R Notebook

Research Question 2

Question: What is the effect of different strategies to simultaneously learn one model from multiple TrD's?

For answering this question we need to evaluate the effect of the transfer lerning methods (MN, M1, M2, M3, MF) and the simple model (S) on the score. We will then perform significance tests to see if there is a significant difference between the methods. We will aslo see how The Methods perform on specific tasks/TeD.

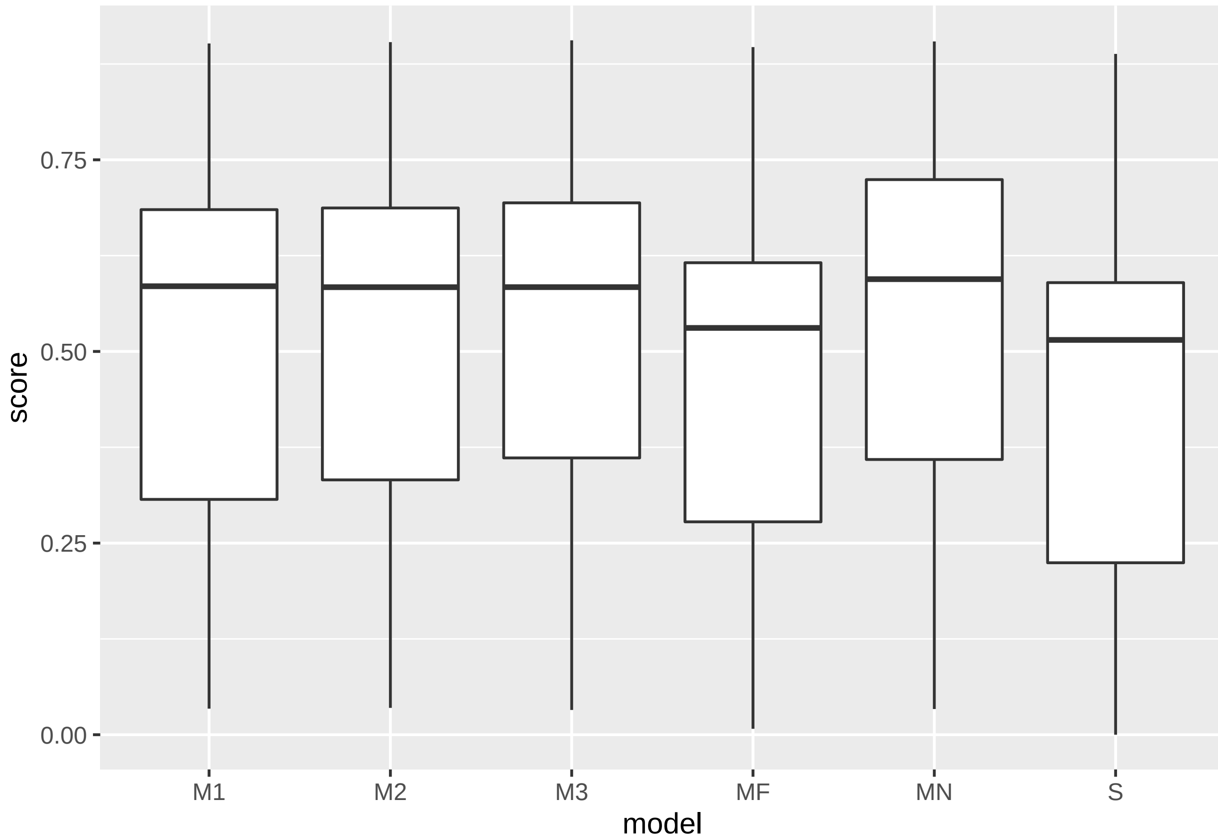
Reading the data

We read the data and create the resevent subset (d) for the current research quetsion. We are **not** going to analyse the baselines here because we *only want to observe effect of different strategies when learning from* **multiple TrD**

```r
library('ggplot2')
library('emmeans')
data = read.csv('../data/data.csv')
d = subset(data, model %in% c('MN', 'M1', 'M2', 'M3', 'MF', 'S'))
```

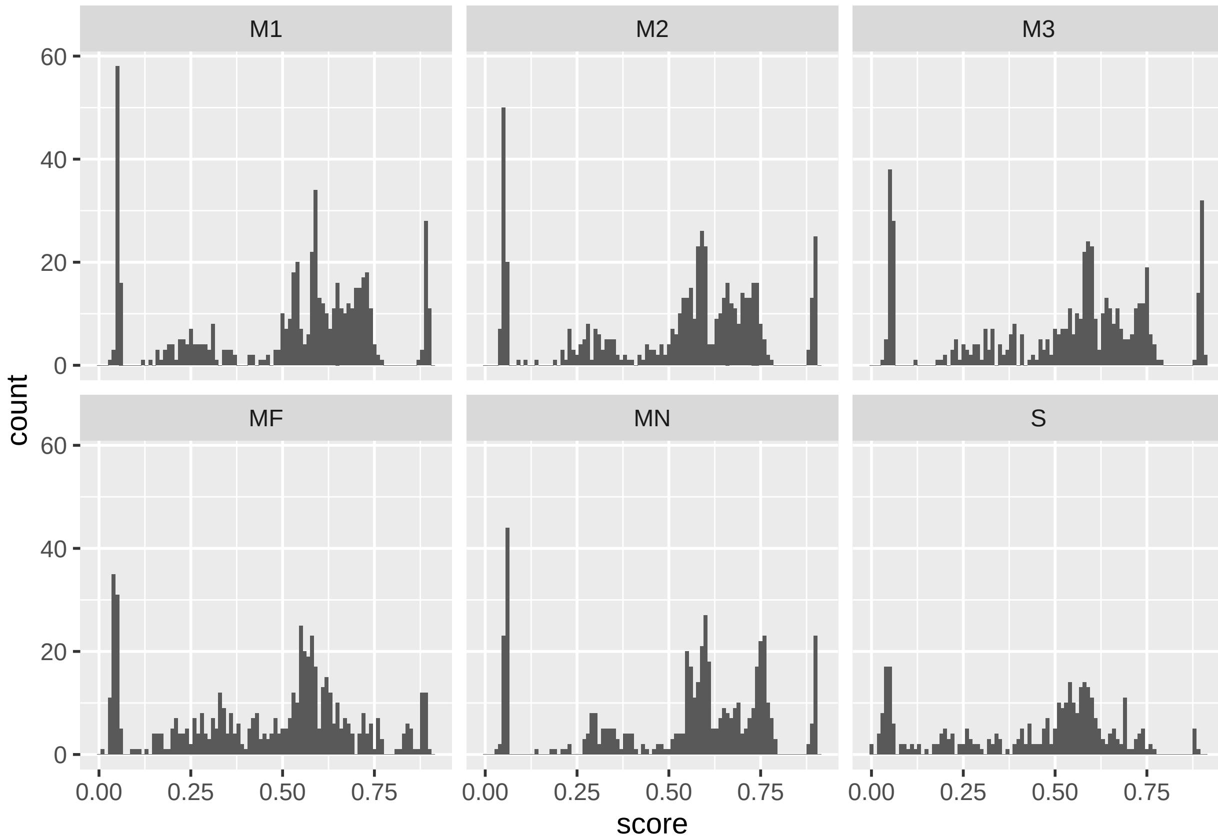
Observing the distribution of scores by models

```r
ggplot(d, aes(model, score)) + geom_boxplot(notch = FALSE)
```



We can see that transfer learning methods' mean performance is better over the simple model *(disregarding all the other independent variables and the interaction effects)*. But we need stronger tests than this.

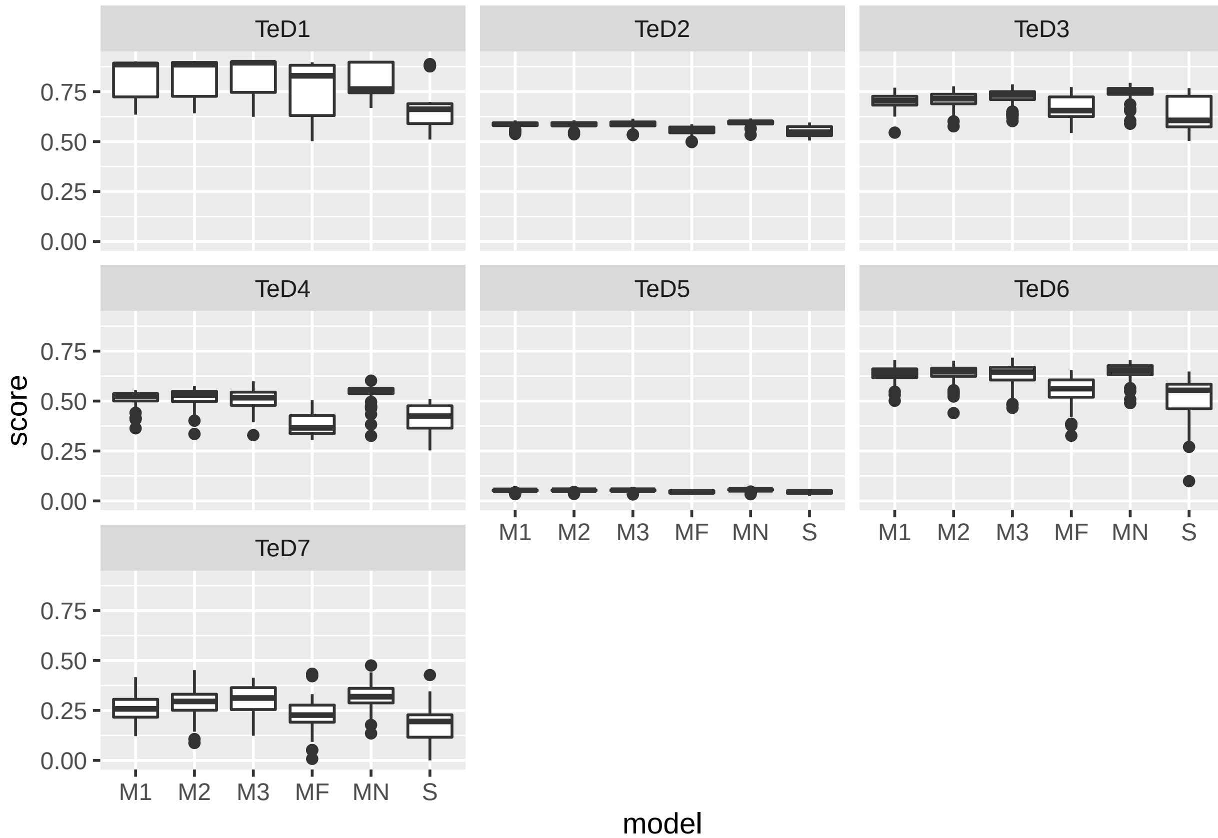
```
ggplot(d, aes(score)) + geom_histogram(binwidth = 0.01) + facet_wrap(~model)
```



We see that the scores for the methods are not normally distributed. Thus, we cannot depend on the mean alone.

Let us observe the effect of other independent variables (test dataset) on the score

```
ggplot(d, aes(model, score)) + geom_boxplot(notch = FALSE) + facet_wrap(~TeD)
```



We see that except TeD5, the use of different test datasets and tasks result in observably significant variations in performance in different models.

Simple Significance Tests

Because the 5 Mx methods were implimented using randamization in selection of training set we are not considering the effect of selection of the training set here. Our purpose here is to **only compare methods against each other considering consistent testing strategy for evaluation on the same test set/task and consistent (emperically comparable) scoring method**.

Lets create a simple linear model and observe the coefficients and confidence intervals.

```
m = lm(score ~ model+TeD, d)
summary(m)
```

```
##
## Call:
## lm(formula = score ~ model + TeD, data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44115 -0.02813 -0.00022  0.03989  0.22644
##
## Coefficients:
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  0.802059   0.003828  209.528  < 2e-16 ***
## modelM2      0.005737   0.003747    1.531  0.12585
```

```
## modelM3      0.011600   0.003812    3.043  0.00236 **
## modelMF     -0.054194   0.003711  -14.603  < 2e-16 ***
## modelMN      0.022119   0.003840    5.760 9.26e-09 ***
## modelS      -0.080192   0.004279  -18.743  < 2e-16 ***
## TeDTeD2     -0.212136   0.004233  -50.115  < 2e-16 ***
## TeDTeD3     -0.089711   0.004233  -21.193  < 2e-16 ***
## TeDTeD4     -0.308013   0.004233  -72.765  < 2e-16 ***
## TeDTeD5     -0.739268   0.004233 -174.643  < 2e-16 ***
## TeDTeD6     -0.182050   0.004233  -43.007  < 2e-16 ***
## TeDTeD7     -0.520991   0.004233 -123.078  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06171 on 2963 degrees of freedom
## Multiple R-squared:  0.9386, Adjusted R-squared:  0.9384
## F-statistic:  4120 on 11 and 2963 DF,  p-value: < 2.2e-16
```

```r
confint(m)
```

```
##                     2.5 %      97.5 %
## (Intercept)  0.794553364  0.80956472
## modelM2     -0.001609790  0.01308316
## modelM3      0.004126552  0.01907403
## modelMF     -0.061470820 -0.04691716
## modelMN      0.014589781  0.02964789
## modelS      -0.088581303 -0.07180280
## TeDTeD2     -0.220435645 -0.20383575
## TeDTeD3     -0.098010994 -0.08141110
## TeDTeD4     -0.316313333 -0.29971343
## TeDTeD5     -0.747567696 -0.73096780
## TeDTeD6     -0.190349527 -0.17374963
## TeDTeD7     -0.529290562 -0.51269066
```

We see that the both Method of training and choice of TeD have a significant effect on the score. Therefore it is also wise to perform an interaction analysis.

Interaction Effect

Let us observe the interaction effect of the TeD and method on the score.

```r
m = lm(score ~ model*TeD, d)
summary(m)
```

```
##
## Call:
## lm(formula = score ~ model * TeD, data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41416 -0.02131  0.00255  0.02837  0.24426
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.813362   0.006583 123.562  < 2e-16 ***
## modelM2         -0.001235   0.009339  -0.132 0.894768
## modelM3          0.024685   0.009501   2.598 0.009420 **
```

```
## modelMF         -0.055233   0.009251  -5.971 2.65e-09 ***
## modelMN         -0.002631   0.009571  -0.275 0.783398
## modelS          -0.150884   0.010665 -14.148  < 2e-16 ***
## TeDTeD2         -0.228385   0.009309 -24.533  < 2e-16 ***
## TeDTeD3         -0.111086   0.009309 -11.933  < 2e-16 ***
## TeDTeD4         -0.300258   0.009309 -32.254  < 2e-16 ***
## TeDTeD5         -0.761399   0.009309 -81.790  < 2e-16 ***
## TeDTeD6         -0.176892   0.009309 -19.002  < 2e-16 ***
## TeDTeD7         -0.553264   0.009309 -59.432  < 2e-16 ***
## modelM2:TeDTeD2  0.001042   0.013208   0.079 0.937102
## modelM3:TeDTeD2 -0.022875   0.013437  -1.702 0.088787 .
## modelMF:TeDTeD2  0.024920   0.013083   1.905 0.056899 .
## modelMN:TeDTeD2  0.012305   0.013536   0.909 0.363392
## modelS:TeDTeD2   0.117038   0.015083   7.760 1.17e-14 ***
## modelM2:TeDTeD3  0.008719   0.013208   0.660 0.509219
## modelM3:TeDTeD3 -0.007320   0.013437  -0.545 0.585951
## modelMF:TeDTeD3  0.020738   0.013083   1.585 0.113032
## modelMN:TeDTeD3  0.045473   0.013536   3.359 0.000791 ***
## modelS:TeDTeD3   0.085371   0.015083   5.660 1.66e-08 ***
## modelM2:TeDTeD4  0.004270   0.013208   0.323 0.746497
## modelM3:TeDTeD4 -0.029303   0.013437  -2.181 0.029273 *
## modelMF:TeDTeD4 -0.075271   0.013083  -5.754 9.64e-09 ***
## modelMN:TeDTeD4  0.030419   0.013536   2.247 0.024698 *
## modelS:TeDTeD4   0.049532   0.015083   3.284 0.001035 **
## modelM2:TeDTeD5  0.001597   0.013208   0.121 0.903760
## modelM3:TeDTeD5 -0.024242   0.013437  -1.804 0.071306 .
## modelMF:TeDTeD5  0.047596   0.013083   3.638 0.000279 ***
## modelMN:TeDTeD5  0.005911   0.013536   0.437 0.662349
## modelS:TeDTeD5   0.141808   0.015083   9.402  < 2e-16 ***
## modelM2:TeDTeD6  0.001360   0.013208   0.103 0.917966
## modelM3:TeDTeD6 -0.028562   0.013437  -2.126 0.033612 *
## modelMF:TeDTeD6 -0.032200   0.013083  -2.461 0.013901 *
## modelMN:TeDTeD6  0.014689   0.013536   1.085 0.277938
## modelS:TeDTeD6   0.027243   0.015083   1.806 0.070986 .
## modelM2:TeDTeD7  0.031816   0.013208   2.409 0.016063 *
## modelM3:TeDTeD7  0.020707   0.013437   1.541 0.123407
## modelMF:TeDTeD7  0.021487   0.013083   1.642 0.100612
## modelMN:TeDTeD7  0.064454   0.013536   4.762 2.01e-06 ***
## modelS:TeDTeD7   0.073851   0.015083   4.896 1.03e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05814 on 2933 degrees of freedom
## Multiple R-squared:  0.9461, Adjusted R-squared:  0.9453
## F-statistic:  1255 on 41 and 2933 DF,  p-value: < 2.2e-16
```

We can see that indeed there is significant interaction effect. We cannot do further analysis just using coefficients. We need to separate instances depending on TeD and see how the models comparatively perform.

Pairwise comparisons for Models

We need to perform pairwise comparisons while making adjustments for experiment design issues (interaction effects of TeD).

```r
m = lm (score ~model*TeD, d)
emmeans(m, ~model)
```

```
## NOTE: Results may be misleading due to involvement in interactions

##  model emmean      SE   df lower.CL upper.CL
##  M1     0.509 0.00249 2933    0.504    0.514
##  M2     0.515 0.00250 2933    0.510    0.520
##  M3     0.520 0.00259 2933    0.515    0.526
##  MF     0.455 0.00246 2933    0.450    0.460
##  MN     0.531 0.00263 2933    0.526    0.536
##  S      0.429 0.00317 2933    0.422    0.435
##
## Results are averaged over the levels of: TeD
## Confidence level used: 0.95
```

The estimated marginal means(EMMs) tell us that the mean method performance, averaged over the test dataset used, is in the order:

$$MN > M3 > M2 > M1 > MF > S$$

We can check pairwise comparison of the methods for understanding how significant these rankings are.

```r
emmeans(m, pairwise~model)
```

```
## NOTE: Results may be misleading due to involvement in interactions

## $emmeans
##  model emmean      SE   df lower.CL upper.CL
##  M1     0.509 0.00249 2933    0.504    0.514
##  M2     0.515 0.00250 2933    0.510    0.520
##  M3     0.520 0.00259 2933    0.515    0.526
##  MF     0.455 0.00246 2933    0.450    0.460
##  MN     0.531 0.00263 2933    0.526    0.536
##  S      0.429 0.00317 2933    0.422    0.435
##
## Results are averaged over the levels of: TeD
## Confidence level used: 0.95
##
## $contrasts
##  contrast estimate      SE   df t.ratio p.value
##  M1 - M2  -0.00574 0.00353 2933  -1.625 0.5818
##  M1 - M3  -0.01160 0.00359 2933  -3.230 0.0158
##  M1 - MF   0.05419 0.00350 2933  15.500 <.0001
##  M1 - MN  -0.02212 0.00362 2933  -6.114 <.0001
##  M1 - S    0.08019 0.00403 2933  19.894 <.0001
##  M2 - M3  -0.00586 0.00360 2933  -1.628 0.5801
##  M2 - MF   0.05993 0.00351 2933  17.084 <.0001
##  M2 - MN  -0.01638 0.00363 2933  -4.515 0.0001
##  M2 - S    0.08593 0.00404 2933  21.264 <.0001
##  M3 - MF   0.06579 0.00357 2933  18.432 <.0001
##  M3 - MN  -0.01052 0.00369 2933  -2.852 0.0500
##  M3 - S    0.09179 0.00409 2933  22.419 <.0001
##  MF - MN  -0.07631 0.00360 2933 -21.220 <.0001
##  MF - S    0.02600 0.00401 2933   6.480 <.0001
##  MN - S    0.10231 0.00412 2933  24.846 <.0001
##
## Results are averaged over the levels of: TeD
```

```
## P value adjustment: tukey method for comparing a family of 6 estimates
```

We observe the following:

- Using any transfer learning method (Mx) is better than using the simple model (S) i.e. (obs. $p < 0.0001$).
- Changing method between M1 to M2 (obs. $p = 0.5818$) OR M2 to M3 (obs. $p = 0.5801$) only gives slight improvements.
- Using MN over M3 gives marginally significant (obs. $p = 0.0500$) performance improvement.
- Using MN over {M1, M2, MF, S} gives highly significant (obs. $p < 0.0001$) performance improvement.

Pairwise comparison for Models per TeD

Lets also account for the performance of models depending on different tasks (TeD) the models have to perform.

Means for particular TeDs:

```r
emmeans(m, pairwise~model|TeD)$emmeans
```

```
## TeD = TeD1:
##  model emmean      SE   df lower.CL upper.CL
##  M1     0.8134 0.00658 2933   0.8005   0.8263
##  M2     0.8121 0.00663 2933   0.7991   0.8251
##  M3     0.8380 0.00685 2933   0.8246   0.8515
##  MF     0.7581 0.00650 2933   0.7454   0.7709
##  MN     0.8107 0.00695 2933   0.7971   0.8244
##  S      0.6625 0.00839 2933   0.6460   0.6789
##
## TeD = TeD2:
##  model emmean      SE   df lower.CL upper.CL
##  M1     0.5850 0.00658 2933   0.5721   0.5979
##  M2     0.5848 0.00663 2933   0.5718   0.5978
##  M3     0.5868 0.00685 2933   0.5734   0.6002
##  MF     0.5547 0.00650 2933   0.5419   0.5674
##  MN     0.5946 0.00695 2933   0.5810   0.6083
##  S      0.5511 0.00839 2933   0.5347   0.5676
##
## TeD = TeD3:
##  model emmean      SE   df lower.CL upper.CL
##  M1     0.7023 0.00658 2933   0.6894   0.7152
##  M2     0.7098 0.00663 2933   0.6968   0.7227
##  M3     0.7196 0.00685 2933   0.7062   0.7331
##  MF     0.6678 0.00650 2933   0.6550   0.6805
##  MN     0.7451 0.00695 2933   0.7315   0.7587
##  S      0.6368 0.00839 2933   0.6203   0.6532
##
## TeD = TeD4:
##  model emmean      SE   df lower.CL upper.CL
##  M1     0.5131 0.00658 2933   0.5002   0.5260
##  M2     0.5161 0.00663 2933   0.5031   0.5291
##  M3     0.5085 0.00685 2933   0.4951   0.5219
##  MF     0.3826 0.00650 2933   0.3699   0.3953
##  MN     0.5409 0.00695 2933   0.5273   0.5545
##  S      0.4118 0.00839 2933   0.3953   0.4282
##
```

```
## TeD = TeD5:
##  model emmean      SE   df lower.CL upper.CL
##  M1    0.0520 0.00658 2933   0.0391   0.0649
##  M2    0.0523 0.00663 2933   0.0393   0.0653
##  M3    0.0524 0.00685 2933   0.0390   0.0658
##  MF    0.0443 0.00650 2933   0.0316   0.0571
##  MN    0.0552 0.00695 2933   0.0416   0.0689
##  S     0.0429 0.00839 2933   0.0264   0.0593
##
## TeD = TeD6:
##  model emmean      SE   df lower.CL upper.CL
##  M1    0.6365 0.00658 2933   0.6236   0.6494
##  M2    0.6366 0.00663 2933   0.6236   0.6496
##  M3    0.6326 0.00685 2933   0.6192   0.6460
##  MF    0.5490 0.00650 2933   0.5363   0.5618
##  MN    0.6485 0.00695 2933   0.6349   0.6622
##  S     0.5128 0.00839 2933   0.4964   0.5293
##
## TeD = TeD7:
##  model emmean      SE   df lower.CL upper.CL
##  M1    0.2601 0.00658 2933   0.2472   0.2730
##  M2    0.2907 0.00663 2933   0.2777   0.3037
##  M3    0.3055 0.00685 2933   0.2921   0.3189
##  MF    0.2264 0.00650 2933   0.2136   0.2391
##  MN    0.3219 0.00695 2933   0.3083   0.3355
##  S     0.1831 0.00839 2933   0.1666   0.1995
##
## Confidence level used: 0.95
```

Comtrasts for particular TeDs:

```
emmeans(m, pairwise~model|TeD)$contrasts
```

```
## TeD = TeD1:
##  contrast  estimate      SE   df t.ratio p.value
##  M1 - M2   1.24e-03 0.00934 2933   0.132 1.0000
##  M1 - M3  -2.47e-02 0.00950 2933  -2.598 0.0980
##  M1 - MF   5.52e-02 0.00925 2933   5.971 <.0001
##  M1 - MN   2.63e-03 0.00957 2933   0.275 0.9998
##  M1 - S    1.51e-01 0.01067 2933  14.148 <.0001
##  M2 - M3  -2.59e-02 0.00953 2933  -2.720 0.0717
##  M2 - MF   5.40e-02 0.00928 2933   5.818 <.0001
##  M2 - MN   1.40e-03 0.00960 2933   0.145 1.0000
##  M2 - S    1.50e-01 0.01069 2933  13.997 <.0001
##  M3 - MF   7.99e-02 0.00944 2933   8.462 <.0001
##  M3 - MN   2.73e-02 0.00976 2933   2.799 0.0578
##  M3 - S    1.76e-01 0.01083 2933  16.207 <.0001
##  MF - MN  -5.26e-02 0.00951 2933  -5.528 <.0001
##  MF - S    9.57e-02 0.01061 2933   9.012 <.0001
##  MN - S    1.48e-01 0.01089 2933  13.608 <.0001
##
## TeD = TeD2:
##  contrast  estimate      SE   df t.ratio p.value
##  M1 - M2   1.93e-04 0.00934 2933   0.021 1.0000
##  M1 - M3  -1.81e-03 0.00950 2933  -0.191 1.0000
```

```
##  M1 - MF   3.03e-02 0.00925 2933   3.277 0.0136
##  M1 - MN  -9.67e-03 0.00957 2933  -1.011 0.9145
##  M1 - S    3.38e-02 0.01067 2933   3.174 0.0190
##  M2 - M3  -2.00e-03 0.00953 2933  -0.210 0.9999
##  M2 - MF   3.01e-02 0.00928 2933   3.245 0.0150
##  M2 - MN  -9.87e-03 0.00960 2933  -1.028 0.9087
##  M2 - S    3.37e-02 0.01069 2933   3.148 0.0206
##  M3 - MF   3.21e-02 0.00944 2933   3.401 0.0089
##  M3 - MN  -7.86e-03 0.00976 2933  -0.806 0.9665
##  M3 - S    3.57e-02 0.01083 2933   3.291 0.0129
##  MF - MN  -4.00e-02 0.00951 2933  -4.203 0.0004
##  MF - S    3.53e-03 0.01061 2933   0.333 0.9995
##  MN - S    4.35e-02 0.01089 2933   3.995 0.0009
##
## TeD = TeD3:
##  contrast  estimate      SE   df t.ratio p.value
##  M1 - M2  -7.48e-03 0.00934 2933  -0.801 0.9673
##  M1 - M3  -1.74e-02 0.00950 2933  -1.828 0.4479
##  M1 - MF   3.45e-02 0.00925 2933   3.729 0.0027
##  M1 - MN  -4.28e-02 0.00957 2933  -4.476 0.0001
##  M1 - S    6.55e-02 0.01067 2933   6.143 <.0001
##  M2 - M3  -9.88e-03 0.00953 2933  -1.037 0.9055
##  M2 - MF   4.20e-02 0.00928 2933   4.523 0.0001
##  M2 - MN  -3.54e-02 0.00960 2933  -3.683 0.0032
##  M2 - S    7.30e-02 0.01069 2933   6.828 <.0001
##  M3 - MF   5.19e-02 0.00944 2933   5.491 <.0001
##  M3 - MN  -2.55e-02 0.00976 2933  -2.611 0.0949
##  M3 - S    8.29e-02 0.01083 2933   7.650 <.0001
##  MF - MN  -7.73e-02 0.00951 2933  -8.128 <.0001
##  MF - S    3.10e-02 0.01061 2933   2.922 0.0409
##  MN - S    1.08e-01 0.01089 2933   9.946 <.0001
##
## TeD = TeD4:
##  contrast  estimate      SE   df t.ratio p.value
##  M1 - M2  -3.03e-03 0.00934 2933  -0.325 0.9995
##  M1 - M3   4.62e-03 0.00950 2933   0.486 0.9967
##  M1 - MF   1.31e-01 0.00925 2933  14.107 <.0001
##  M1 - MN  -2.78e-02 0.00957 2933  -2.903 0.0432
##  M1 - S    1.01e-01 0.01067 2933   9.503 <.0001
##  M2 - M3   7.65e-03 0.00953 2933   0.803 0.9670
##  M2 - MF   1.34e-01 0.00928 2933  14.388 <.0001
##  M2 - MN  -2.48e-02 0.00960 2933  -2.578 0.1029
##  M2 - S    1.04e-01 0.01069 2933   9.764 <.0001
##  M3 - MF   1.26e-01 0.00944 2933  13.330 <.0001
##  M3 - MN  -3.24e-02 0.00976 2933  -3.321 0.0117
##  M3 - S    9.67e-02 0.01083 2933   8.930 <.0001
##  MF - MN  -1.58e-01 0.00951 2933 -16.636 <.0001
##  MF - S   -2.92e-02 0.01061 2933  -2.747 0.0667
##  MN - S    1.29e-01 0.01089 2933  11.853 <.0001
##
## TeD = TeD5:
##  contrast  estimate      SE   df t.ratio p.value
##  M1 - M2  -3.62e-04 0.00934 2933  -0.039 1.0000
##  M1 - M3  -4.43e-04 0.00950 2933  -0.047 1.0000
```

```
##  M1 - MF   7.64e-03 0.00925 2933   0.825 0.9629
##  M1 - MN  -3.28e-03 0.00957 2933  -0.343 0.9994
##  M1 - S    9.08e-03 0.01067 2933   0.851 0.9577
##  M2 - M3  -8.15e-05 0.00953 2933  -0.009 1.0000
##  M2 - MF   8.00e-03 0.00928 2933   0.862 0.9555
##  M2 - MN  -2.92e-03 0.00960 2933  -0.304 0.9997
##  M2 - S    9.44e-03 0.01069 2933   0.883 0.9507
##  M3 - MF   8.08e-03 0.00944 2933   0.856 0.9568
##  M3 - MN  -2.84e-03 0.00976 2933  -0.291 0.9997
##  M3 - S    9.52e-03 0.01083 2933   0.879 0.9516
##  MF - MN  -1.09e-02 0.00951 2933  -1.147 0.8614
##  MF - S    1.44e-03 0.01061 2933   0.136 1.0000
##  MN - S    1.24e-02 0.01089 2933   1.134 0.8672
##
## TeD = TeD6:
##  contrast  estimate      SE   df t.ratio p.value
##  M1 - M2  -1.25e-04 0.00934 2933  -0.013 1.0000
##  M1 - M3   3.88e-03 0.00950 2933   0.408 0.9986
##  M1 - MF   8.74e-02 0.00925 2933   9.451 <.0001
##  M1 - MN  -1.21e-02 0.00957 2933  -1.260 0.8069
##  M1 - S    1.24e-01 0.01067 2933  11.593 <.0001
##  M2 - M3   4.00e-03 0.00953 2933   0.420 0.9983
##  M2 - MF   8.76e-02 0.00928 2933   9.434 <.0001
##  M2 - MN  -1.19e-02 0.00960 2933  -1.243 0.8157
##  M2 - S    1.24e-01 0.01069 2933  11.576 <.0001
##  M3 - MF   8.36e-02 0.00944 2933   8.848 <.0001
##  M3 - MN  -1.59e-02 0.00976 2933  -1.633 0.5767
##  M3 - S    1.20e-01 0.01083 2933  11.056 <.0001
##  MF - MN  -9.95e-02 0.00951 2933 -10.456 <.0001
##  MF - S    3.62e-02 0.01061 2933   3.411 0.0086
##  MN - S    1.36e-01 0.01089 2933  12.455 <.0001
##
## TeD = TeD7:
##  contrast  estimate      SE   df t.ratio p.value
##  M1 - M2  -3.06e-02 0.00934 2933  -3.274 0.0137
##  M1 - M3  -4.54e-02 0.00950 2933  -4.778 <.0001
##  M1 - MF   3.37e-02 0.00925 2933   3.648 0.0037
##  M1 - MN  -6.18e-02 0.00957 2933  -6.459 <.0001
##  M1 - S    7.70e-02 0.01067 2933   7.223 <.0001
##  M2 - M3  -1.48e-02 0.00953 2933  -1.554 0.6290
##  M2 - MF   6.43e-02 0.00928 2933   6.931 <.0001
##  M2 - MN  -3.12e-02 0.00960 2933  -3.254 0.0146
##  M2 - S    1.08e-01 0.01069 2933  10.065 <.0001
##  M3 - MF   7.91e-02 0.00944 2933   8.380 <.0001
##  M3 - MN  -1.64e-02 0.00976 2933  -1.684 0.5427
##  M3 - S    1.22e-01 0.01083 2933  11.301 <.0001
##  MF - MN  -9.56e-02 0.00951 2933 -10.044 <.0001
##  MF - S    4.33e-02 0.01061 2933   4.078 0.0007
##  MN - S    1.39e-01 0.01089 2933  12.745 <.0001
##
## P value adjustment: tukey method for comparing a family of 6 estimates
```

For TeD1 (Classification task)

Looking simply at means:

$$M3 > M1 > M2 > MN > MF > S$$

After looking at contrasts, with reference to the ranking above:

- There is significant difference on moving from S to Mx (obs. p < 0.0001)
- {MN, M1, M2, M3} result in similar performance ($obs.p \in [0.0578, 1.0]$)
- Only for this task MN performs worse than {M1, M2, M3}. But judging by previous point this is not too significant.

For TeD2 (Classification task)

Looking simply at means:

$$MN > M3 > M1 > M2 > MF > S$$

After looking at contrasts, with reference to the ranking above:

- {MN, M1, M2, M3} do not show much performance difference $obs.p \in [0.9087, 1.0]$)
- MF and MS perform similarly (obs. p = 0.9995)
- There is significant difference between {MN, M1, M2, M3} and {MF, MS} ($obs.p \in [0.0089, 0.01]$)

For TeD3 (Classification task)

Looking simply at means:

$$MN > M3 > M2 > M1 > MF > S$$

After looking at contrasts, with reference to the ranking above:

- In the above, going from MN to M1 with 1 step jump there isn't too much score change (obs. p > 0.09)
- There is a significant difference with a jump from M1 to MF (obs. p = 0.0027)
- Again, all Mx are better than S (obs. p < 0.0409)

For TeD4 (Classification task)

Looking simply at means:

$$MN > M2 > M1 > M3 > S > F$$

After looking at contrasts, with reference to the ranking above:

- {M1, M2, M3} perform similarly (obs. p > 0.9)
- MN performs significantly better than any other method (obs. p < 0.04)
- An oddity where S performs better than MF, *but with marginal insignificance (obs. p = 0.0667).*

For TeD5 (Recommendation task)

Looking simply at means:

$$MN > M3 > M2 > M1 > MF > S$$

After looking at contrasts, with reference to the ranking above:

- {M1, M2, M3, MF, MN, S} all perform very similarly (obs. p > 0.86).
- Overall **no significant different in score based on learning method used**.
- This meets our prior expectation from observing the data graphically.

For TeD6 (Regression task)

Looking simply at means:

$$MN > M2 > M1 > M3 > MF > S$$

After looking at contrasts, with reference to the ranking above:

- {MN, M1, M2, M3} perform sililarly (obs. $p > 0.57$)
- There is a stark degradation when juping from M3 to MF (obs. $p < 0.001$)
- Again, S is significantly inferior to all Mx (obs. $p < 0.008$)

For TeD7 (Regression task)

Looking simply at means:

$$MN > M3 > M2 > M1 > MF > S$$

After looking at contrasts, with reference to the ranking above:

- {MN, M2, M3} perform sililarly (obs. $p > 0.54$)
- Stark degredation in score on going from M2 to M1 (obs. $p = 0.0137$), showing thats tuning parts refined is give better performance.
- Again, S is significantly inferior (obs. $p < 0.0007$).

RQ-2 Conclusion

In general, transfer learning is better than a simple model trained with a single dataset (obs. $p < 0.0001$). No retraining for the specific TeD (MN) performed significantly better (obs. $p < 0.05$) than the simple model or any form of refininf.

For particular testsets and tasks transfer learning significantly was better in most cases (6 out of 7). For one recommendation task (TeD5) the choice of the methods did not show any significant difference in performance. For the regresion tasks no refinement (MN) worked very similarly (obs. $p > 0.86$) as refining 2 and 3 parts(M2, M3).