188
Epoch 71/100
21/25 [=======================>.....] - ETA: 0s - loss: 157095.5938
Epoch 71: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 165016.4688 - val_lo
ss: 230485.7812
Epoch 72/100
24/25 [===========================>..] - ETA: 0s - loss: 114339.9609
Epoch 72: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 114869.3047 - val_lo
ss: 208880.6094
Epoch 73/100
21/25 [=======================>.....] - ETA: 0s - loss: 81045.1875
Epoch 73: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 80273.9375 - val_los
s: 185197.6562
Epoch 74/100
21/25 [=======================>.....] - ETA: 0s - loss: 64419.2695
Epoch 74: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 62622.4180 - val_los
s: 180950.7656
Epoch 75/100
21/25 [=======================>.....] - ETA: 0s - loss: 73477.9453
Epoch 75: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 68461.8750 - val_los
s: 265095.9688
Epoch 76/100
21/25 [=======================>.....] - ETA: 0s - loss: 51504.7148
Epoch 76: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 46929.8086 - val_los
s: 81836.2266
Epoch 77/100
21/25 [=======================>.....] - ETA: 0s - loss: 22671.8086
Epoch 77: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 23456.9277 - val_los
s: 102311.5859
Epoch 78/100
21/25 [=======================>.....] - ETA: 0s - loss: 90958.8984
Epoch 78: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 80702.0938 - val_los
s: 299278.4688
Epoch 79/100
21/25 [=======================>.....] - ETA: 0s - loss: 129266.7656
Epoch 79: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 113372.4922 - val_lo
ss: 302208.8438
```

```
Epoch 80/100
21/25 [========================>.....] - ETA: 0s - loss: 130332.2891
Epoch 80: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 112688.9375 - val_lo
ss: 237447.3594
Epoch 81/100
24/25 [===========================>..] - ETA: 0s - loss: 120581.4297
Epoch 81: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 15ms/step - loss: 117243.4375 - val_lo
ss: 288891.9375
Epoch 82/100
24/25 [===========================>..] - ETA: 0s - loss: 103357.6875
Epoch 82: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 15ms/step - loss: 100448.1797 - val_lo
ss: 40029.0820
Epoch 83/100
21/25 [========================>.....] - ETA: 0s - loss: 34481.3516
Epoch 83: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 36108.6914 - val_los
s: 43352.9258
Epoch 84/100
21/25 [========================>.....] - ETA: 0s - loss: 72916.4141
Epoch 84: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 80313.6094 - val_los
s: 41729.9102
Epoch 85/100
21/25 [========================>.....] - ETA: 0s - loss: 49406.5312
Epoch 85: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 49420.3242 - val_los
s: 41530.7539
Epoch 86/100
21/25 [========================>.....] - ETA: 0s - loss: 20957.6523
Epoch 86: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 23200.7656 - val_los
s: 51660.7812
Epoch 87/100
21/25 [========================>.....] - ETA: 0s - loss: 10693.7422
Epoch 87: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 10868.9570 - val_los
s: 48435.3320
Epoch 88/100
21/25 [========================>.....] - ETA: 0s - loss: 9336.3916
Epoch 88: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 9105.0029 - val_los
s: 41756.6211
Epoch 89/100
25/25 [==============================] - ETA: 0s - loss: 14597.3457
Epoch 89: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 14597.3457 - val_los
s: 39404.0312
Epoch 90/100
21/25 [========================>.....] - ETA: 0s - loss: 19172.2461
Epoch 90: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 20528.5098 - val_los
s: 40159.0664
Epoch 91/100
```

```
21/25 [========================>.....] - ETA: 0s - loss: 19872.0977
Epoch 91: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 20890.8652 - val_los
s: 40678.5273
Epoch 92/100
21/25 [========================>.....] - ETA: 0s - loss: 16383.7705
Epoch 92: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 16778.2070 - val_los
s: 39987.7969
Epoch 93/100
21/25 [========================>.....] - ETA: 0s - loss: 11863.0293
Epoch 93: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 11707.1514 - val_los
s: 38826.6953
Epoch 94/100
21/25 [========================>.....] - ETA: 0s - loss: 7926.8867
Epoch 94: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 7441.4614 - val_los
s: 37557.3086
Epoch 95/100
21/25 [========================>.....] - ETA: 0s - loss: 5727.5234
Epoch 95: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 5436.2153 - val_los
s: 36351.0078
Epoch 96/100
21/25 [========================>.....] - ETA: 0s - loss: 5952.1060
Epoch 96: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 6131.3203 - val_los
s: 35893.7383
Epoch 97/100
21/25 [========================>.....] - ETA: 0s - loss: 7416.6592
Epoch 97: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 7778.2539 - val_los
s: 35905.4023
Epoch 98/100
21/25 [========================>.....] - ETA: 0s - loss: 8882.5020
Epoch 98: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 9687.7070 - val_los
s: 36758.5664
Epoch 99/100
21/25 [========================>.....] - ETA: 0s - loss: 9582.3477
Epoch 99: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 14ms/step - loss: 10638.1592 - val_los
s: 38262.7109
Epoch 100/100
21/25 [========================>.....] - ETA: 0s - loss: 8628.2402
Epoch 100: val_loss did not improve from 31375.27344
25/25 [==============================] - 0s 13ms/step - loss: 9929.9707 - val_los
s: 42266.3438
```

In [24]:

```python
regressor.summary()

'''
Load weight takes the file we created using
Model Checkpoint and loads the weight it
found while the validation loss was lowest.
```

```
'''
regressor.load_weights(checkpoint_filepath)
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 512)               56623616

 dense_1 (Dense)             (None, 10)                5130

 dense_2 (Dense)             (None, 1)                 11

=================================================================
Total params: 56,628,757
Trainable params: 56,628,757
Non-trainable params: 0
_________________________________________________________________
```

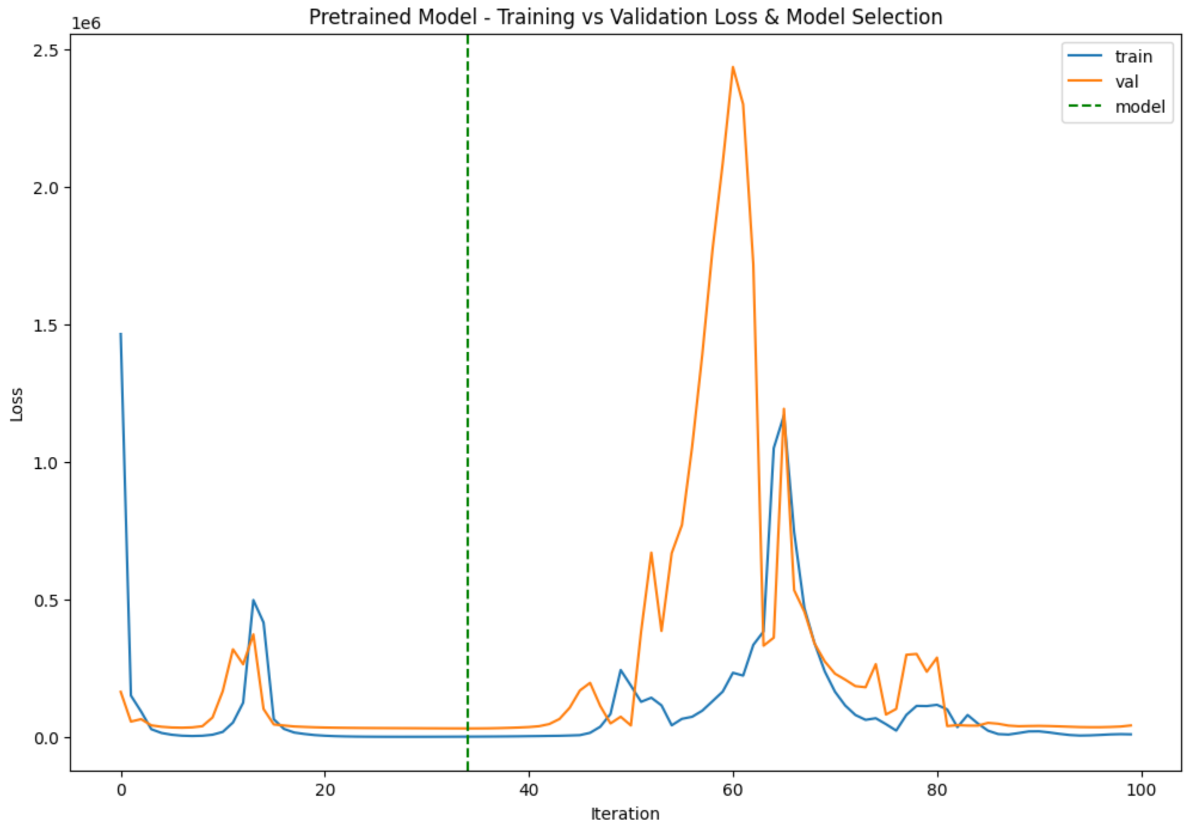
Out[24]:

```
<tensorflow.python.checkpoint.checkpoint.CheckpointLoadStatus at 0x1caa76cffa0>
```

In [56]:

```
lowest = np.amin(history.history['val_loss'])
time = np.where(history.history['val_loss'] == lowest)


fig = plt.figure(figsize = (12,8))
fig = plt.plot(history.history['loss'])
fig = plt.plot(history.history['val_loss'])
fig = plt.axvline(x = time, color = 'green', linestyle = '--')
plt.title(
    'Pretrained Model - Training vs Validation Loss & Model Selection')
plt.ylabel('Loss')
plt.xlabel('Iteration')
plt.legend(['train', 'val', 'model'], loc= 'upper right')
plt.show()
```



In [25]:
```
preds = regressor.predict(X_test)
preds = preds.flatten()
preds_round = [round(val, 0) for val in preds]
```

```
8/8 [==============================] - 0s 3ms/step
```

In [26]:
```
print(f"Mean Absolute Error:
      {mean_absolute_error(y_test, preds_round)}\n")
print(f"Mean Squared Error:
      {mean_squared_error(y_test, preds_round)}\n")
print(f"Root Mean Squared Error:
      {math.sqrt(mean_squared_error(y_test, preds_round))}\n")
```

```
Mean Absolute Error: 152.7701612903226

Mean Squared Error: 37984.13306451613

Root Mean Squared Error: 194.89518481613683
```

In [27]:
```
out_df = pd.DataFrame({"Y_test" : y_test,
                       "Predictions" : preds_round,
                       "Difference" : y_test - preds_round},
                      columns =
                      ["Y_test", "Predictions", "Difference"])
out_df
```

Out[27]:

	Y_test	Predictions	Difference
0	1991.0	1818.0	173.0
1	1992.0	1837.0	155.0

2	2011.0	2331.0	-320.0
3	1967.0	1823.0	144.0
4	1992.0	1733.0	259.0
...	...	...	...
243	2007.0	2189.0	-182.0
244	1998.0	1899.0	99.0
245	1952.0	1798.0	154.0
246	1980.0	2089.0	-109.0
247	2014.0	2040.0	-26.0

248 rows × 3 columns

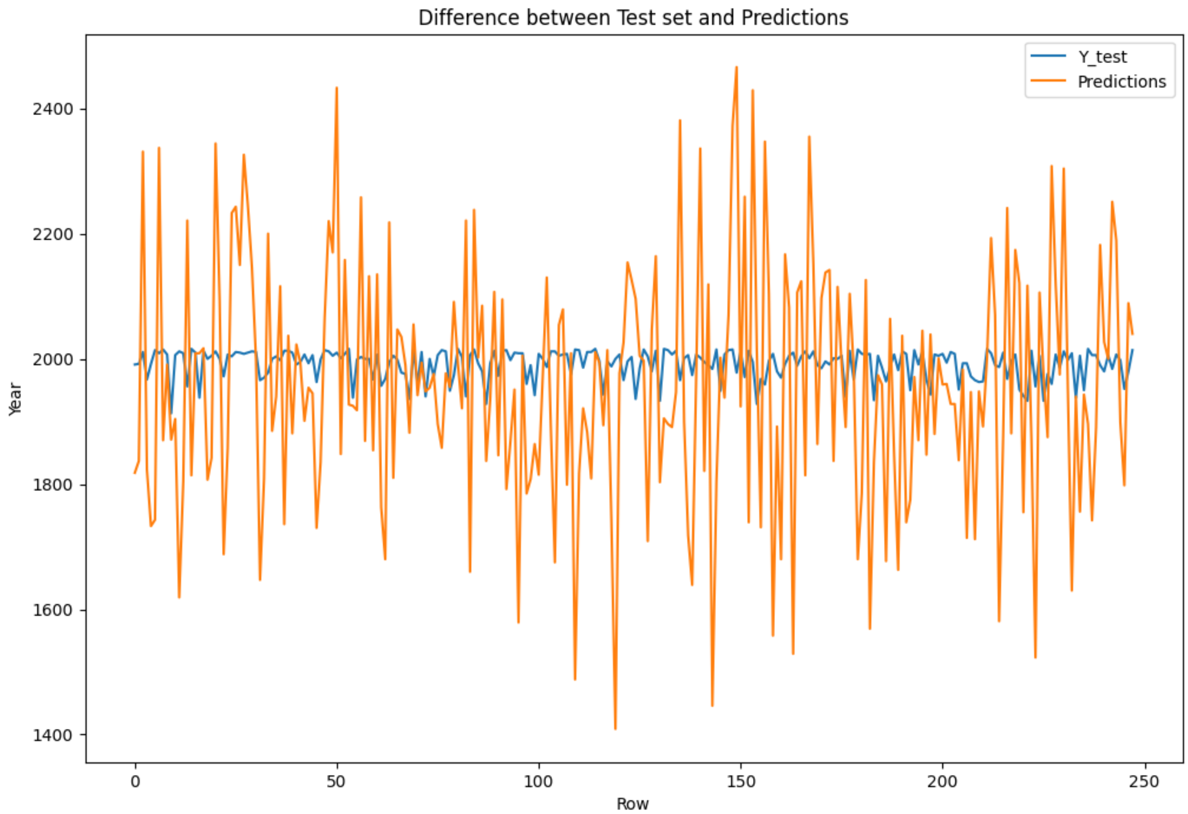
In [70]:

```
fig = plt.figure(figsize = (12,8))
plt.plot(out_df[['Y_test','Predictions']])
plt.legend(['Y_test', 'Predictions'], loc= 'upper right')
plt.xlabel('Row')
plt.ylabel('Year')
plt.title("Difference between Test set and Predictions")
plt.suptitle('Pretrained Model')

plt.plot()
```

Out[70]: []

Pretrained Model



Part 5: Improvements [5 marks]

Propose a possible improvement. Some ideas are suggested below. The chosen improvement must be notified to the lecturer at least 1 week before submission and **must be approved by the lecturer to avoid duplication with other students**. Compare the performance between your original and your new model (the proposed improvement might not actually improve on model performance -- that is ok). Some marks will be awarded for more interesting / challenging / novel improvements.

Ideas:

- Try a different pretrained model for creating the embedding
- Alternative ways of reducing the pretrained model's output to a flat vector for the embedding
- Gather more data (see the csv file for URLs)
- Add different architectural details to the regression head
- Fine-tuning
- Training an end-to-end convnet of your own design (no pretraining)
- Improve the embedding by training a multi-headed model, eg predicting both genre and year
- Create a good visualisation of the embedding.

5.1 Model Implementation

Creating a model with 4 Conv2D layers and 4 Average Pooling layers, Flatten & Dropout layers, and 4 Dense layers with final dense layer as linear activation for regression task.

In [28]:
```
'''
Training an end-to-end convnet of your
own design (no pretraining)
'''

new_model = keras.Sequential([
    layers.Conv2D(32, (3, 3), padding = "same",
                  activation = 'relu',
                  input_shape = (268, 182, 3),
                  name = 'Conv2D_32'),
    layers.AveragePooling2D(pool_size = (2, 2),
                            strides = None,
                            padding = "valid",
                            name = 'Avg_Pool1'),
    layers.Conv2D(64, (3, 3), padding = "valid",
                  activation = 'relu',
                  name = 'Conv2D_64'),
    layers.AveragePooling2D(pool_size = (2, 2),
                            strides = None,
                            padding = "valid",
                            name = 'Avg_Pool2'),
    layers.Conv2D(128, (5, 5),
                  padding = "same",
                  activation = 'relu',
                  name = 'Conv2D_128'),
    layers.AveragePooling2D(pool_size = (2, 2),
                            strides = None,
                            padding = "valid",
                            name = 'Avg_Pool3'),
    layers.Conv2D(256, (7, 7), padding = "same",
                  activation = 'relu',
                  name = 'Conv2D_64_2'),
    layers.AveragePooling2D(pool_size = (2, 2),
                            strides = None,
                            padding = "valid",
                            name = 'Avg_Pool4'),
    layers.Flatten(),
    layers.Dropout(0.5),
    layers.Dense(128, activation="relu",
                 name = 'dense_1'),
    layers.Dense(64, activation="relu",
                 name = 'dense_2'),
    layers.Dense(10, activation="relu",
                 name = 'dense_3'),
    layers.Dense(1, activation = "linear",
                 name = 'linear_layer')
])
```

5.2 Model Compilation

Choosing Adadelta with learning rate 0.01 Calculating loss with Mean Squared Error for Regression Tasks

In [29]:
```
new_model.compile(
    optimizer = tf.keras.optimizers.Adadelta(learning_rate = 0.01),
    loss = tf.keras.losses.MeanSquaredError()
)
```

In [30]:
```
new_model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 Conv2D_32 (Conv2D)          (None, 268, 182, 32)      896       
                                                                 
 Avg_Pool1 (AveragePooling2D  (None, 134, 91, 32)      0         
 )                                                               
                                                                 
 Conv2D_64 (Conv2D)          (None, 132, 89, 64)       18496     
                                                                 
 Avg_Pool2 (AveragePooling2D  (None, 66, 44, 64)       0         
 )                                                               
                                                                 
 Conv2D_128 (Conv2D)         (None, 66, 44, 128)       204928    
                                                                 
 Avg_Pool3 (AveragePooling2D  (None, 33, 22, 128)      0         
 )                                                               
                                                                 
 Conv2D_64_2 (Conv2D)        (None, 33, 22, 256)       1605888   
                                                                 
 Avg_Pool4 (AveragePooling2D  (None, 16, 11, 256)      0         
 )                                                               
                                                                 
 flatten_1 (Flatten)         (None, 45056)             0         
                                                                 
 dropout (Dropout)           (None, 45056)             0         
                                                                 
 dense_1 (Dense)             (None, 128)               5767296   
                                                                 
 dense_2 (Dense)             (None, 64)                8256      
                                                                 
 dense_3 (Dense)             (None, 10)                650       
                                                                 
 linear_layer (Dense)        (None, 1)                 11        
                                                                 
=================================================================
Total params: 7,606,421
Trainable params: 7,606,421
Non-trainable params: 0
_________________________________________________________________
```

5.3 Model Checkpoint

Saving the weights of the lowest value in validation loss found during training phase.

In [31]:
```
new_checkpoint_filepath = './tmp/checkpoint1'
new_best = keras.callbacks.ModelCheckpoint(
    new_checkpoint_filepath, monitor='val_loss',
    save_best_only=True, save_weights_only=True,
    verbose = 1)
```

In [32]:
```
X_train1, X_test1, y_train1, y_test1 =
train_test_split(img_array/255, y, test_size=0.2,
                 shuffle = True, random_state = 2023)
```

In [33]:
```
#Ref: https://www.linkedin.com/pulse/
#solving-out-memory-oom-errors-keras-tensorflow-running-wayne-cheng/
'''
This code is taken from the above reference, and is
used to dynamically allocate memory to GPU making it
available for each tensorflow operation. This code is
making sure the below code runs without issues.
'''

from tensorflow.compat.v1 import ConfigProto
from tensorflow.compat.v1 import InteractiveSession
config = ConfigProto()
config.gpu_options.allow_growth = True
session = InteractiveSession(config=config)
```

5.4 Model Training

Training the model with 100 epochs, batch size 8 and 20% data split for validation purposes.

In [34]:
```
with tf.device("/GPU:0"):
    new_history = new_model.fit(X_train1, y_train1,
                                epochs = 100, batch_size = 8,
                                validation_split = 0.2,
                                callbacks = [new_best],
                                shuffle = False)
```

```
Epoch 1/100
97/99 [============================>.] - ETA: 0s - loss: 1113675.8750
Epoch 1: val_loss improved from inf to 199493.71875, saving model to ./tmp\checkpo
int1
99/99 [==============================] - 6s 31ms/step - loss: 1097709.2500 - val_l
oss: 199493.7188
Epoch 2/100
97/99 [============================>.] - ETA: 0s - loss: 115009.9141
Epoch 2: val_loss improved from 199493.71875 to 49021.37500, saving model to ./tmp
\checkpoint1
99/99 [==============================] - 3s 26ms/step - loss: 113602.0469 - val_lo
ss: 49021.3750
Epoch 3/100
97/99 [============================>.] - ETA: 0s - loss: 52829.0938
Epoch 3: val_loss improved from 49021.37500 to 27767.32031, saving model to ./tmp
\checkpoint1
99/99 [==============================] - 3s 26ms/step - loss: 52240.7227 - val_los
s: 27767.3203
```

```
Epoch 4/100
97/99 [============================>.] - ETA: 0s - loss: 31089.6992
Epoch 4: val_loss improved from 27767.32031 to 17154.51953, saving model to ./tmp
\checkpoint1
99/99 [==============================] - 3s 26ms/step - loss: 30807.8262 - val_los
s: 17154.5195
Epoch 5/100
97/99 [============================>.] - ETA: 0s - loss: 17615.9395
Epoch 5: val_loss improved from 17154.51953 to 9473.31543, saving model to ./tmp\c
heckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 17469.0137 - val_los
s: 9473.3154
Epoch 6/100
97/99 [============================>.] - ETA: 0s - loss: 9523.3838
Epoch 6: val_loss improved from 9473.31543 to 5432.30127, saving model to ./tmp\ch
eckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 9425.7119 - val_los
s: 5432.3013
Epoch 7/100
97/99 [============================>.] - ETA: 0s - loss: 5875.0405
Epoch 7: val_loss improved from 5432.30127 to 4634.50391, saving model to ./tmp\ch
eckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 5881.0977 - val_los
s: 4634.5039
Epoch 8/100
97/99 [============================>.] - ETA: 0s - loss: 4534.8784
Epoch 8: val_loss improved from 4634.50391 to 2756.56763, saving model to ./tmp\ch
eckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 4484.3726 - val_los
s: 2756.5676
Epoch 9/100
97/99 [============================>.] - ETA: 0s - loss: 3720.7610
Epoch 9: val_loss did not improve from 2756.56763
99/99 [==============================] - 2s 23ms/step - loss: 3672.4504 - val_los
s: 3195.5369
Epoch 10/100
97/99 [============================>.] - ETA: 0s - loss: 3257.9434
Epoch 10: val_loss improved from 2756.56763 to 2052.87085, saving model to ./tmp\c
heckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 3233.5334 - val_los
s: 2052.8708
Epoch 11/100
97/99 [============================>.] - ETA: 0s - loss: 2925.9756
Epoch 11: val_loss did not improve from 2052.87085
99/99 [==============================] - 2s 23ms/step - loss: 2893.0789 - val_los
s: 2324.3059
Epoch 12/100
97/99 [============================>.] - ETA: 0s - loss: 2711.5857
Epoch 12: val_loss improved from 2052.87085 to 1768.86731, saving model to ./tmp\c
heckpoint1
99/99 [==============================] - 3s 25ms/step - loss: 2688.6140 - val_los
s: 1768.8673
Epoch 13/100
97/99 [============================>.] - ETA: 0s - loss: 2313.4993
Epoch 13: val_loss improved from 1768.86731 to 1597.73645, saving model to ./tmp\c
heckpoint1
```

```
99/99 [==============================] - 3s 26ms/step - loss: 2296.9609 - val_los
s: 1597.7365
Epoch 14/100
97/99 [============================>.] - ETA: 0s - loss: 2215.9700
Epoch 14: val_loss did not improve from 1597.73645
99/99 [==============================] - 2s 23ms/step - loss: 2193.0525 - val_los
s: 1608.4677
Epoch 15/100
97/99 [============================>.] - ETA: 0s - loss: 2041.8558
Epoch 15: val_loss improved from 1597.73645 to 1495.98511, saving model to ./tmp\c
heckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 2020.8800 - val_los
s: 1495.9851
Epoch 16/100
97/99 [============================>.] - ETA: 0s - loss: 1912.2028
Epoch 16: val_loss did not improve from 1495.98511
99/99 [==============================] - 2s 23ms/step - loss: 1887.9541 - val_los
s: 1568.5176
Epoch 17/100
97/99 [============================>.] - ETA: 0s - loss: 1968.9691
Epoch 17: val_loss improved from 1495.98511 to 1385.56665, saving model to ./tmp\c
heckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 1954.1174 - val_los
s: 1385.5667
Epoch 18/100
97/99 [============================>.] - ETA: 0s - loss: 1801.2687
Epoch 18: val_loss improved from 1385.56665 to 1360.62781, saving model to ./tmp\c
heckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 1785.8141 - val_los
s: 1360.6278
Epoch 19/100
97/99 [============================>.] - ETA: 0s - loss: 1878.4467
Epoch 19: val_loss did not improve from 1360.62781
99/99 [==============================] - 2s 23ms/step - loss: 1864.9908 - val_los
s: 1411.4452
Epoch 20/100
97/99 [============================>.] - ETA: 0s - loss: 1708.6461
Epoch 20: val_loss did not improve from 1360.62781
99/99 [==============================] - 2s 23ms/step - loss: 1684.6356 - val_los
s: 1363.9288
Epoch 21/100
97/99 [============================>.] - ETA: 0s - loss: 1686.0084
Epoch 21: val_loss improved from 1360.62781 to 1352.38257, saving model to ./tmp\c
heckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 1670.6486 - val_los
s: 1352.3826
Epoch 22/100
97/99 [============================>.] - ETA: 0s - loss: 1706.5745
Epoch 22: val_loss improved from 1352.38257 to 1276.40649, saving model to ./tmp\c
heckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 1691.6255 - val_los
s: 1276.4065
Epoch 23/100
97/99 [============================>.] - ETA: 0s - loss: 1665.7821
Epoch 23: val_loss improved from 1276.40649 to 1231.35181, saving model to ./tmp\c
heckpoint1
```

```
99/99 [==============================] - 3s 26ms/step - loss: 1650.0054 - val_los
s: 1231.3518
Epoch 24/100
97/99 [============================>.] - ETA: 0s - loss: 1638.4753
Epoch 24: val_loss did not improve from 1231.35181
99/99 [==============================] - 2s 24ms/step - loss: 1622.0646 - val_los
s: 1233.0322
Epoch 25/100
97/99 [============================>.] - ETA: 0s - loss: 1560.0590
Epoch 25: val_loss did not improve from 1231.35181
99/99 [==============================] - 2s 23ms/step - loss: 1556.0712 - val_los
s: 1268.5778
Epoch 26/100
97/99 [============================>.] - ETA: 0s - loss: 1530.2804
Epoch 26: val_loss improved from 1231.35181 to 1206.68848, saving model to ./tmp\c
heckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 1523.9667 - val_los
s: 1206.6885
Epoch 27/100
97/99 [============================>.] - ETA: 0s - loss: 1608.7812
Epoch 27: val_loss did not improve from 1206.68848
99/99 [==============================] - 2s 23ms/step - loss: 1600.3706 - val_los
s: 1245.7645
Epoch 28/100
97/99 [============================>.] - ETA: 0s - loss: 1523.9133
Epoch 28: val_loss improved from 1206.68848 to 1162.13074, saving model to ./tmp\c
heckpoint1
99/99 [==============================] - 3s 25ms/step - loss: 1511.5260 - val_los
s: 1162.1307
Epoch 29/100
97/99 [============================>.] - ETA: 0s - loss: 1428.4041
Epoch 29: val_loss did not improve from 1162.13074
99/99 [==============================] - 2s 23ms/step - loss: 1410.4139 - val_los
s: 1195.3500
Epoch 30/100
97/99 [============================>.] - ETA: 0s - loss: 1557.0293
Epoch 30: val_loss did not improve from 1162.13074
99/99 [==============================] - 2s 23ms/step - loss: 1552.2477 - val_los
s: 1165.4738
Epoch 31/100
97/99 [============================>.] - ETA: 0s - loss: 1422.7166
Epoch 31: val_loss did not improve from 1162.13074
99/99 [==============================] - 2s 23ms/step - loss: 1411.0532 - val_los
s: 1162.4868
Epoch 32/100
97/99 [============================>.] - ETA: 0s - loss: 1466.3696
Epoch 32: val_loss did not improve from 1162.13074
99/99 [==============================] - 2s 23ms/step - loss: 1450.3861 - val_los
s: 1343.4525
Epoch 33/100
97/99 [============================>.] - ETA: 0s - loss: 1513.7653
Epoch 33: val_loss improved from 1162.13074 to 1128.46033, saving model to ./tmp\c
heckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 1495.3000 - val_los
s: 1128.4603
Epoch 34/100
```

```
97/99 [============================>.] - ETA: 0s - loss: 1442.6057
Epoch 34: val_loss did not improve from 1128.46033
99/99 [==============================] - 2s 23ms/step - loss: 1428.1914 - val_los
s: 1390.7494
Epoch 35/100
97/99 [============================>.] - ETA: 0s - loss: 1490.0479
Epoch 35: val_loss did not improve from 1128.46033
99/99 [==============================] - 2s 23ms/step - loss: 1482.1483 - val_los
s: 1154.0919
Epoch 36/100
97/99 [============================>.] - ETA: 0s - loss: 1438.4510
Epoch 36: val_loss improved from 1128.46033 to 1094.68567, saving model to ./tmp\c
heckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 1423.6289 - val_los
s: 1094.6857
Epoch 37/100
97/99 [============================>.] - ETA: 0s - loss: 1407.3519
Epoch 37: val_loss did not improve from 1094.68567
99/99 [==============================] - 2s 24ms/step - loss: 1406.6230 - val_los
s: 1207.4171
Epoch 38/100
97/99 [============================>.] - ETA: 0s - loss: 1529.8103
Epoch 38: val_loss improved from 1094.68567 to 1076.73438, saving model to ./tmp\c
heckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 1514.6276 - val_los
s: 1076.7344
Epoch 39/100
97/99 [============================>.] - ETA: 0s - loss: 1417.6285
Epoch 39: val_loss did not improve from 1076.73438
99/99 [==============================] - 2s 24ms/step - loss: 1414.5140 - val_los
s: 1113.1228
Epoch 40/100
97/99 [============================>.] - ETA: 0s - loss: 1421.3180
Epoch 40: val_loss did not improve from 1076.73438
99/99 [==============================] - 2s 23ms/step - loss: 1436.8005 - val_los
s: 1465.8625
Epoch 41/100
97/99 [============================>.] - ETA: 0s - loss: 1465.8661
Epoch 41: val_loss improved from 1076.73438 to 1076.26660, saving model to ./tmp\c
heckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 1459.4215 - val_los
s: 1076.2666
Epoch 42/100
97/99 [============================>.] - ETA: 0s - loss: 1366.7908
Epoch 42: val_loss improved from 1076.26660 to 1056.73804, saving model to ./tmp\c
heckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 1360.2592 - val_los
s: 1056.7380
Epoch 43/100
97/99 [============================>.] - ETA: 0s - loss: 1367.2927
Epoch 43: val_loss improved from 1056.73804 to 1042.40808, saving model to ./tmp\c
heckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 1356.0323 - val_los
s: 1042.4081
Epoch 44/100
97/99 [============================>.] - ETA: 0s - loss: 1359.8033
```

```
Epoch 44: val_loss improved from 1042.40808 to 1038.53003, saving model to ./tmp\c
heckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 1351.3352 - val_los
s: 1038.5300
Epoch 45/100
97/99 [============================>.] - ETA: 0s - loss: 1340.7799
Epoch 45: val_loss did not improve from 1038.53003
99/99 [==============================] - 2s 23ms/step - loss: 1326.6633 - val_los
s: 1094.0311
Epoch 46/100
97/99 [============================>.] - ETA: 0s - loss: 1381.8496
Epoch 46: val_loss improved from 1038.53003 to 1028.87610, saving model to ./tmp\c
heckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 1372.3013 - val_los
s: 1028.8761
Epoch 47/100
97/99 [============================>.] - ETA: 0s - loss: 1279.8744
Epoch 47: val_loss did not improve from 1028.87610
99/99 [==============================] - 2s 24ms/step - loss: 1274.8235 - val_los
s: 1270.1997
Epoch 48/100
97/99 [============================>.] - ETA: 0s - loss: 1459.4921
Epoch 48: val_loss did not improve from 1028.87610
99/99 [==============================] - 2s 23ms/step - loss: 1444.5565 - val_los
s: 1086.2560
Epoch 49/100
97/99 [============================>.] - ETA: 0s - loss: 1324.5466
Epoch 49: val_loss did not improve from 1028.87610
99/99 [==============================] - 2s 23ms/step - loss: 1312.2728 - val_los
s: 1181.9878
Epoch 50/100
97/99 [============================>.] - ETA: 0s - loss: 1365.6447
Epoch 50: val_loss improved from 1028.87610 to 1023.65765, saving model to ./tmp\c
heckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 1354.4451 - val_los
s: 1023.6577
Epoch 51/100
97/99 [============================>.] - ETA: 0s - loss: 1359.2201
Epoch 51: val_loss improved from 1023.65765 to 1021.91522, saving model to ./tmp\c
heckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 1361.3651 - val_los
s: 1021.9152
Epoch 52/100
97/99 [============================>.] - ETA: 0s - loss: 1400.5740
Epoch 52: val_loss did not improve from 1021.91522
99/99 [==============================] - 2s 23ms/step - loss: 1384.5225 - val_los
s: 1054.0020
Epoch 53/100
97/99 [============================>.] - ETA: 0s - loss: 1360.0934
Epoch 53: val_loss did not improve from 1021.91522
99/99 [==============================] - 2s 24ms/step - loss: 1354.3094 - val_los
s: 1280.8002
Epoch 54/100
97/99 [============================>.] - ETA: 0s - loss: 1370.4792
Epoch 54: val_loss did not improve from 1021.91522
99/99 [==============================] - 2s 24ms/step - loss: 1377.7965 - val_los
```

```
s: 1193.0270
Epoch 55/100
97/99 [============================>.] - ETA: 0s - loss: 1320.1320
Epoch 55: val_loss improved from 1021.91522 to 998.85254, saving model to ./tmp\ch
eckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 1311.8623 - val_los
s: 998.8525
Epoch 56/100
97/99 [============================>.] - ETA: 0s - loss: 1390.1060
Epoch 56: val_loss did not improve from 998.85254
99/99 [==============================] - 2s 24ms/step - loss: 1376.1422 - val_los
s: 1069.1898
Epoch 57/100
97/99 [============================>.] - ETA: 0s - loss: 1306.5199
Epoch 57: val_loss did not improve from 998.85254
99/99 [==============================] - 2s 24ms/step - loss: 1297.9812 - val_los
s: 1381.3689
Epoch 58/100
97/99 [============================>.] - ETA: 0s - loss: 1326.4186
Epoch 58: val_loss did not improve from 998.85254
99/99 [==============================] - 2s 23ms/step - loss: 1340.6998 - val_los
s: 1492.8315
Epoch 59/100
97/99 [============================>.] - ETA: 0s - loss: 1251.1320
Epoch 59: val_loss did not improve from 998.85254
99/99 [==============================] - 2s 23ms/step - loss: 1245.2216 - val_los
s: 1002.4512
Epoch 60/100
97/99 [============================>.] - ETA: 0s - loss: 1264.3192
Epoch 60: val_loss did not improve from 998.85254
99/99 [==============================] - 2s 24ms/step - loss: 1279.7291 - val_los
s: 1000.9149
Epoch 61/100
97/99 [============================>.] - ETA: 0s - loss: 1290.8635
Epoch 61: val_loss did not improve from 998.85254
99/99 [==============================] - 2s 24ms/step - loss: 1284.7278 - val_los
s: 1043.2782
Epoch 62/100
97/99 [============================>.] - ETA: 0s - loss: 1502.5223
Epoch 62: val_loss did not improve from 998.85254
99/99 [==============================] - 2s 23ms/step - loss: 1489.9105 - val_los
s: 1241.5518
Epoch 63/100
97/99 [============================>.] - ETA: 0s - loss: 1224.8746
Epoch 63: val_loss did not improve from 998.85254
99/99 [==============================] - 2s 24ms/step - loss: 1218.9043 - val_los
s: 1186.6465
Epoch 64/100
97/99 [============================>.] - ETA: 0s - loss: 1274.6855
Epoch 64: val_loss did not improve from 998.85254
99/99 [==============================] - 2s 23ms/step - loss: 1292.0513 - val_los
s: 1311.2971
Epoch 65/100
97/99 [============================>.] - ETA: 0s - loss: 1251.3997
Epoch 65: val_loss did not improve from 998.85254
99/99 [==============================] - 2s 23ms/step - loss: 1254.6912 - val_los
```

```
s: 1374.3438
Epoch 66/100
97/99 [============================>.] - ETA: 0s - loss: 1292.9016
Epoch 66: val_loss did not improve from 998.85254
99/99 [==============================] - 2s 23ms/step - loss: 1292.4977 - val_los
s: 1080.4169
Epoch 67/100
97/99 [============================>.] - ETA: 0s - loss: 1363.8981
Epoch 67: val_loss did not improve from 998.85254
99/99 [==============================] - 2s 24ms/step - loss: 1354.8452 - val_los
s: 1154.3430
Epoch 68/100
97/99 [============================>.] - ETA: 0s - loss: 1381.8081
Epoch 68: val_loss did not improve from 998.85254
99/99 [==============================] - 2s 23ms/step - loss: 1373.2101 - val_los
s: 1690.4812
Epoch 69/100
97/99 [============================>.] - ETA: 0s - loss: 1343.1292
Epoch 69: val_loss improved from 998.85254 to 943.26910, saving model to ./tmp\che
ckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 1334.4358 - val_los
s: 943.2691
Epoch 70/100
97/99 [============================>.] - ETA: 0s - loss: 1261.7444
Epoch 70: val_loss did not improve from 943.26910
99/99 [==============================] - 2s 23ms/step - loss: 1266.0175 - val_los
s: 1410.7622
Epoch 71/100
97/99 [============================>.] - ETA: 0s - loss: 1267.6425
Epoch 71: val_loss did not improve from 943.26910
99/99 [==============================] - 2s 23ms/step - loss: 1276.0065 - val_los
s: 1426.8193
Epoch 72/100
97/99 [============================>.] - ETA: 0s - loss: 1335.0004
Epoch 72: val_loss did not improve from 943.26910
99/99 [==============================] - 2s 23ms/step - loss: 1329.1245 - val_los
s: 1247.9089
Epoch 73/100
97/99 [============================>.] - ETA: 0s - loss: 1222.7552
Epoch 73: val_loss did not improve from 943.26910
99/99 [==============================] - 2s 23ms/step - loss: 1226.2208 - val_los
s: 1213.8613
Epoch 74/100
97/99 [============================>.] - ETA: 0s - loss: 1273.9613
Epoch 74: val_loss did not improve from 943.26910
99/99 [==============================] - 2s 23ms/step - loss: 1267.1825 - val_los
s: 1178.2267
Epoch 75/100
97/99 [============================>.] - ETA: 0s - loss: 1353.1821
Epoch 75: val_loss did not improve from 943.26910
99/99 [==============================] - 2s 23ms/step - loss: 1341.7770 - val_los
s: 950.1476
Epoch 76/100
97/99 [============================>.] - ETA: 0s - loss: 1321.3445
Epoch 76: val_loss improved from 943.26910 to 938.28510, saving model to ./tmp\che
ckpoint1
```

```
99/99 [==============================] - 3s 26ms/step - loss: 1314.4153 - val_los
s: 938.2851
Epoch 77/100
97/99 [============================>.] - ETA: 0s - loss: 1215.0144
Epoch 77: val_loss did not improve from 938.28510
99/99 [==============================] - 2s 24ms/step - loss: 1232.5220 - val_los
s: 1316.0547
Epoch 78/100
97/99 [============================>.] - ETA: 0s - loss: 1209.2529
Epoch 78: val_loss did not improve from 938.28510
99/99 [==============================] - 2s 23ms/step - loss: 1205.7981 - val_los
s: 995.8564
Epoch 79/100
97/99 [============================>.] - ETA: 0s - loss: 1173.7175
Epoch 79: val_loss did not improve from 938.28510
99/99 [==============================] - 2s 23ms/step - loss: 1180.4390 - val_los
s: 1295.3560
Epoch 80/100
97/99 [============================>.] - ETA: 0s - loss: 1250.6451
Epoch 80: val_loss did not improve from 938.28510
99/99 [==============================] - 2s 23ms/step - loss: 1249.4589 - val_los
s: 1033.6307
Epoch 81/100
97/99 [============================>.] - ETA: 0s - loss: 1192.4326
Epoch 81: val_loss did not improve from 938.28510
99/99 [==============================] - 2s 23ms/step - loss: 1191.9651 - val_los
s: 979.3018
Epoch 82/100
97/99 [============================>.] - ETA: 0s - loss: 1282.1741
Epoch 82: val_loss did not improve from 938.28510
99/99 [==============================] - 2s 23ms/step - loss: 1284.0363 - val_los
s: 1077.5154
Epoch 83/100
97/99 [============================>.] - ETA: 0s - loss: 1235.5325
Epoch 83: val_loss did not improve from 938.28510
99/99 [==============================] - 2s 23ms/step - loss: 1236.5139 - val_los
s: 1045.7056
Epoch 84/100
97/99 [============================>.] - ETA: 0s - loss: 1231.5883
Epoch 84: val_loss did not improve from 938.28510
99/99 [==============================] - 2s 24ms/step - loss: 1217.9836 - val_los
s: 1187.8184
Epoch 85/100
97/99 [============================>.] - ETA: 0s - loss: 1246.7794
Epoch 85: val_loss did not improve from 938.28510
99/99 [==============================] - 2s 23ms/step - loss: 1232.7632 - val_los
s: 1110.5825
Epoch 86/100
97/99 [============================>.] - ETA: 0s - loss: 1151.5029
Epoch 86: val_loss did not improve from 938.28510
99/99 [==============================] - 2s 23ms/step - loss: 1155.4165 - val_los
s: 1189.0972
Epoch 87/100
97/99 [============================>.] - ETA: 0s - loss: 1210.0145
Epoch 87: val_loss did not improve from 938.28510
99/99 [==============================] - 2s 24ms/step - loss: 1219.5201 - val_los
```

```
s: 1189.1774
Epoch 88/100
97/99 [============================>.] - ETA: 0s - loss: 1225.9098
Epoch 88: val_loss did not improve from 938.28510
99/99 [==============================] - 2s 24ms/step - loss: 1224.3751 - val_los
s: 1183.8450
Epoch 89/100
97/99 [============================>.] - ETA: 0s - loss: 1303.2488
Epoch 89: val_loss improved from 938.28510 to 917.28430, saving model to ./tmp\che
ckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 1293.1726 - val_los
s: 917.2843
Epoch 90/100
97/99 [============================>.] - ETA: 0s - loss: 1228.6052
Epoch 90: val_loss improved from 917.28430 to 915.03528, saving model to ./tmp\che
ckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 1216.7367 - val_los
s: 915.0353
Epoch 91/100
97/99 [============================>.] - ETA: 0s - loss: 1312.5813
Epoch 91: val_loss did not improve from 915.03528
99/99 [==============================] - 2s 24ms/step - loss: 1306.0142 - val_los
s: 1191.5292
Epoch 92/100
97/99 [============================>.] - ETA: 0s - loss: 1198.3763
Epoch 92: val_loss did not improve from 915.03528
99/99 [==============================] - 2s 23ms/step - loss: 1183.2498 - val_los
s: 952.7643
Epoch 93/100
97/99 [============================>.] - ETA: 0s - loss: 1238.5533
Epoch 93: val_loss did not improve from 915.03528
99/99 [==============================] - 2s 23ms/step - loss: 1232.7961 - val_los
s: 949.6810
Epoch 94/100
97/99 [============================>.] - ETA: 0s - loss: 1317.2937
Epoch 94: val_loss did not improve from 915.03528
99/99 [==============================] - 2s 24ms/step - loss: 1327.9763 - val_los
s: 1949.3805
Epoch 95/100
97/99 [============================>.] - ETA: 0s - loss: 1377.9698
Epoch 95: val_loss did not improve from 915.03528
99/99 [==============================] - 2s 24ms/step - loss: 1375.8728 - val_los
s: 1164.4780
Epoch 96/100
97/99 [============================>.] - ETA: 0s - loss: 1126.2793
Epoch 96: val_loss did not improve from 915.03528
99/99 [==============================] - 2s 23ms/step - loss: 1117.2419 - val_los
s: 1042.5039
Epoch 97/100
97/99 [============================>.] - ETA: 0s - loss: 1171.2009
Epoch 97: val_loss did not improve from 915.03528
99/99 [==============================] - 2s 24ms/step - loss: 1164.8180 - val_los
s: 1091.7428
Epoch 98/100
97/99 [============================>.] - ETA: 0s - loss: 1350.1179
Epoch 98: val_loss did not improve from 915.03528
```

```
99/99 [==============================] - 2s 24ms/step - loss: 1347.5385 - val_los
s: 972.9324
Epoch 99/100
97/99 [============================>.] - ETA: 0s - loss: 1280.2053
Epoch 99: val_loss improved from 915.03528 to 913.56488, saving model to ./tmp\che
ckpoint1
99/99 [==============================] - 3s 26ms/step - loss: 1275.2301 - val_los
s: 913.5649
Epoch 100/100
97/99 [============================>.] - ETA: 0s - loss: 1311.1982
Epoch 100: val_loss did not improve from 913.56488
99/99 [==============================] - 2s 24ms/step - loss: 1304.4565 - val_los
s: 1218.6624
```

In [37]:

```
new_model.load_weights(new_checkpoint_filepath)
```

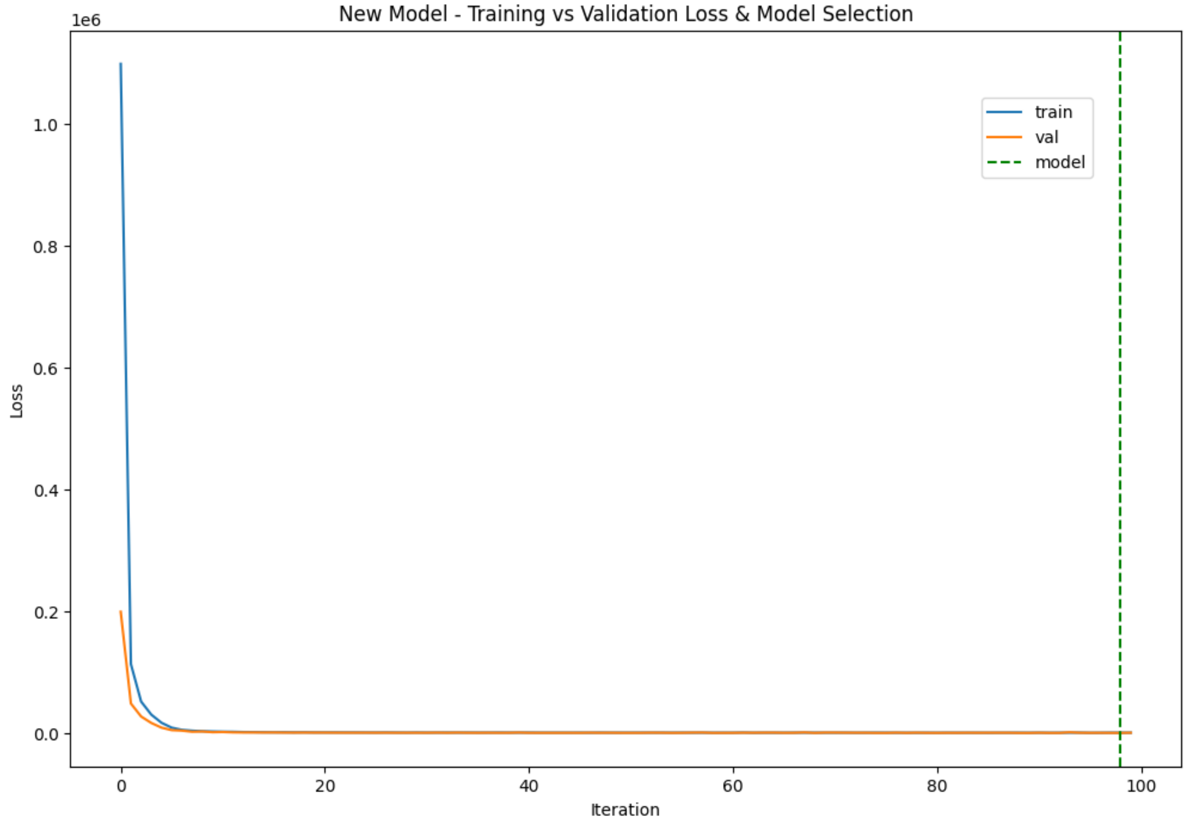
Out[37]:

```
<tensorflow.python.checkpoint.checkpoint.CheckpointLoadStatus at 0x1caa9852f20>
```

In [53]:

```
lowest = np.amin(new_history.history['val_loss'])
time = np.where(new_history.history['val_loss'] == lowest)


fig = plt.figure(figsize = (12,8))
fig = plt.plot(new_history.history['loss'])
fig = plt.plot(new_history.history['val_loss'])
fig = plt.axvline(x = time, color = 'green', linestyle = '--')
plt.title(
    'New Model - Training vs Validation Loss & Model Selection')
plt.ylabel('Loss')
plt.xlabel('Iteration')
plt.legend(['train', 'val', 'model'], loc= (0.82, 0.8))
plt.show()
```



5.5 Model Evaluation

In [38]:
```
test_loss = new_model.evaluate(X_test1, y_test1)
print(f"Test MSLE Loss: {test_loss}")
```

```
8/8 [==============================] - 1s 93ms/step - loss: 724.2055
Test MSLE Loss: 724.2055053710938
```

In [39]:
```
new_preds = new_model.predict(X_test1)
new_preds = new_preds.flatten()
new_preds_round = [round(val, 0) for val in new_preds]
```

```
8/8 [==============================] - 0s 16ms/step
```

In [40]:
```
print(f"Mean Absolute Error:
      {mean_absolute_error(y_test, new_preds_round)}\n")
print(f"Mean Squared Error:
      {mean_squared_error(y_test, new_preds_round)}\n")
print(f"Root Mean Squared Error:
      {math.sqrt(mean_squared_error(y_test, new_preds_round))}\n")
```

```
Mean Absolute Error: 21.83467741935484

Mean Squared Error: 723.3991935483871

Root Mean Squared Error: 26.89608137904827
```

In [58]:
```
new_out = pd.DataFrame({"Y_test" : y_test1,
                        "Predictions" : new_preds_round,
                        "Difference" : y_test1 - new_preds_round},
```

```
                             columns =
                              ["Y_test", "Predictions", "Difference"])
new_out.sort_values(['Y_test'])
```

Out[58]:

	Y_test	Predictions	Difference
9	1913.0	1976.0	-63.0
87	1928.0	1983.0	-55.0
154	1928.0	1966.0	-38.0
130	1933.0	1973.0	-40.0
221	1933.0	1974.0	-41.0
...	...	...	...
211	2016.0	1991.0	25.0
80	2016.0	1986.0	30.0
114	2016.0	1909.0	107.0
236	2016.0	1987.0	29.0
53	2016.0	1980.0	36.0

248 rows × 3 columns

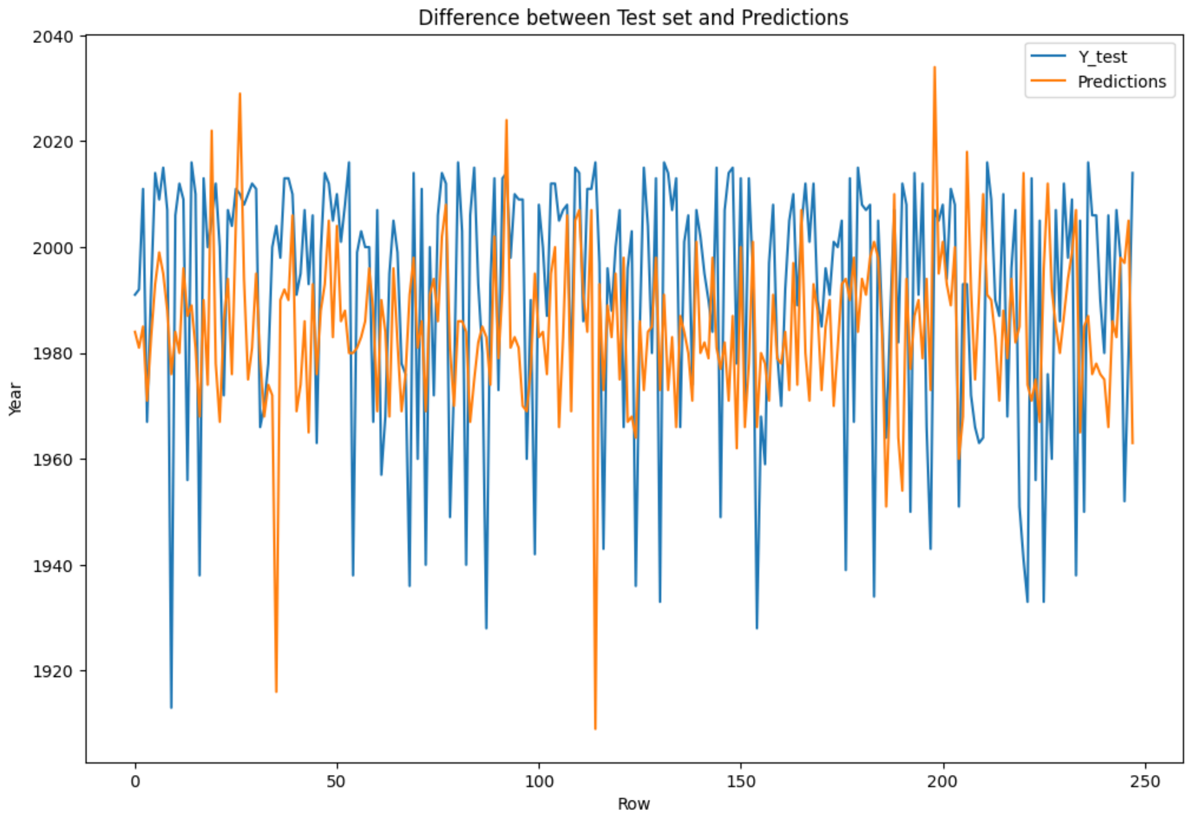
In [69]:

```
fig = plt.figure(figsize = (12,8))
plt.plot(new_out[['Y_test','Predictions']])
plt.legend(['Y_test', 'Predictions'], loc= 'upper right')
plt.xlabel('Row')
plt.ylabel('Year')
plt.title("Difference between Test set and Predictions")
plt.suptitle('New Model')

plt.plot()
```

Out[69]: []

New Model



5.6 Model Comparison

A direct comparison between both the models suggest that the new trained model works well with the data, and the major reason being that this model was designed to specifically work with the `Poster` dataset. The Adadelta helped with the decaying learning rate factor which improved the training of course, but the final verdict is seen in Part 5.6 where both are compared. The Pretrained model has imagenet weights and so it doesn't fit quite well when given to do a regression task. On the other hand, our new model has effectively proven both the regression task and also resource management as it took less time to train the model.

It's usually better to use a pretrained model for Classification task as that saves a lot of time, but Regression is a task that usually takes a whole model into consideration and so requires a different weight and needs to be trained together with the convolution layers. So for Regression, it's better to train a new model rather than using a Pre-trained model.

In [79]:
```
fig = plt.figure(figsize = (15,6))

plt.subplot(1,2,1)
plt.plot(out_df[['Y_test','Predictions']])
plt.legend(['Y_test', 'Predictions'], loc= 'lower right')
plt.xlabel('Row')
plt.ylabel('Year')
plt.title(
    "Pretrained Model - Difference b/w Test set and Predictions")
```

```python
plt.subplot(1,2,2)
plt.plot(new_out[['Y_test','Predictions']])
plt.legend(['Y_test', 'Predictions'],
           loc= 'lower right')
plt.xlabel('Row')
plt.ylabel('Year')
plt.title(
    "New Model - Difference b/w Test set and Predictions")

plt.plot()
plt.plot()
```

Out[79]: []