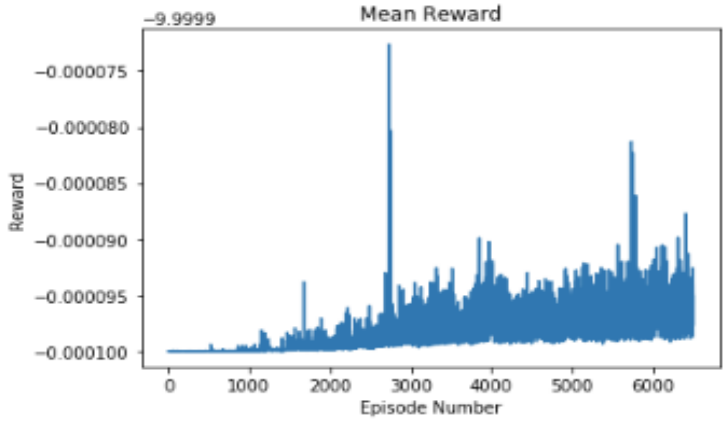
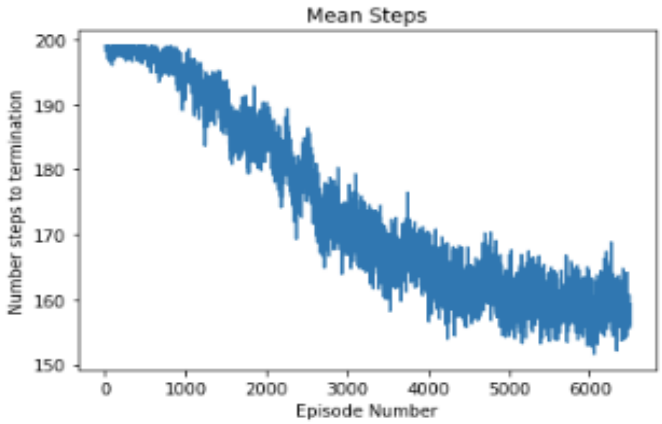
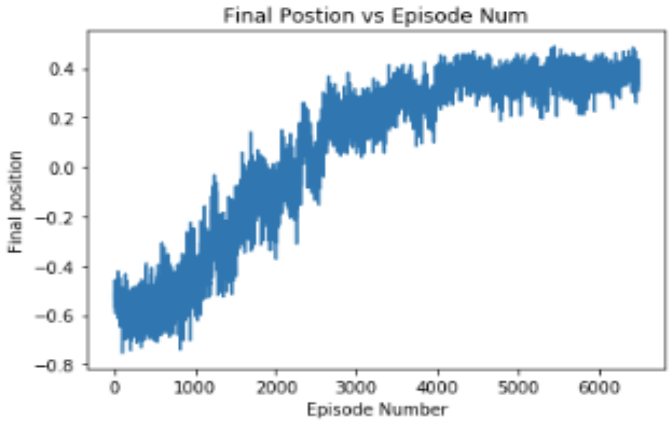
RL Assignment 2:
CCBR - RL Workshop.
Ravi Gupta
CS18B043
(I have included the relevant codes as q1_taxi.ipynb and q2_mountain.ipynb respectively in the zip folder.)

Question 2 relevant plots (Part a, Part b and Part c respectively.) These plots are obtained with 50 independent runs with each run, having 6500 episodes.

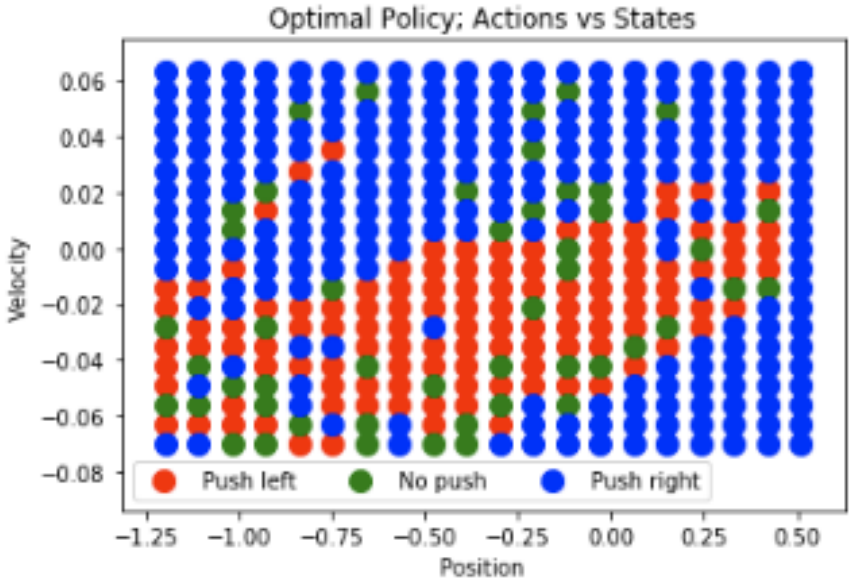




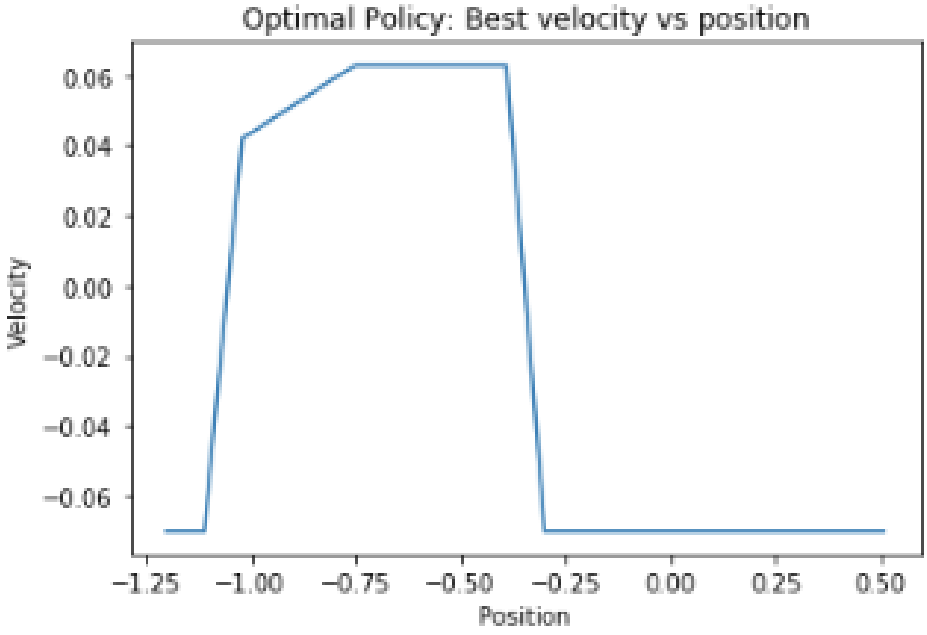


Question 2 (Part d)

This plot shows the best policy (best choice of action given for a particular state([position velocity]) tuple).



The following plot shows the plot of (Velocity with highest expected reward vs Position).



Question 1 (Taxi Problem)

a) Table of J*(s) for value iteration(on left) Table for \pi*(s) on right.

	A	B	C
0.00	8.000000	16.000000	7.000000
0.05	8.511527	16.400260	7.498869
0.10	9.076506	16.856369	8.050865
0.15	9.708121	17.464503	8.669160
0.20	10.437030	18.482143	9.384398
0.25	11.274074	19.629630	10.207407
0.30	12.243837	20.934066	11.162756
0.35	13.378714	22.430769	12.282824
0.40	14.722222	24.166667	13.611111
0.45	16.334131	26.205534	15.207371
0.50	18.298701	28.636364	17.155844
0.55	20.789989	31.607397	19.830725
0.60	24.025686	35.327724	23.458813
0.65	28.276692	40.096281	28.129979
0.70	34.061931	46.435416	34.366041
0.75	42.317411	55.285054	43.106317
0.80	55.079365	68.558201	56.269841
0.85	77.246512	90.811701	78.433456
0.90	121.653471	135.306276	122.836903
0.95	255.022908	268.764619	256.202849

J*

	A	B	C
0.00	1	1	1
0.05	1	1	1
0.10	1	1	1
0.15	1	2	1
0.20	1	2	1
0.25	1	2	1
0.30	1	2	1
0.35	1	2	1
0.40	1	2	1
0.45	1	2	1
0.50	1	2	1
0.55	1	2	2
0.60	1	2	2
0.65	1	2	2
0.70	1	2	2
0.75	1	2	2
0.80	2	2	2
0.85	2	2	2
0.90	2	2	2
0.95	2	2	2

\pi*

b) Table of Q*(s, a) for value iteration(on left) Table for \pi*(s) on right.

	A			B		C		
	1	2	3	1	2	1	2	3
0.00	7.427177	2.539811	4.234101	15.863845	15.116183	7.258074	3.843987	4.228700
0.05	8.547555	3.322070	4.512047	15.829690	15.523411	7.476628	4.613717	4.850772
0.10	8.895851	4.380142	5.057852	16.444044	16.086885	8.214452	5.333476	5.268947
0.15	9.524300	5.199579	6.069211	17.270629	16.998834	9.138125	6.126182	6.046186
0.20	10.312213	5.765740	6.714259	17.995575	17.673438	9.287023	7.203204	6.511637
0.25	11.064952	6.914617	7.194415	18.242793	19.372984	10.088877	8.141920	7.001846
0.30	12.111226	8.055273	7.761257	19.241180	21.779678	10.997648	9.424033	8.273041
0.35	13.224748	9.731249	8.852220	20.465723	22.570014	12.000980	10.961166	9.311260
0.40	15.023839	11.476057	10.296032	21.700999	25.482644	14.176802	12.800307	10.366009
0.45	16.276764	13.255345	12.027278	22.961278	26.215956	15.366218	14.517798	12.016972
0.50	18.065365	15.680276	13.287936	24.737133	27.297910	17.020054	16.425976	13.841088
0.55	20.605415	18.514380	15.615096	26.903729	30.121067	19.421141	19.142674	16.124843
0.60	23.773445	22.059822	18.694338	30.372075	34.430881	22.473762	23.234112	19.415917
0.65	28.481859	26.192303	22.766475	34.025429	39.358471	26.371902	27.547764	22.831894
0.70	33.562016	32.894618	29.219702	39.685119	45.192461	32.497523	33.532975	28.892284
0.75	42.519469	41.278927	37.539684	48.108750	54.527041	41.027804	43.745768	36.125331
0.80	55.210184	53.042591	50.262707	59.641031	67.056146	53.311122	56.764259	48.931181
0.85	74.153563	75.561587	71.195786	80.490257	88.984246	74.906730	77.066846	70.307476
0.90	118.205849	118.537542	114.793410	123.733916	134.234425	117.872165	119.957488	114.043023
0.95	250.854817	253.456228	248.032079	257.779586	265.655014	251.110201	254.054113	246.387334

Q*

	A	B	C
0.00	1	1	1
0.05	1	1	1
0.10	1	1	1
0.15	1	1	1
0.20	1	1	1
0.25	1	2	1
0.30	1	2	1
0.35	1	2	1
0.40	1	2	1
0.45	1	2	1
0.50	1	2	1
0.55	1	2	1
0.60	1	2	2
0.65	1	2	2
0.70	1	2	2
0.75	1	2	2
0.80	1	2	2
0.85	2	2	2
0.90	2	2	2
0.95	2	2	2

\pi*

c)
Table for policy which forces driver to go to the nearest taxi stand [2, 2, 2].
For this question I wasn't sure what to do, since TD learning is an evaluation scheme I didn't know how to use it directly to obtain the optimal policy. So I talked to the TA (Nirav) and he asked me to simply compute J values for the given policy and leave it at that.

	A	B	C
0.00	3.245897	15.595116	3.764823
0.05	3.401948	16.020231	4.537337
0.10	4.232484	16.069555	5.083271
0.15	4.773753	17.419229	6.154648
0.20	5.566269	18.387649	6.997948
0.25	7.065073	19.643256	8.242474
0.30	8.098521	20.607007	9.491314
0.35	9.110557	20.519571	10.158968
0.40	10.860666	23.726547	12.300443
0.45	13.007065	25.752240	14.019890
0.50	15.553469	27.650054	16.626794
0.55	18.630472	31.695481	19.862491
0.60	21.366447	34.115243	23.305322
0.65	26.503531	38.195566	27.910637
0.70	32.311683	45.831327	34.079886
0.75	41.738675	57.052330	42.564942
0.80	54.823783	69.101857	56.377794
0.85	77.968322	93.349940	79.282296
0.90	120.203271	136.505519	122.063868
0.95	251.218116	268.327766	252.646016

J_{\pi}(s) where \pi = [2,2,2]