

Model on FakeNewsCorpus without metadata

	accuracy	precision macro	recall macro	f1-score macro	support macro	precision weighted	recall weighted	f1-score weighted	support weighted
_ngram_2_LogisticRegression(solver='sag')_	0.874373	0.902019	0.818084	0.843615	13962.0	0.884888	0.874373	0.867151	13962.0
_ngram_3_LogisticRegression(max_iter=10000, n_j...	0.882467	0.922142	0.823413	0.852045	13962.0	0.898419	0.882467	0.874812	13962.0
_ngram_2_SVC(kernel='linear')_	0.875591	0.916948	0.813555	0.842405	13962.0	0.892685	0.875591	0.866946	13962.0
_ngram_3_SVC(kernel='linear')_	0.882037	0.922295	0.822598	0.851358	13962.0	0.898296	0.882037	0.874274	13962.0
_ngram_2_DecisionTreeClassifier()_	0.834766	0.815642	0.806092	0.810469	13962.0	0.832868	0.834766	0.833497	13962.0
_ngram_3_DecisionTreeClassifier()_	0.876092	0.903146	0.820686	0.846047	13962.0	0.886276	0.876092	0.869126	13962.0
_ngramKNN_2k=1	0.800530	0.775152	0.770312	0.772602	13962.0	0.798972	0.800530	0.799645	13962.0
_ngramKNN_2k=3	0.805758	0.781718	0.773777	0.777418	13962.0	0.803514	0.805758	0.804370	13962.0
_ngramKNN_2k=5	0.847801	0.856640	0.792751	0.813031	13962.0	0.850906	0.847801	0.840392	13962.0
_ngramKNN_2k=7	0.854677	0.876139	0.794589	0.818448	13962.0	0.862994	0.854677	0.845970	13962.0
_ngramKNN_2k=10	0.858187	0.896031	0.791605	0.819099	13962.0	0.874276	0.858187	0.847635	13962.0
_ngramKNN_3k=1	0.866710	0.882400	0.814285	0.836333	13962.0	0.872205	0.866710	0.860261	13962.0
_ngramKNN_3k=3	0.874158	0.903336	0.817045	0.842987	13962.0	0.885392	0.874158	0.866728	13962.0
_ngramKNN_3k=5	0.876808	0.910998	0.818254	0.845489	13962.0	0.890320	0.876808	0.869096	13962.0
_ngramKNN_3k=7	0.878456	0.916354	0.818825	0.846971	13962.0	0.893705	0.878456	0.870526	13962.0
_ngramKNN_3k=10	0.877883	0.917581	0.817299	0.845845	13962.0	0.894046	0.877883	0.869694	13962.0
TFIDF_LogisticRegression(solver='sag')_	0.898582	0.913193	0.857695	0.877902	13962.0	0.902943	0.898582	0.894954	13962.0
TFIDF_SVC(kernel='linear')_	0.909612	0.920698	0.874610	0.892401	13962.0	0.912529	0.909612	0.907004	13962.0
TFIDF_DecisionTreeClassifier()_	0.863916	0.847159	0.843898	0.845486	13962.0	0.863343	0.863916	0.863595	13962.0
_KNN_1	0.705057	0.827833	0.554710	0.509637	13962.0	0.782704	0.705057	0.614685	13962.0
_KNN_3	0.693454	0.841510	0.536048	0.474195	13962.0	0.788743	0.693454	0.589417	13962.0
_KNN_5	0.690302	0.841890	0.531223	0.464873	13962.0	0.788235	0.690302	0.582737	13962.0
_KNN_7	0.689156	0.841490	0.529488	0.461508	13962.0	0.787700	0.689156	0.580322	13962.0
_KNN_10	0.686936	0.840719	0.526127	0.454931	13962.0	0.786667	0.686936	0.575607	13962.0

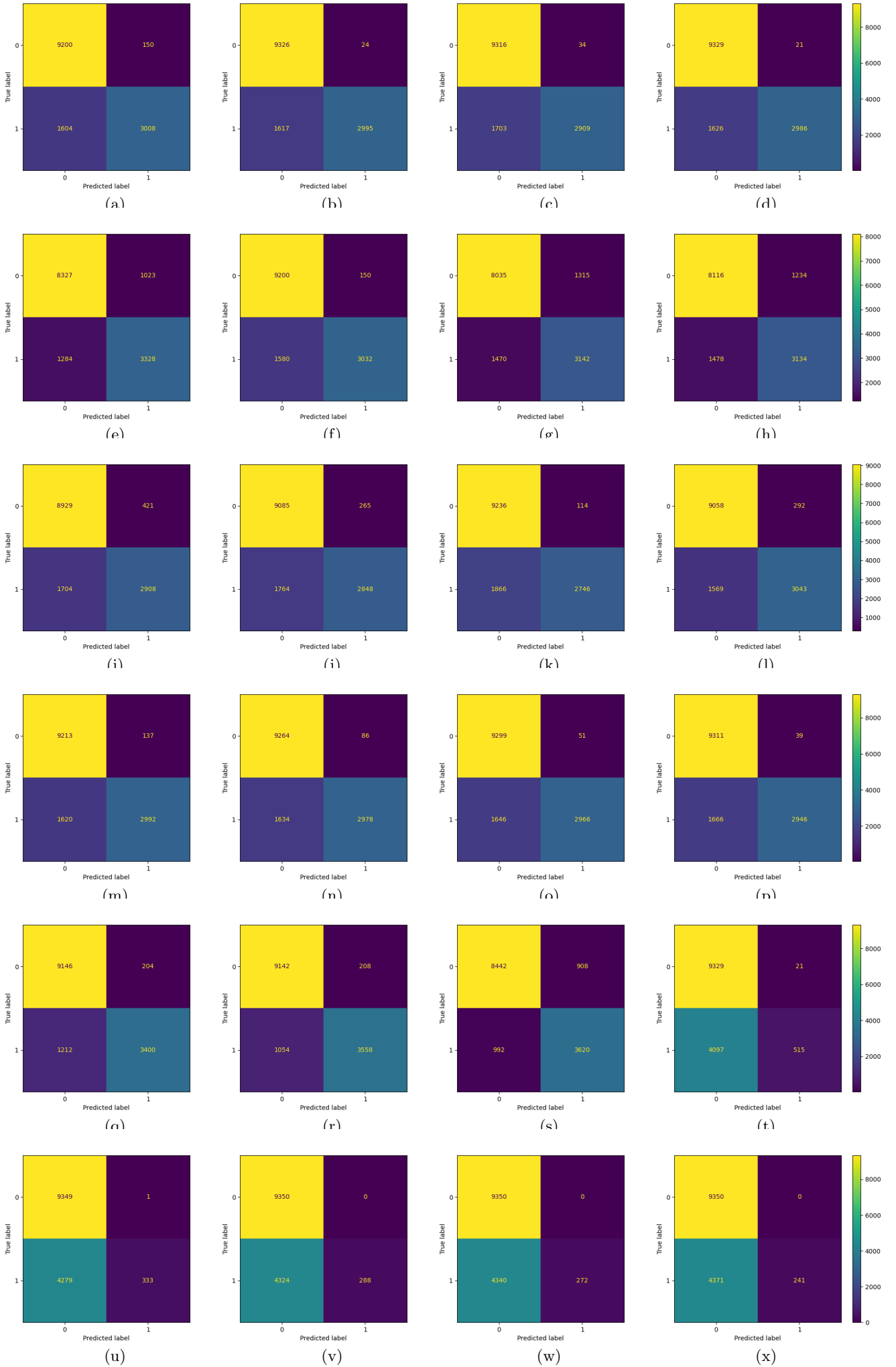


Figure 1: The models used in the above confusion matrices are respectively (a) bigram vectoriser with logistic regression; (b)

Model on FakeNewsCorpus with metadata

	accuracy	precision macro	recall macro	f1-score macro	support macro	precision weighted	recall weighted	f1-score weighted	support weighted
_ngram_2_LogisticRegression(solver='sag')_	0.884830	0.912307	0.832155	0.857550	13962.0	0.894907	0.884830	0.878704	13962.0
_ngram_3_LogisticRegression(max_iter=10000, n_j...	0.894284	0.928872	0.841301	0.868671	13962.0	0.907292	0.894284	0.888353	13962.0
_ngram_2_SVC(kernel='linear')_	0.894284	0.925916	0.842675	0.869183	13962.0	0.905917	0.894284	0.888629	13962.0
_ngram_3_SVC(kernel='linear')_	0.894857	0.927656	0.842883	0.869730	13962.0	0.907008	0.894857	0.889145	13962.0
_ngram_2_DecisionTreeClassifier()_	0.850451	0.832249	0.827527	0.829794	13962.0	0.849532	0.850451	0.849916	13962.0
_ngram_3_DecisionTreeClassifier()_	0.887480	0.906629	0.840176	0.862746	13962.0	0.893879	0.887480	0.882518	13962.0
_ngramKNN_2k=1	0.805472	0.780017	0.781419	0.780707	13962.0	0.805976	0.805472	0.805715	13962.0
_ngramKNN_2k=3	0.841642	0.833421	0.798920	0.812012	13962.0	0.839557	0.841642	0.837339	13962.0
_ngramKNN_2k=5	0.853531	0.856413	0.805105	0.822894	13962.0	0.854427	0.853531	0.847891	13962.0
_ngramKNN_2k=7	0.858473	0.871117	0.804949	0.826206	13962.0	0.862904	0.858473	0.851618	13962.0
_ngramKNN_2k=10	0.865492	0.899091	0.803323	0.830218	13962.0	0.879167	0.865492	0.856480	13962.0
_ngramKNN_3k=1	0.873012	0.879554	0.828990	0.847195	13962.0	0.874955	0.873012	0.868509	13962.0
_ngramKNN_3k=3	0.884687	0.908703	0.833751	0.858073	13962.0	0.893230	0.884687	0.878929	13962.0
_ngramKNN_3k=5	0.888555	0.917867	0.836199	0.862092	13962.0	0.899335	0.888555	0.882592	13962.0
_ngramKNN_3k=7	0.889271	0.921634	0.835690	0.862464	13962.0	0.901432	0.889271	0.883069	13962.0
_ngramKNN_3k=10	0.890274	0.927191	0.834955	0.862974	13962.0	0.904551	0.890274	0.883729	13962.0
TFIDF_LogisticRegression(solver='sag')_	0.930239	0.940848	0.901932	0.917764	13962.0	0.932659	0.930239	0.928633	13962.0
TFIDF_SVC(kernel='linear')_	0.935826	0.940826	0.913465	0.925267	13962.0	0.936712	0.935826	0.934800	13962.0
TFIDF_DecisionTreeClassifier()_	0.911259	0.901007	0.897650	0.899292	13962.0	0.910945	0.911259	0.911073	13962.0
_KNN_1	0.751898	0.848188	0.626820	0.625280	13962.0	0.808496	0.751898	0.699198	13962.0
_KNN_3	0.742945	0.856689	0.611456	0.602087	13962.0	0.811310	0.742945	0.682427	13962.0
_KNN_5	0.740152	0.856837	0.607063	0.595355	13962.0	0.810635	0.740152	0.677497	13962.0
_KNN_7	0.738433	0.859057	0.604131	0.590707	13962.0	0.811581	0.738433	0.674151	13962.0
_KNN_10	0.736929	0.858980	0.601800	0.587073	13962.0	0.811125	0.736929	0.671488	13962.0

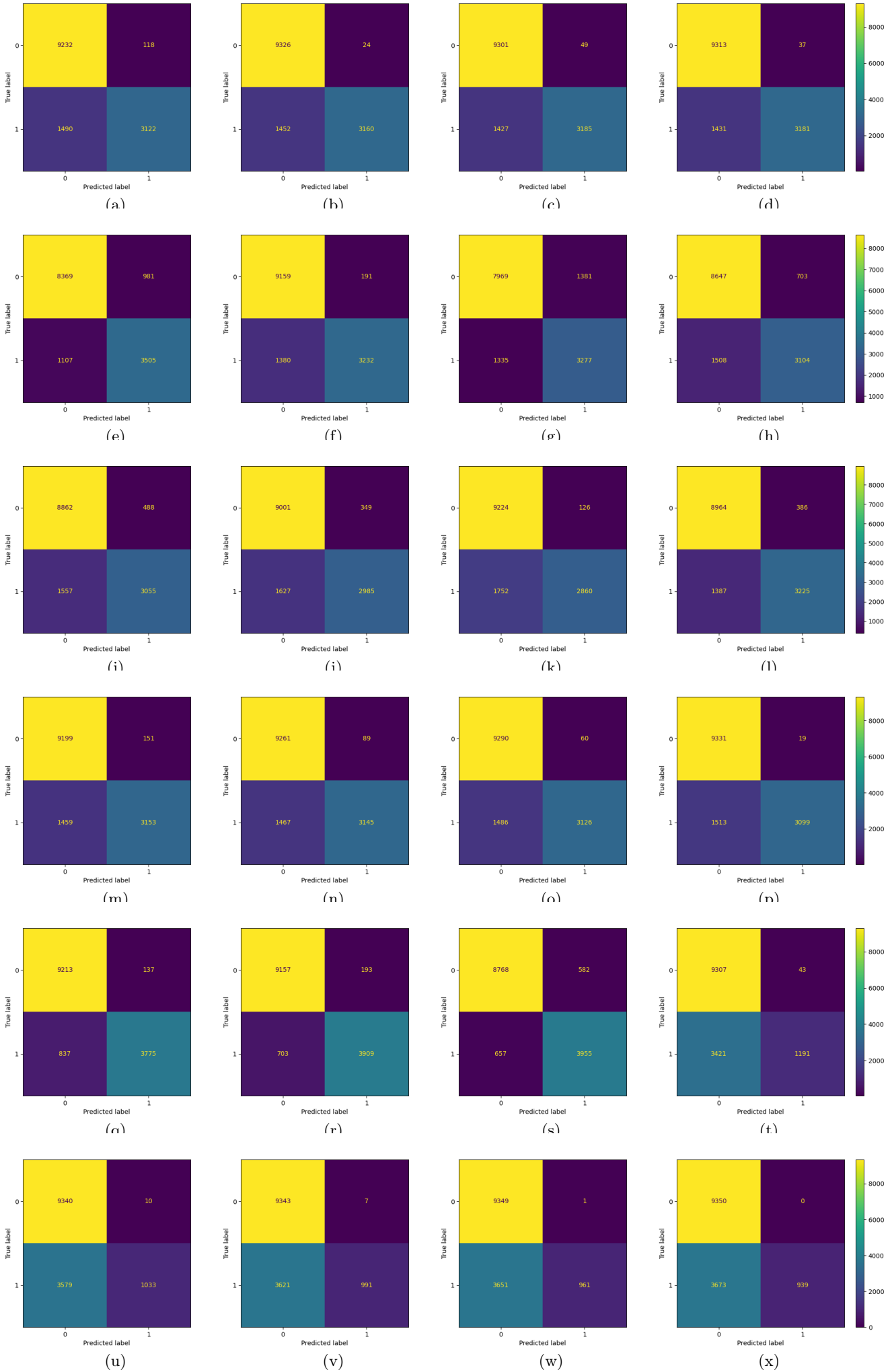


Figure 2: The models used in the above confusion matrices are respectively (a) bigram vectoriser with logistic regression; (b)

Model on FakeNewsCorpus only with metadata

	accuracy	precision macro	recall macro	f1-score macro	support macro	precision weighted	recall weighted	f1-score weighted	support weighted
_ngram_2_LogisticRegression(solver='sag')_	0.700043	0.726276	0.557064	0.521941	13962.0	0.716176	0.700043	0.620961	13962.0
_ngram_3_LogisticRegression(max_iter=10000, n_j...	0.696390	0.787860	0.543460	0.491161	13962.0	0.754512	0.696390	0.600824	13962.0
_ngram_2_SVC(kernel='linear')_	0.700115	0.733636	0.555909	0.519031	13962.0	0.720823	0.700115	0.619180	13962.0
_ngram_3_SVC(kernel='linear')_	0.696963	0.780121	0.544986	0.494583	13962.0	0.749654	0.696963	0.603115	13962.0
_ngram_2_DecisionTreeClassifier()_	0.694600	0.682052	0.556736	0.527063	13962.0	0.687055	0.694600	0.622585	13962.0
_ngram_3_DecisionTreeClassifier()_	0.695459	0.753605	0.544742	0.496100	13962.0	0.732126	0.695459	0.603657	13962.0
_ngramKNN_2k=1	0.666022	0.590653	0.548584	0.534026	13962.0	0.625926	0.666022	0.618187	13962.0
_ngramKNN_2k=3	0.681206	0.624777	0.555196	0.535758	13962.0	0.649324	0.681206	0.623939	13962.0
_ngramKNN_2k=5	0.566967	0.605512	0.614386	0.564911	13962.0	0.670318	0.566967	0.575062	13962.0
_ngramKNN_2k=7	0.573199	0.604754	0.615028	0.570060	13962.0	0.668322	0.573199	0.582526	13962.0
_ngramKNN_2k=10	0.580862	0.603589	0.615256	0.575968	13962.0	0.665544	0.580862	0.591427	13962.0
_ngramKNN_3k=1	0.692881	0.712095	0.545289	0.501137	13962.0	0.704855	0.692881	0.606091	13962.0
_ngramKNN_3k=3	0.693597	0.724868	0.544725	0.498674	13962.0	0.713178	0.693597	0.604756	13962.0
_ngramKNN_3k=5	0.694528	0.742832	0.544212	0.495986	13962.0	0.724935	0.694528	0.603334	13962.0
_ngramKNN_3k=7	0.695602	0.755072	0.544849	0.496187	13962.0	0.733110	0.695602	0.603750	13962.0
_ngramKNN_3k=10	0.695889	0.774268	0.543635	0.492408	13962.0	0.745574	0.695889	0.601468	13962.0
TFIDF_LogisticRegression(solver='sag')_	0.802965	0.815412	0.726698	0.746686	13962.0	0.808411	0.802965	0.787204	13962.0
TFIDF_SVC(kernel='linear')_	0.802822	0.812601	0.728074	0.747723	13962.0	0.807027	0.802822	0.787732	13962.0
TFIDF_DecisionTreeClassifier()_	0.752757	0.720096	0.709153	0.713751	13962.0	0.747879	0.752757	0.749609	13962.0
_KNN_1	0.525068	0.620562	0.610956	0.524071	13962.0	0.695769	0.525068	0.516679	13962.0
_KNN_3	0.742301	0.730131	0.647892	0.657283	13962.0	0.736419	0.742301	0.715208	13962.0
_KNN_5	0.726185	0.702752	0.628608	0.634495	13962.0	0.714758	0.726185	0.696618	13962.0
_KNN_7	0.730053	0.724738	0.621497	0.624162	13962.0	0.727236	0.730053	0.691860	13962.0
_KNN_10	0.740223	0.801197	0.615686	0.611297	13962.0	0.775730	0.740223	0.687264	13962.0

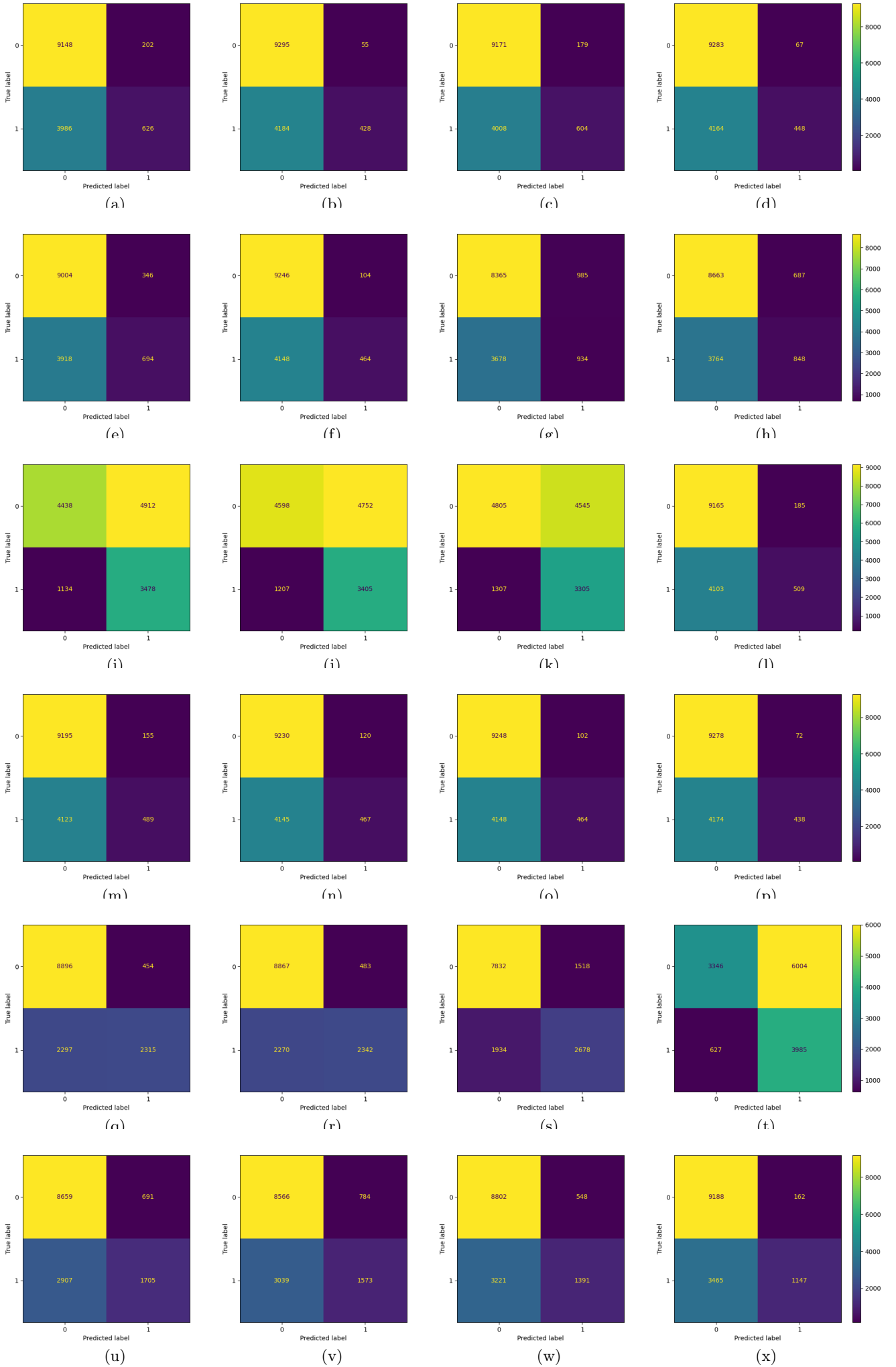


Figure 3: The models used in the above confusion matrices are respectively (a) bigram vectoriser with logistic regression; (b)

Model on LIAR dataset without metadata

	accuracy	precision macro	recall macro	f1-score macro	support macro	precision weighted	recall weighted	f1-score weighted	support weighted
_ngram_2_LogisticRegression(solver='sag')_	0.721970	0.461087	0.499924	0.419544	12790.0	0.577117	0.721970	0.605743	12790.0
_ngram_3_LogisticRegression(max_iter=10000, n_j...	0.722205	0.361102	0.500000	0.419349	12790.0	0.521580	0.722205	0.605712	12790.0
_ngram_2_SVC(kernel='linear')_	0.722205	0.361102	0.500000	0.419349	12790.0	0.521580	0.722205	0.605712	12790.0
_ngram_3_SVC(kernel='linear')_	0.722127	0.527776	0.500032	0.419597	12790.0	0.614188	0.722127	0.605819	12790.0
_ngram_2_DecisionTreeClassifier()_	0.701720	0.478201	0.495950	0.440303	12790.0	0.585887	0.701720	0.610294	12790.0
_ngram_3_DecisionTreeClassifier()_	0.722048	0.486098	0.499978	0.419570	12790.0	0.591022	0.722048	0.605781	12790.0
_ngramKNN_2k=1	0.676153	0.507580	0.503708	0.482340	12790.0	0.603705	0.676153	0.623106	12790.0
_ngramKNN_2k=3	0.491634	0.507236	0.508973	0.473177	12790.0	0.606211	0.491634	0.517000	12790.0
_ngramKNN_2k=5	0.708991	0.495212	0.499338	0.439102	12790.0	0.595969	0.708991	0.612011	12790.0
_ngramKNN_2k=7	0.716341	0.504014	0.500270	0.430399	12790.0	0.601029	0.716341	0.609752	12790.0
_ngramKNN_2k=10	0.721501	0.521144	0.500206	0.421289	12790.0	0.610534	0.721501	0.606526	12790.0
_ngramKNN_3k=1	0.721188	0.549989	0.501028	0.424405	12790.0	0.626707	0.721188	0.608085	12790.0
_ngramKNN_3k=3	0.721814	0.537623	0.500249	0.420854	12790.0	0.619697	0.721814	0.606391	12790.0
_ngramKNN_3k=5	0.721267	0.444357	0.499610	0.419851	12790.0	0.567766	0.721267	0.605689	12790.0
_ngramKNN_3k=7	0.722127	0.361092	0.499946	0.419323	12790.0	0.521564	0.722127	0.605674	12790.0
_ngramKNN_3k=10	0.722127	0.361092	0.499946	0.419323	12790.0	0.521564	0.722127	0.605674	12790.0
_TFIDF_LogisticRegression(solver='sag')_	0.718061	0.503412	0.500162	0.427207	12790.0	0.600675	0.718061	0.608600	12790.0
_TFIDF_SVC(kernel='linear')_	0.718139	0.533220	0.502121	0.432874	12790.0	0.617590	0.718139	0.611625	12790.0
_TFIDF_DecisionTreeClassifier()_	0.645113	0.491020	0.493909	0.483198	12790.0	0.592464	0.645113	0.611753	12790.0
_KNN_1	0.721423	0.454796	0.499718	0.419904	12790.0	0.573585	0.721423	0.605766	12790.0
_KNN_3	0.722205	0.361102	0.500000	0.419349	12790.0	0.521580	0.722205	0.605712	12790.0
_KNN_5	0.722205	0.361102	0.500000	0.419349	12790.0	0.521580	0.722205	0.605712	12790.0
_KNN_7	0.722205	0.361102	0.500000	0.419349	12790.0	0.521580	0.722205	0.605712	12790.0
_KNN_10	0.722205	0.361102	0.500000	0.419349	12790.0	0.521580	0.722205	0.605712	12790.0

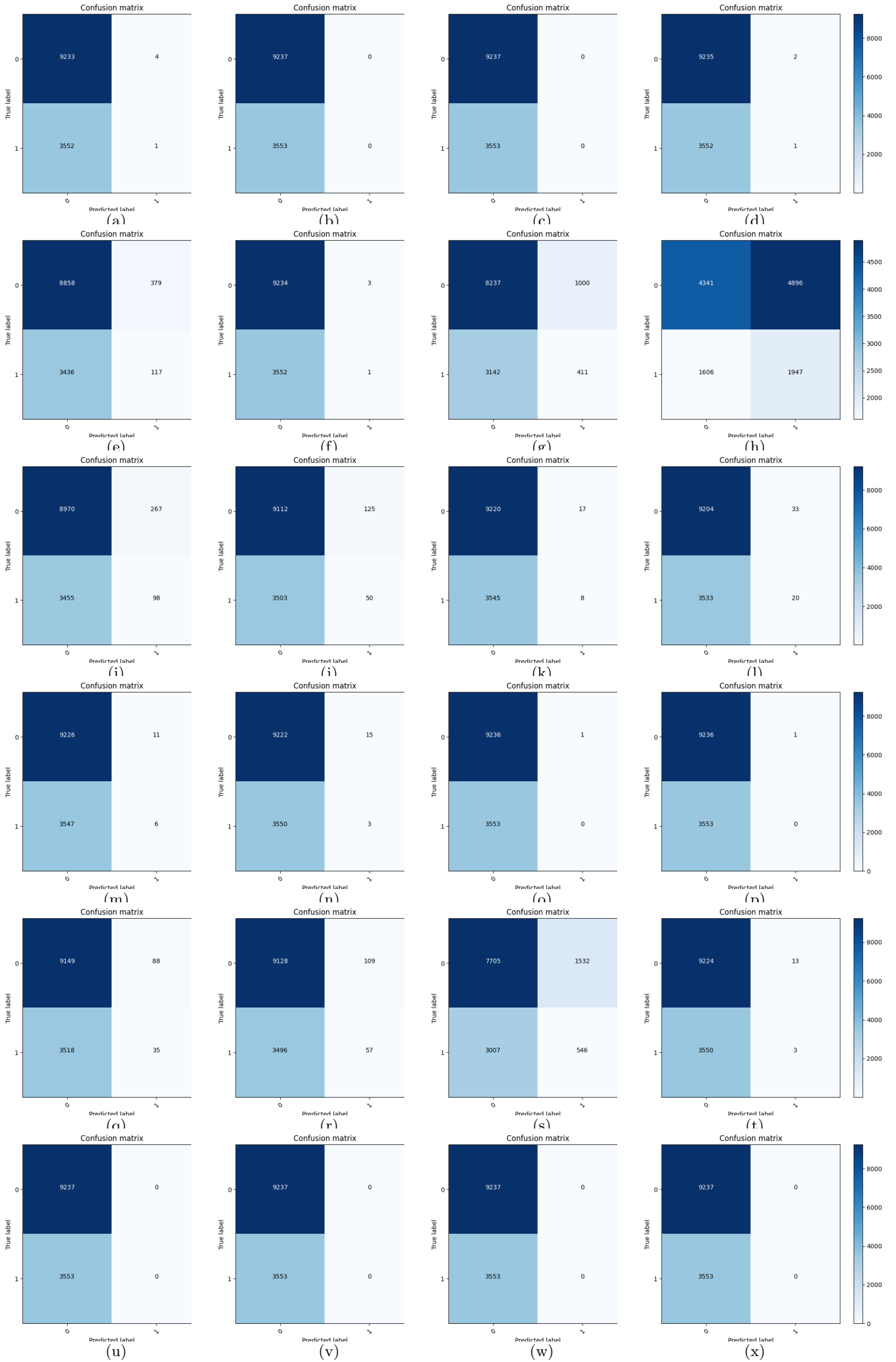


Figure 4: The models used in the above confusion matrices are respectively (a) bigram vectoriser with logistic regression; (b)

Model on LIAR dataset with metadata

	accuracy	precision macro	recall macro	f1-score macro	support macro	precision weighted	recall weighted	f1-score weighted	support weighted
_ngram_2_LogisticRegression(solver='sag')_	0.720641	0.360885	0.498917	0.418821	12790.0	0.521266	0.720641	0.604950	12790.0
_ngram_3_LogisticRegression(max_iter=10000, n_j...	0.722205	0.361102	0.500000	0.419349	12790.0	0.521580	0.722205	0.605712	12790.0
_ngram_2_SVC(kernel='linear')_	0.722205	0.361102	0.500000	0.419349	12790.0	0.521580	0.722205	0.605712	12790.0
_ngram_3_SVC(kernel='linear')_	0.722127	0.527776	0.500032	0.419597	12790.0	0.614188	0.722127	0.605819	12790.0
_ngram_2_DecisionTreeClassifier()_	0.699062	0.487595	0.497227	0.446998	12790.0	0.591339	0.699062	0.612919	12790.0
_ngram_3_DecisionTreeClassifier()_	0.721970	0.361070	0.499838	0.419270	12790.0	0.521533	0.721970	0.605597	12790.0
_ngramKNN_2k=1	0.686083	0.494476	0.498026	0.463055	12790.0	0.595299	0.686083	0.616845	12790.0
_ngramKNN_2k=3	0.715481	0.506844	0.500541	0.432579	12790.0	0.602650	0.715481	0.610634	12790.0
_ngramKNN_2k=5	0.712119	0.501822	0.500205	0.436856	12790.0	0.599800	0.712119	0.611827	12790.0
_ngramKNN_2k=7	0.716028	0.504409	0.500313	0.431036	12790.0	0.601257	0.716028	0.609990	12790.0
_ngramKNN_2k=10	0.722127	0.601300	0.500985	0.422588	12790.0	0.655208	0.722127	0.607410	12790.0
_ngramKNN_3k=1	0.719859	0.506975	0.500195	0.424194	12790.0	0.602660	0.719859	0.607561	12790.0
_ngramKNN_3k=3	0.721970	0.578635	0.500704	0.421993	12790.0	0.642565	0.721970	0.607045	12790.0
_ngramKNN_3k=5	0.721267	0.444357	0.499610	0.419851	12790.0	0.567766	0.721267	0.605689	12790.0
_ngramKNN_3k=7	0.721892	0.486094	0.499957	0.419790	12790.0	0.591016	0.721892	0.605850	12790.0
_ngramKNN_3k=10	0.722205	0.361102	0.500000	0.419349	12790.0	0.521580	0.722205	0.605712	12790.0
_TFIDF_LogisticRegression(solver='sag')_	0.719390	0.460919	0.499090	0.421631	12790.0	0.576874	0.719390	0.606055	12790.0
_TFIDF_SVC(kernel='linear')_	0.714386	0.496362	0.499696	0.431896	12790.0	0.596673	0.714386	0.609928	12790.0
_TFIDF_DecisionTreeClassifier()_	0.697420	0.510534	0.503018	0.462652	12790.0	0.605176	0.697420	0.620497	12790.0
_KNN_1	0.705004	0.476193	0.496318	0.436702	12790.0	0.584845	0.705004	0.609471	12790.0
_KNN_3	0.718374	0.479231	0.499253	0.423924	12790.0	0.587077	0.718374	0.606957	12790.0
_KNN_5	0.720485	0.473520	0.499589	0.421207	12790.0	0.583964	0.720485	0.606169	12790.0
_KNN_7	0.722205	0.361102	0.500000	0.419349	12790.0	0.521580	0.722205	0.605712	12790.0
_KNN_10	0.722205	0.361102	0.500000	0.419349	12790.0	0.521580	0.722205	0.605712	12790.0

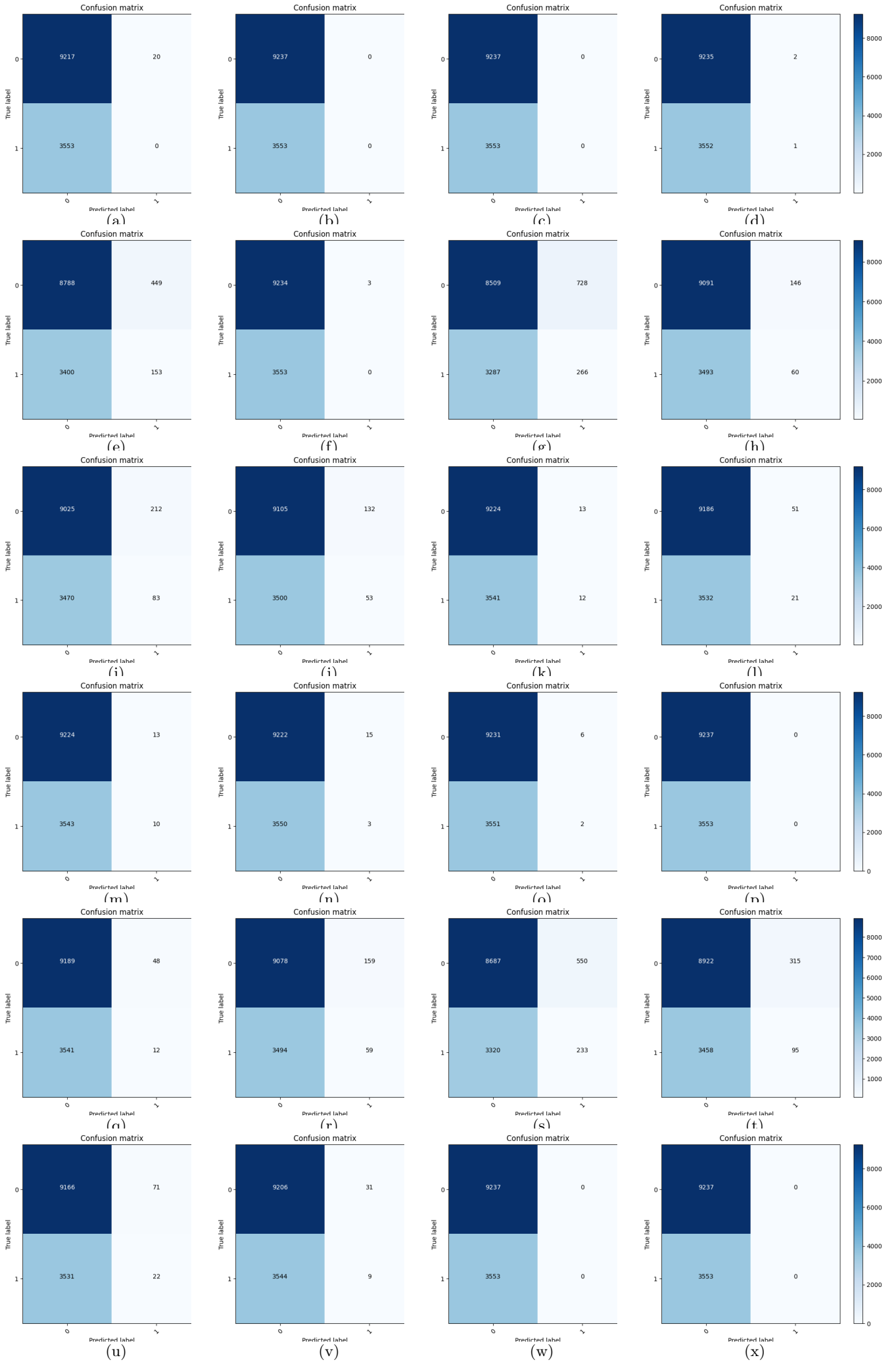


Figure 5: The models used in the above confusion matrices are respectively (a) bigram vectoriser with logistic regression; (b)

Model on LIAR dataset only with metadata

	accuracy	precision macro	recall macro	f1-score macro	support macro	precision weighted	recall weighted	f1-score weighted	support weighted
_ngram_2_LogisticRegression(solver='sag')_	0.721814	0.537623	0.500249	0.420854	12790.0	0.619697	0.721814	0.606391	12790.0
_ngram_3_LogisticRegression(max_iter=10000, n_j...	0.722205	0.361102	0.500000	0.419349	12790.0	0.521580	0.722205	0.605712	12790.0
_ngram_2_SVC(kernel='linear')_	0.721423	0.497462	0.499978	0.420719	12790.0	0.597336	0.721423	0.606200	12790.0
_ngram_3_SVC(kernel='linear')_	0.722127	0.361092	0.499946	0.419323	12790.0	0.521564	0.722127	0.605674	12790.0
_ngram_2_DecisionTreeClassifier()_	0.720876	0.464470	0.499599	0.420531	12790.0	0.578938	0.720876	0.605931	12790.0
_ngram_3_DecisionTreeClassifier()_	0.722048	0.486098	0.499978	0.419570	12790.0	0.591022	0.722048	0.605781	12790.0
_ngramKNN_2k=1	0.716028	0.466584	0.498235	0.424899	12790.0	0.579866	0.716028	0.606743	12790.0
_ngramKNN_2k=3	0.716966	0.454177	0.498105	0.422891	12790.0	0.572950	0.716966	0.605972	12790.0
_ngramKNN_2k=5	0.351290	0.485866	0.489399	0.345596	12790.0	0.580653	0.351290	0.318470	12790.0
_ngramKNN_2k=7	0.352072	0.486313	0.489681	0.346538	12790.0	0.581240	0.352072	0.319814	12790.0
_ngramKNN_2k=10	0.352463	0.486587	0.489865	0.346998	12790.0	0.581598	0.352463	0.320452	12790.0
_ngramKNN_3k=1	0.721814	0.537623	0.500249	0.420854	12790.0	0.619697	0.721814	0.606391	12790.0
_ngramKNN_3k=3	0.722127	0.361092	0.499946	0.419323	12790.0	0.521564	0.722127	0.605674	12790.0
_ngramKNN_3k=5	0.722127	0.527776	0.500032	0.419597	12790.0	0.614188	0.722127	0.605819	12790.0
_ngramKNN_3k=7	0.722127	0.527776	0.500032	0.419597	12790.0	0.614188	0.722127	0.605819	12790.0
_ngramKNN_3k=10	0.722205	0.361102	0.500000	0.419349	12790.0	0.521580	0.722205	0.605712	12790.0
TFIDF_LogisticRegression(solver='sag')_	0.700078	0.479957	0.495939	0.442463	12790.0	0.586858	0.700078	0.610888	12790.0
TFIDF_SVC(kernel='linear')_	0.697811	0.485807	0.496707	0.447269	12790.0	0.590247	0.697811	0.612648	12790.0
TFIDF_DecisionTreeClassifier()_	0.611415	0.489712	0.490843	0.489003	12790.0	0.590906	0.611415	0.600152	12790.0
_KNN_1	0.335184	0.477060	0.484916	0.325235	12790.0	0.568804	0.335184	0.288824	12790.0
_KNN_3	0.699844	0.522934	0.506687	0.468418	12790.0	0.612764	0.699844	0.624292	12790.0
_KNN_5	0.674433	0.496296	0.498274	0.473440	12790.0	0.596349	0.674433	0.618017	12790.0
_KNN_7	0.691087	0.505423	0.501837	0.466446	12790.0	0.602117	0.691087	0.620303	12790.0
_KNN_10	0.714543	0.514942	0.501450	0.436646	12790.0	0.607316	0.714543	0.612485	12790.0

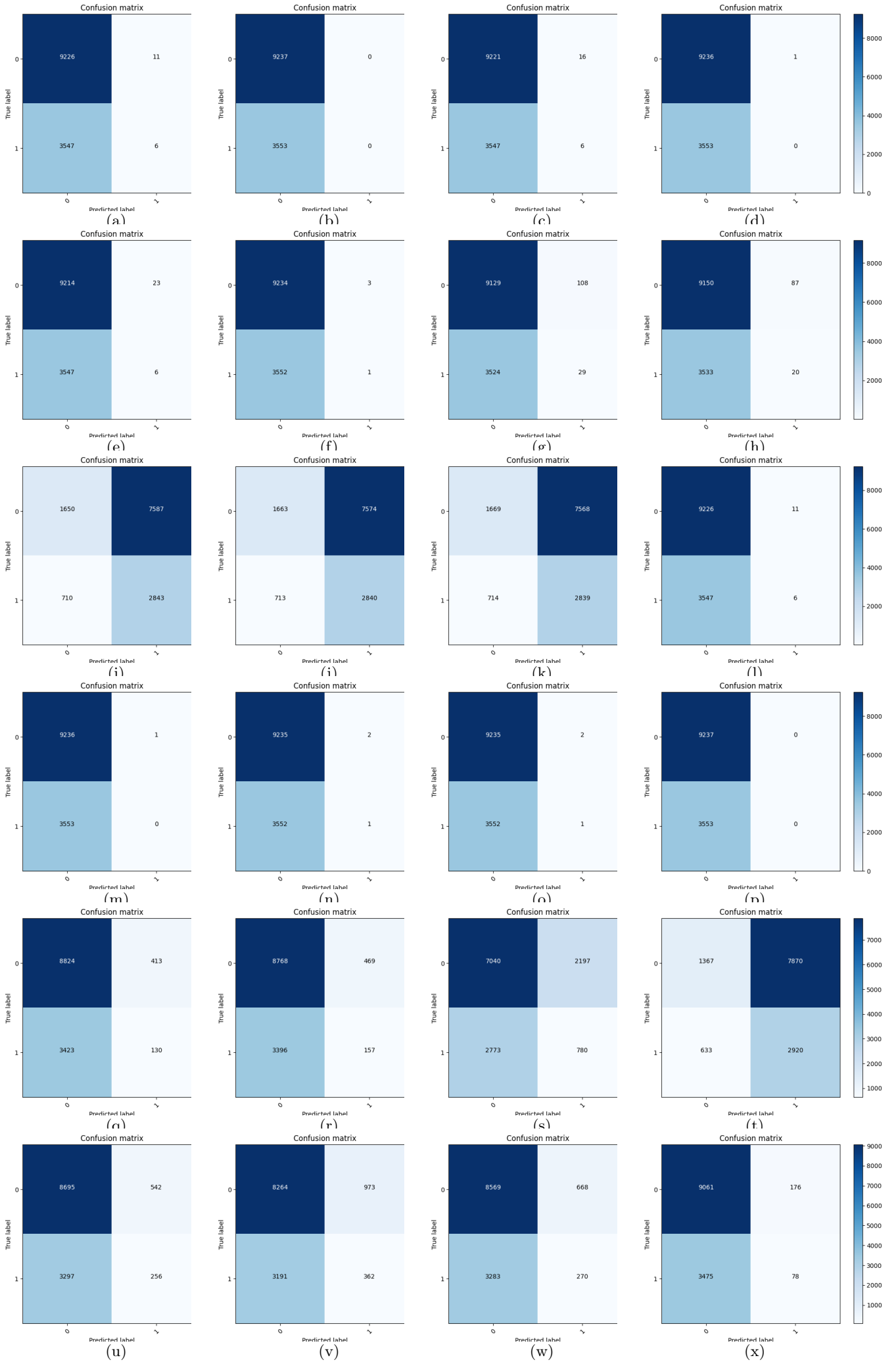


Figure 6: The models used in the above confusion matrices are respectively (a) bigram vectoriser with logistic regression; (b)